

UAB

**Universitat Autònoma
de Barcelona**

eyePictures

Memòria del projecte
d'Enginyeria Tècnica en
Informàtica de Sistemes

realitzat per

Pau Peris Rodriguez

i dirigit per

Daniel Riera Terrén

Escola Universitària d'Informàtica

Sabadell, setembre de 2009

El sotasignant, Daniel Riera Terrén
professor/a de l'Escola Universitària d'Informàtica de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en Pau Peris Rodriguez

I per a que consti firma la present.
Sabadell, Setembre de 2009

Signat: Daniel Riera Terrén

Resum introductori

El present projecte és la incorporació d'una aplicació de visualització i tractament d'imatges per a la plataforma web eyeOS¹. eyeOS és un projecte català de programari lliure que proporciona un escriptori web i un conjunt d'eines lliures per treballar-hi. El fet que sigui una eina web permet que s'integri dins qualsevol navegador d'Internet. Actualment està en etapa de desenvolupament i consta d'un paquet per defecte que proporciona aplicacions de sistema, ofimàtiques i multimèdia.

El programa desenvolupat, té per nom *eyePictures* i l'objectiu principal és que proporcioni un mètode ràpid i intuïtiu per a la visualització d'imatges, així com per el retoc d'aquestes mitjançant filtres fotogràfics. L'aplicació a més incorpora un minimalista però eficient navegador d'arxius que permet a l'usuari moure's per l'arbre de directoris en busca d'imatges.

Fins al moment previ al desenvolupament d'*eyePictures*, eyeOS incorpora un visor d'imatges que únicament permet fer clic sobre una imatge per a poder visualitzar-la i avançar o retrocedir el mostreig d'aquestes sempre i quan es trobin dins el mateix directori. L'aparició d'*eyePictures* dins eyeOS suposa un canvi radical en el que a la visualització d'imatges es tracta, ja que no només incorpora funcionalitats bàsiques com pot ser un ràpid accés a la meta-informació de les pròpies imatges sense haver d'accedir al navegador d'arxius, sinó que també ofereix característiques addicionals que faciliten la visualització avançada de fotografies i el seu retoc.

Com a característiques principals d'*eyePictures* cal destacar:

- ❖ Un navegador d'arxius que permet a l'usuari conèixer el nombre de directoris i fitxers de la ruta actual. Mostra també informació sobre el contingut dels directoris i imatges al passar el ratolí per sobre d'aquests. Òbviament també permet navegar per l'arbre de directoris entrant i sortint dels mateixos.

¹ <http://eyeos.org>

- ❖ Un visor d'imatges que permet desplaçar-se pel contingut fotogràfic del directori actual, avançant o retrocedint en el mostreig d'imatges.
- ❖ Rotacions arbitràries de les il·lustracions en angles de 90 graus.
- ❖ Diferents filtres que permeten a l'usuari tant canviar el color i tonalitat de les imatges com transformar les imatges mitjançant filtres professionals.
- ❖ Reproducció automàtica de diapositives mostrant cadascuna d'elles durant un temps predeterminat de 5 segons.
- ❖ Comprovació de l'existència de la llibreria GD, en la qual es recolza l'aplicació per fer ús del tractament d'imatges mitjançant *PHP*. Notificació a l'usuari en cas de no complir amb el requisit.
- ❖ Estètica i disseny treballat per a obtenir un bon impacte visual, prova d'això és l'efecte que realitzen les icones de l'aplicació al ser redimensionada juntament amb una barra d'eines acompanyada de modernes icones.

Índex

Resum introductori	3
Índex	5
1. Introducció	7
1.1. Actualitat tecnològica i "Cloud Computing "	7
1.2. Arquitectura, Avantatges i desavantatges	10
1.3. Exemples de serveis basats en "Cloud Computing"	14
1.4. Estat de l'art	17
1.5. Motivació	18
1.6. Objectius	19
1.6.1. Problemes i solucions	20
1.6.2. Descripció del sistema a realitzar	23
1.7. Planificació	23
1.7.1. Planificació inicial - Tasques	25
1.7.2. Planificació inicial - Diagrama de Gantt	26
1.7.3. Planificació final - Tasques	28
1.7.4. Planificació final - Diagrama de Gantt	28
2. Anàlisi de l'entorn	31
2.1. Història	31
2.2. eyeOS	31
2.3. Entorn	32
3. Organització del projecte	35
3.1. Especificació de requeriments	36
3.1.1. Requeriments funcionals	36
3.1.2. Requeriments no funcionals	38
3.1.3. Requeriments de rendiment	38
3.1.4. Requeriments de disseny	38
3.1.5. Objectius de disseny	39
3.2. Eines de desenvolupament	39
3.2.1. GNU/Gentoo Linux	39
3.2.2. Subversion	41

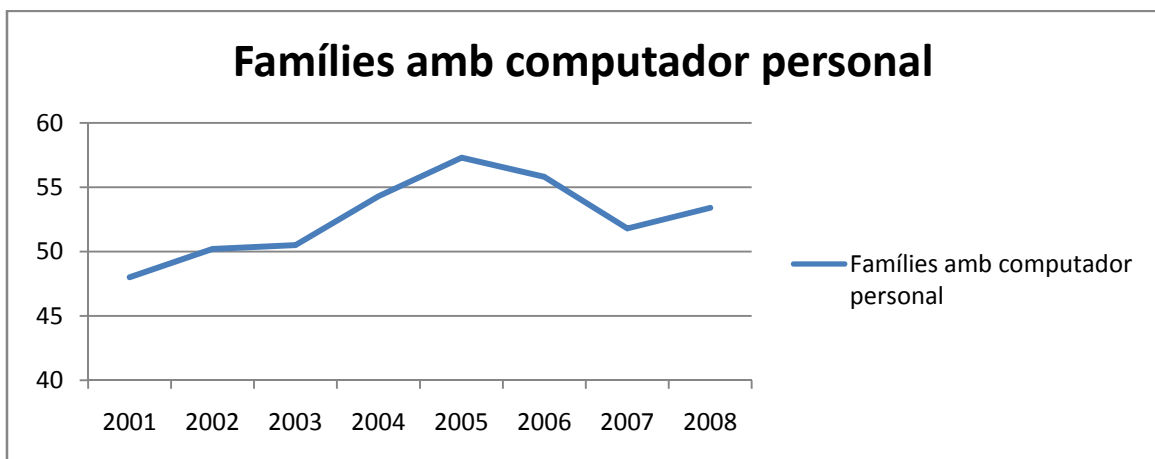
3.2.3. <i>Eclipse</i>	42
3.2.4. <i>PHPEclipse</i>	43
3.2.5. <i>Subclipse</i>	43
3.2.6. <i>Mozilla Firefox</i>	44
3.2.7. <i>Firebug</i>	45
3.2.8. <i>Firecookie</i>	46
3.2.9. <i>Apache</i>	46
4. Fonaments teòrics	47
4.1. Llenguatges i tecnologies utilitzats	47
4.1.1. <i>PHP</i>	47
4.1.2. Llibreria <i>GD</i>	48
4.1.3. <i>JavaScript</i>	49
4.1.4. <i>CSS</i>	50
4.1.5. <i>Toolkit d'eyeOS</i>	51
4.2. Formats d'imatge	53
4.2.1. <i>JPEG</i>	53
4.2.2. <i>BMP</i>	53
4.2.3. <i>GIF</i>	54
4.2.4. <i>PNG</i>	55
5. Anàlisi funcional	57
5.1. Presentació de l'aplicació	57
5.2. Composició i lògica	58
6. Disseny tècnic i codificació	63
6.1. Arquitectura tècnica de l'aplicació	63
6.2. Estructura de fitxers i directoris	64
6.3. Disseny de la interfície gràfica	73
7. Conclusions	75
7.1. Estat dels objectius	75
7.2. Possibles ampliacions	76
7.3. Valoració personal	76
8. Referències bibliogràfiques	78

1. Introducció

Aquest capítol presenta la tecnologia “*Cloud Computing*”², en la qual es basa l'eina producte d'aquest projecte, l'*eyePictures*. De l'estudi de les característiques, els avantatges e inconvenients, així com l'estat actual d'aquesta tecnologia, juntament amb les necessitats d'*eyeOS*, sorgeixen les motivacions per la realització del projecte, així com els seus objectius i la planificació — tant inicial (a priori) com final (a posteriori).

1.1 Actualitat tecnològica i “*Cloud Computing*”

En els últims quaranta anys la tecnologia de la informàtica no ha parat d'avançar a passos agegantats. El progrés tecnològic ha suposat una transformació de la societat en totes les seves àrees (educació, empresa i família) permetent així una millora del benestar. Aquestes tecnologies juguen un paper primordial en les nostres vides, facilitant o millorant la comunicació, l'accés a la informació o la difusió del coneixement. Tots aquests motius expliquen la seva elevada penetració i intensitat d'ús.



Imatge 1 - Gràfic de Famílies amb ordinador personal a casa.³

² Conegut també com "computació en el núvol" s'utilitza per referenciar un conjunt de serveis i aplicacions allotjades a Internet i accedides per usuaris finals des de qualsevol explorador d'Internet.

³ Societat de la informació 2001-2008.pdf, <http://www.conocimientoytecnologia.org>.

Entre tots aquests avenços que sorgeixen dia a dia, el sector multimèdia, i més concretament el de la imatge, no es queden enrere. I és que cada dia apareixen noves idees com pàgines web per a jocs i aplicacions ofimàtiques en línia, serveis de retoc fotogràfic o reproductors de contingut videogràfic. Amb el pas del temps la multimèdia ha format part de molts dels avenços tecnològics i no s'ha limitat únicament al desenvolupament de videojocs o jocs interactius com va succeir en els seus inicis, sinó que també s'ha dedicat a acaparar altres àrees. L'últim fenomen de masses ha estat imposat per les xarxes socials gràcies a la seva facilitat per moure enormes grups de gent. Ara bé, el que realment està marcant les últimes tendències de desenvolupament és el "*Cloud Computing*".

"*Cloud Computing*", el paradigma en el que la informació es troba emmagatzemada de forma permanent, mitjançant servidors, a Internet. "Computació en el núvol"⁴ és un concepte general que utilitza programari com a servei, tal com la Web 2.0⁵ on un dels factors més importants el juga la confiança de l'usuari en la xarxa permetent així satisfer les necessitats computacionals d'aquest. Aquesta emergent tecnologia suposa una revolució en la disposició tradicional de la informació, enfocant-se en la oferta de serveis a l'usuari i potenciant un estalvi de costos i flexibilitat que la converteixen en una opció molt atractiva pels usuaris. Gartner⁶ calcula el mercat actual de "*Cloud Computing*" en 46.000 milions de dòlars, pel que es preveu que cada cop més proveïdors muntin el seu propi negoci enfocant-lo al "núvol".

Ara que ja es té una visió definida del concepte de "*Cloud Computing*" resulta més senzill i intuïtiu enumerar les seves principals característiques, que provenen de les tecnologies que formen el "núvol" i de les noves capacitats conseqüents de la difusió d'aquests serveis:

⁴ En termes informàtics *núvol* és una metàfora que es refereix a Internet.

⁵ Després que les primeres pàgines web, estàtiques, aparegudes als inicis de d'Internet revessin el nom de Web 1.0, Tim O'Reilly utilitzà el terme Web 2.0 per referir-se a la segona generació en la història del desenvolupament de tecnologia web. Normalment basada en la inclusió d'una gamma especial de serveis tals com xarxes socials, blocs, *wikis*, fòrums, etc els quals fomenten la col·laboració i l'intercanvi àgil d'informació.

⁶ **Gartner** (<http://www.gartner.com/technology/home.jsp>) conegut també com *Grup Gartner* fins 2001, es tracta d'un grup empresarial d'investigació i assessoria en tecnologies de la informació (wikipedia, 2009).

- Escalabilitat, el sistema treballa de forma eficient augmentat o disminuint l'ús dels recursos del sistema sota demanda, de forma completament automàtica i transparent per l'usuari.
- Virtualització, es reflexa principalment en dos punts. L'usuari pot fer ús de la plataforma o entorn que desitgi, ja sigui mitjançant la contractació o a través de creacions pròpies. L'altre punt a enumerar és la capacitat d'independència que existeix entre dades i hardware, ja que les aplicacions que s'executen es troben en servidors i són accedides a través del navegador del client, augmentant d'aquesta forma la llibertat d'elecció d'aquest i el nombre de dispositius als quals es proporciona accés.
- Trobar-se regida per un Acord de Nivell de Servei (SLA), que defineix un conjunt de polítiques per assegurar el correcte funcionament com és el rendiment, el temps d'accés, la capacitat de tràfic, els pics de connexió suportats, etc. Tot això està directament relacionat amb al escalabilitat ja que depenent d'aquestes termes el sistema s'encarrega de gestionar els recursos.
- Accessible, bàsicament pel que ja s'ha vist en els punts anteriors, l'ampli ventall de dispositius es multiplica.

Entre els principals serveis nuvolers destaquen *Google Apps*⁷, *Amazon EC2*⁸ o *photoshop.com*⁹.

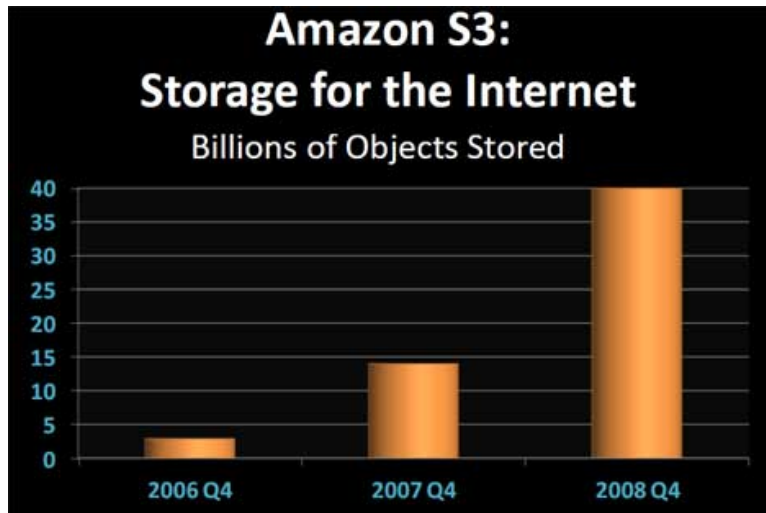
Google Apps ofereix serveis de missatgeria instantània i de calendari, també brinda accés a *Google Docs*, *Google Vídeo* i *Google Sites* que permeten l'edició i compartiment de documents en línia, emmagatzemament de fotografies, serveis de traducció i allotjament de blocs.

⁷ <http://www.google.com/apps/>

⁸ <http://aws.amazon.com/ec2/>

⁹ <https://www.photoshop.com/>

Amazon EC2, que es troba en plena producció des de fa 10 mesos, ofereix serveis web comercials que consisteixen en el lloguer d'equips informàtics per a la execució d'aplicacions. Així doncs, *EC2* permet la implantació escalable d'aplicacions proporcionant una xarxa de serveis amb interfície d'usuari amigable de forma que un client pot crear màquines virtuals¹⁰ que funcionen com a servidor privat on es pot carregar qualsevol tipus de programari a pròpia elecció.

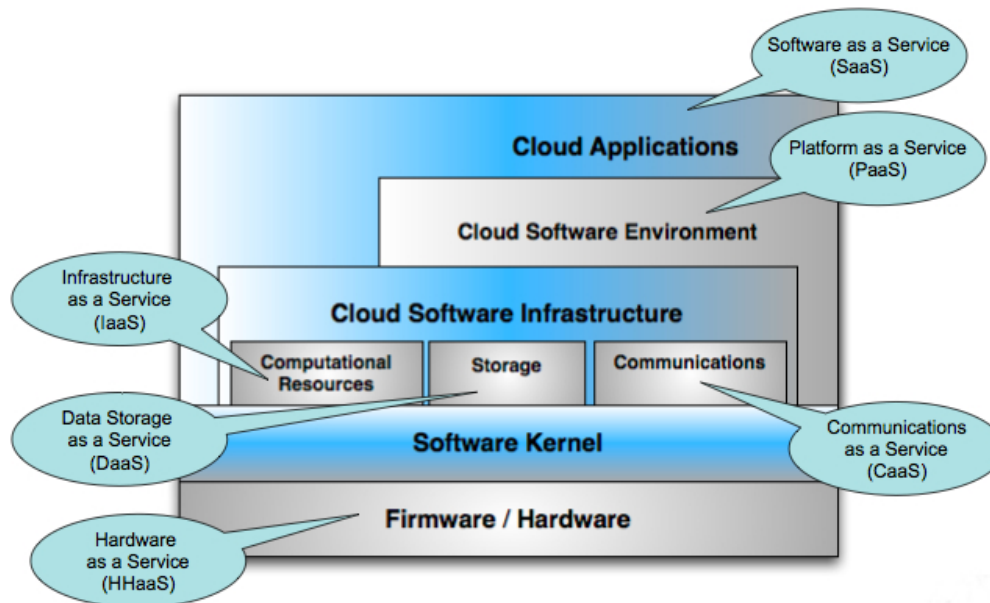


Imatge 2 - Gràfic del nombre d'objectes emmagatzemats per *Amazon* (Font: <http://www.datacenterknowledge.com>)

1.2 Arquitectura, avantatges i desavantatges

Com s'ha pogut comprovar durant el transcurs de les cites anteriors, la definició de "*Cloud Computing*" és molt ambigua i pot tenir diferents variants, però si sembla existir un consens en el que a les diferents parts que componen aquesta tecnologia es refereix, identificant com a tals les capes d'una arquitectura completa.

¹⁰ S'entén per **màquina virtual** qualsevol programari capaç d'emular el funcionament d'un ordinador i executar aplicacions com si d'una computadora personal es tractés.



Imatge 3 - "Cloud Computing" desglossat en capes.¹¹

El gràfic superior és bastant auto-descriptiu, tot i així a continuació detallarem breument els aspectes més característics de cadascuna de les seves capes centrant-nos en les principals, com són *SaaS*, *PaaS* i *IaaS*.

- *SaaS (Software As A Service)*, programari com a servei, es tracta de la capa superior d'aquest model de computació i consisteix en la distribució de programari on una empresa s'encarrega del manteniment, suport i operació que utilitzarà el client durant el temps que hagi contractat els seus serveis. Aquest tipus d'aplicacions són accessibles des del navegador de l'usuari i permeten el maneig de la informació de forma remota, ja que aquesta queda emmagatzemada en el propi servidor. Alguns exemples de servei són: *GMail*, *Google Docs*, *Amazon S3*, etc
- *PaaS (Platform As A Service)*, plataforma com a servei, és la capa que ofereix el suport directe al programari que es troba al "núvol" fent ús de la infraestructura contractada per cada client. El seu principal ús consisteix en oferir una solució completa per a la construcció i posta en

¹¹ Font: <http://trycatch.lacotelera.net/post/2008/12/18/cloud-computing-arquitectura-ventajas-y-desventajas>

marxa de les aplicacions i serveis web que estaran completament disponibles a través de la xarxa. Alguns exemples poden ser: *Google Apps Engine*, *Amazon SimpleDB*, etc

- *IaaS (Infrastructure As A Service)*, infraestructura com a servei, és considerada per alguns com la capa inferior del model complet (inclús per sota la capa *HaaS*) en canvi per altres, com és el cas del gràfic anterior, la zona intermèdia de tota la arquitectura. En ambdós casos la seva utilitat és la mateixa, proporcionar al client una infraestructura de computació com un servei, utilitzant principalment la virtualització. El client compra recursos a un proveïdor extern, per allotjament, capacitat de còmput, manteniment i gestió de xarxes, etc beneficiant-se d'aspectes com la escalabilitat. Un bon exemple és *Azure* de *Microsoft*.
- *DaaS (Data Storage As A Service)* i *CaaS (Communications As A Service)*, emmagatzemament de dades com a servei i comunicacions com a servei són dues capes que treballen horitzontalment i juntament amb *IaaS*. *DaaS* proporciona gestió i manteniment complet de les dades manejades per clients i *CaaS* proveeix l'equipament de xarxes i la gestió d'aspectes com balanceig de càrrega.
- *Kernel* de programari, és la capa que gestiona els servidors físics a través del *Sistema operatiu*¹², programari de virtualització, *middleware*¹³ de gestió de *clusters*¹⁴, etc.
- *HaaS (Hardware As A Service)*, hardware com a servei, és la primera escala de la piràmide i com el seu nom indica es tracta d'elements de físics dins el "núvol". Es tracta de centres de dades gegantescos amb tot

¹² Conjunt d'eines de programari de sistema destinades a realitzar diferents tasques entre les que destaquen la correcta administració de recursos de maquinari i la interacció entre aquest i l'usuari final.

¹³ Middleware és un programari de connectivitat que ofereixen un conjunt de serveis que fan possible el funcionament d'aplicacions distribuïdes sobre plataformes heterogènies.

¹⁴ També conegut com "*Data Centers*", es un conjunt d'ordinadors amb característiques de servidor, normalment mantinguts per empreses i destinats a l'emmagatzemament de dades de forma redundat, oferiment de serveis i infraestructures de telecomunicacions.

tipus de màquines que fan possible la computació, l'emmagatzemament, etc. Com s'ha assenyalat anteriorment també és vàlid considerar els últims quatre punts com una sola capa.

Ara que l'estructura del "núvol" ja no és quelcom desconegut, ha arribat el moment de llistar seves principals avantatges i desavantatges que presenta el seu ús:

- Escalabilitat, virtualització i facilitat d'accés.
- Capacitat de processament i d'emmagatzemament sense haver de fer cap instal·lació local prèvia.
- Baix cost, tot i existir serveis gratuïts i altres de pagament, resulta més rentable per les empreses el contracte d'aquestes infraestructures que no pas el de serveis separats o la compra de "*Data Centers*¹⁵" propis.
- Comunicacions unificades, possessió de programari i xarxa social, i especialització dels sistemes.

Desavantatges:

- Seguretat de la informació, com que tota aquesta es troba en mans de terceres empreses aquest pot ser un punt important a tenir en compte.
- Dependència dels serveis en línia dels proveïdors, encara que cada cop disminueix el problema ja que segons els contractes de *SLA* és menys probable una caiguda o falla en aquests sistemes que en un allotjament habitual (*Amazon* ofereix un 99,95% de fiabilitat, el que es tradueix en 4.35 hores de fallada del servei de forma anual).

¹⁵ També conegut com "*clusters*", referència ⁴, pàgina 12.

1.3 Exemples de serveis basats en "Cloud Computing"

Com a concepte i unió de diferents sectors de computació només existeix un "Cloud Computing", però a la hora d'oferir-la i lluitar per una porció important dels seus ingressos es pot trobar diversificacions del mateix.

- *Amazon*. Com a empresa que ofereix un conjunt de serveis basats en la computació en el "núvol" *Amazon* és, avui dia, l'empresa que ofereix una solució més completa. A continuació s'exposa una llarga llista de tots els serveis contractables a través d'*Amazon*: *S3 (Simple Storage Service)* és un servei d'emmagatzemament escalable de baix cost al qual s'accedeix a través de serveis web, *EC2 (Elastic Comput Cloud)* és un servei que proporciona capacitat de còmput escalable, *CloudFront o CDN (Content Delivery Networks)* treballa com a memòria cau¹⁶ del contingut dels clients que en realitat és una replica de les seves peticions per accedir de forma més ràpida als continguts, *SimpleDB (Simple DataBase)* és un servei que permet fer consultes sobre dades estructurades en temps real oblidant-te de la dificultat implícita de les bases de dades estructurals,

SQS (Simple Queue Service)

és bàsicament

un servei de

cuà de

missatges,

escalable i que

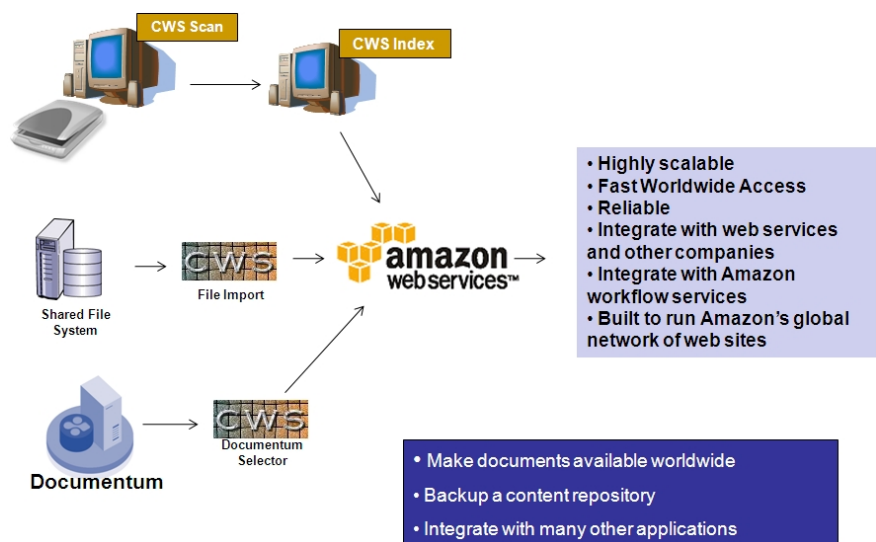
permet

l'intercanvi

d'informació entre

computadors

evitant la pèrdua



Imatge 4 - Amazon web services (Font: <http://contentworkspace.com>)

¹⁶ La memòria cau és un tipus de memòria intermèdia d'alta velocitat que s'encarrega de mantenir l'accés a informació de forma ràpida i eficient. Pot ser tant de tipus programari com maquinari.

de dades i sense la necessitat de que cada component hagi d'estar disponible.

Com s'ha pogut observar *Amazon* ofereix serveis per a cadascuna de les capes que formen l'entramat de computació en el "núvol".

- *Google App Engine*, és la proposta de l'empresa *Google* i planteja una infraestructura d'allotjament, processament i bases de dades com a servei. Per reflexar millor la seva oferta mencionarem les diferències més significatives existents entre *Amazon Web Service* i *Google App Engine*. La solució de *Google* consta d'un paquet complet que no es pot dividir, en canvi *Amazon* posa a disposició del client cadascun per separat. En qüestió de llenguatges de programació, *Google*, fins al moment, suporta únicament *Python* tot i que te pensat suportar altres llenguatges a més ofereix un framework¹⁷ de desenvolupament propi que ja integra funcions de la API¹⁸ com a serveis, al contrari que *Amazon* que suporta varis llenguatges oferint API's en lloc de frameworks.



Per últim mencionar que pot resultar molt útil poder integrar aquests serveis

amb altres, que fins al moment també es troben gratuïtament, com són

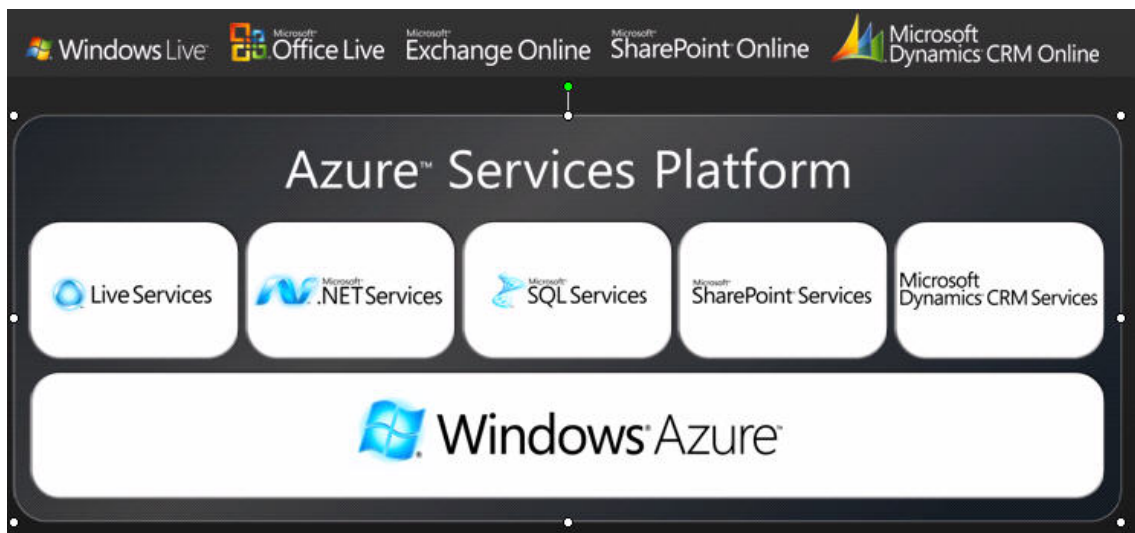
Imatge 5 - A l'esquerra Amazon web services, a la dreta infraestructura Google (Font: <http://contentworkspace.com>)

¹⁷ **Framework**, en el desenvolupament de software fa referència a una estructura de suport definida, mitjançant la qual un altre projecte pot ser organitzat o desenvolupat.

¹⁸ **Interfície de Programació d'Aplicacions (API)**, representa una interfície de comunicació entre components de programari. Es tracta d'un conjunt de crides a biblioteques de sistema que ofereixen accés a certs serveis a través de processos, es tracta d'una forma d'aconseguir abstracció entre capes i/o aplicacions.

GMail o GCalendar.

- *Microsoft Windows Azure*, és l'equivalent a la solució *Google* que proposa *Microsoft*. Aquesta consta d'un "núvol de serveis" útils per a programadors, per a serveis d'allotjament o de gestió d'entorns. Proporciona les característiques de computació i emmagatzemament sota demanda, escalabilitat i de manteniment d'aplicacions Web a través dels seus propis "*Data Centers*". Per construir les aplicacions es disposa de *Microsoft Visual Studio 2008*, que a més suporta estàndards i protocols del tipus *SOAP*¹⁹, *REST*²⁰ i *XML*²¹. *Azure* és una plataforma oberta que suportarà tant llenguatges i entorns propis de *Microsoft* com aliens.



Imatge 6 - Windows Azure, proposta de Microsoft per al "Cloud Computing"
(Font: <http://n99projects.web.officelive.com>)

¹⁹ **Simple Object Access Protocol (SOAP)**, és un protocol estàndard que defineix com dos objectes pertanyents a diferents processos poden comunicar-se per mitjà d'intercanvis de dades en format XML.

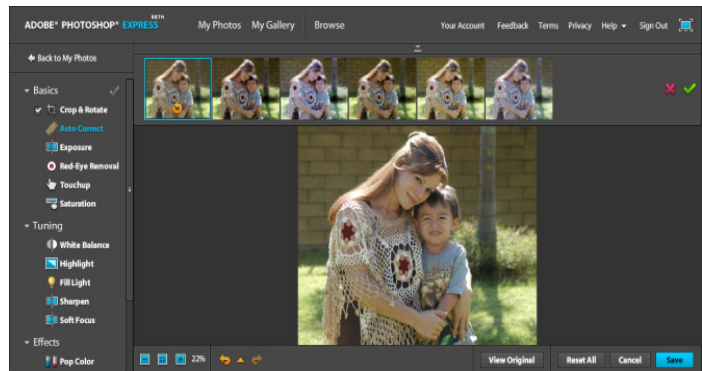
²⁰ **Representational State Transfer (REST)**, consisteix en una tècnica d'arquitectura de programari per a sistemes distribuïts hipermedia com la World Wide Web.

²¹ **Extensible Markup Language (XML)**, és un meta-llenguatge extensible d'etiquetes, desenvolupat per a la World Wide Web.

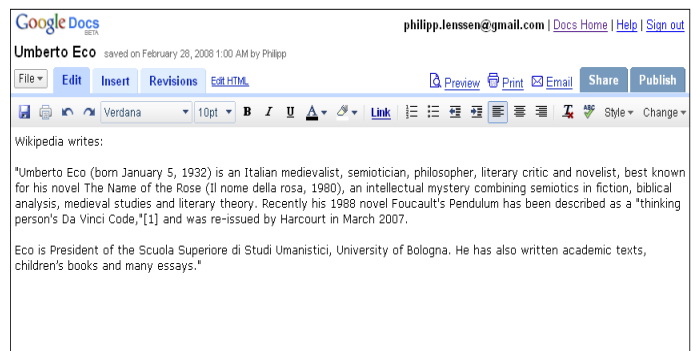
1.4 Estat de l'Art

Tot i que aquest projecte consisteix en el desenvolupament d'una eina de visualització i tractament d'imatges per el que ja és un producte de "Cloud Computing", a continuació es mostren alguns exemples de serveis Web 2.0 basats en "computació als núvols":

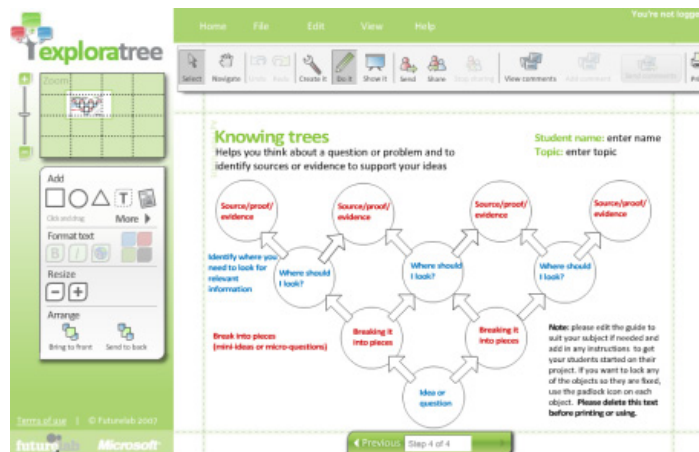
- Adobe tampoc es vol perdre el negoci que suposa el "Cloud Computing" i ha posat a disposició de tots els usuaris, mitjançant la web www.photoshop.com, una versió reduïda i en línia del seu producte estrella *Adobe Photoshop* per a poder retocar imatges i contingut fotogràfic des de la xarxa, emmagatzemar-les, accedir a elles des d'una ruta del tipus *nomdusuari.photoshop.com* inserir-les en altres pàgines web, etc.



- Aquest cop, *Google* ens ofereix a través de *Google Docs* la possibilitat d'editar contingut ofimàtic en línia utilitzant potents editors de textos al més pur estil *MS Word*.



- Aquesta aplicació, a la qual es pot accedir des de www.exploratree.org.uk, ens permet realitzar, guardar, explorar i compartir diagrames de forma totalment gratuïta i en línia.



1.5 Motivació

El temps de vida del terme "*Cloud Computing*" és curt ja que les tecnologies i serveis que els formen així ho són igualment. Tot i així, el concepte no és nou, utilitzar la computació i la xarxa per a ús públic i obert sempre ha estat en ment dels gurus d'Internet. Si bé és cert que el mot "Cloud" ja va ser utilitzat a principis dels anys 90 per fer referència a xarxes *ATM*²², no ha sigut fins al mateix segle XXI que les solucions han aparegut en el mercat sota el nom de "*Cloud Computing*".

Si bé cal destacar un nom propi en el tema al que estem fent referència, aquest és *Amazon*. La seva iniciativa per aprofitar els seus grans "*Data Centers*" que la major part del temps es trobaven sense prou càrrega de treball, ha fet que amb el temps anessin modernitzant i desenvolupant una arquitectura tant eficient i innovadora que preveu accés d'usuaris als seus sistemes a través d'*Amazon Web Services* els quals ja l'any 2002 oferien un conjunt d'utilitats bàsiques. Amb el pas dels anys, *Amazon* s'ha aferrat com a proveïdor de programari, de plataforma i d'infraestructura. Per altra banda, l'aparició de gegants incontestables en el mercat com *Google*, *IBM*, i com no, *Microsoft*, donen al "núvol" una importància a tenir en compte de forma molt seria en el món tecnològic actual i al futur immediat.

Els anys 2007 i 2008 van veure incrementar l'aparició del terme sempre orientat sobretot a solucions professionals i del món universitari, molt influent en qualsevol referent mundial i d'avenç tecnològic. Però ha sigut aproximadament a mitjans del 2008 quan ja nombrosos esdeveniments al voltant de "*Cloud Computing*" han començat a planificar-se.

Com s'ha pogut constatar hi ha varies empreses punteres oferint serveis de "*Cloud Computing*", gairebé totes elles de programari privatiu, mentre que els projectes de programari lliure basats amb el "*Cloud Computing*" no són gaires encara n'hi ha menys si es comptabilitzen a nivell nacional. És per aquest motiu que agafant un dels projectes, de codi obert, més importants a nivell

²² *Asynchronous Transfer Mode (ATM)*, és una tecnologia desenvolupada per fer front a la gran demanda de transmissió de serveis i aplicacions.

nacional com és eyeOS, i estudiant la seva tecnologia, he decidit desenvolupar una evolució d'alguna de les seves aplicacions essencials que incorpora el paquet base. D'aquesta forma espero que a petit o mitjà termini es substitueixi l'eina que ve per defecte per la que he realitzat com a projecte de fi de carrera. I és que tenint en compte el gran creixement que està tenint eyeOS i la gran distància que pot recórrer, no s'acaba d'entendre que algunes de les seves aplicacions com el visor d'imatges que incorpora per defecte constin d'unes característiques tant senzilles o gairebé nul·les. D'aquesta manera, i aprofitant les avantatges del programari lliure, puc incorporar una aplicació personal, que de moment es troba disponible a www.eyeos-apps.org, a un projecte que promet fer-se un lloc en l'entramat negoci nuvoler.

Especialistes de la xarxa assenyalen el "núvol" com el camí a seguir durant els transcurso del 2009 i els anys vinents. El que és essencial ara és saber vendre el producte, detallant la seva arquitectura i venent els seus avantatges a les empreses davant els seus possibles detractors i les seves possibles alternatives.

1.6 Objectius

Com s'ha pogut comprovar en el resum introductori (apartat 1) l'objectiu final és el de realitzar una aplicació multimèdia per a la visualització i tractament d'imatges que es pugui incrustar en un producte de computació en el "núvol", basat en programari lliure, de forma que aquest sigui accessible per el màxim nombre de persones. La seva voluntat principal, però, és la de millorar en tots els aspectes l'aplicació ja existent, de manera que accedir al contingut fotogràfic a través d'eyeOS sigui el més senzill i còmode possible, sempre utilitzant una interfície gràfica senzilla i intuïtiva.

Per al desenvolupament de l'aplicació s'han utilitzat tecnologies orientades a la programació web com són *PHP*, *JavaScript*, *CSS* i el propi "*toolkit*"²³ d'eyeOS que s'encarrega de la creació dels propis objectes que permeten la creació de finestres i botons de forma ràpida i senzilla.

²³ *Toolkit* és normalment quelcom que fa referència a un conjunt d'eines de desenvolupament.

És important assenyalar que per a la creació d'*eyePictures* no únicament he hagut d'implementar la interfície gràfica per a cadascun dels modes (navegador d'arxius i visualitzador d'imatges) sinó que també s'han hagut d'escriure les llibreries que permeten la creació de previsualitzacions incrustades en el navegador d'arxius i el mostreig d'imatges a part del conjunt de funcions que s'encarreguen del tractament del contingut, com poden ser la discriminació de directoris respecte altres tipus d'arxius, la rotació de les pròpies imatges, etc.

Així, en termes molt generals, els principals objectius plantejats a l'hora de realitzar aquest projecte han estat:

- 1.- Estudi de l'estat de l'art en quant a aplicacions de "*Cloud Computing*" actuals, i anàlisi d'*eyeOS* per decidir quina aplicació seria interessant incorporar-hi.
- 2.- Familiarització amb la "*toolkit*" d'*eyeOS* i amb les tecnologies de programació web necessàries per implementar una aplicació en aquest sistema.
- 3.- Anàlisi, disseny , implementació i incorporació a *eyeOS* d'una aplicació plenament funcional.

1.6.1 Problemes i Solucions

Com es pot constatar a continuació, el visor d'imatges que introdueix *eyeOS* en el seu conjunt d'aplicacions bàsiques consisteix en una aplicació senzilla sense funcionalitats complementàries.

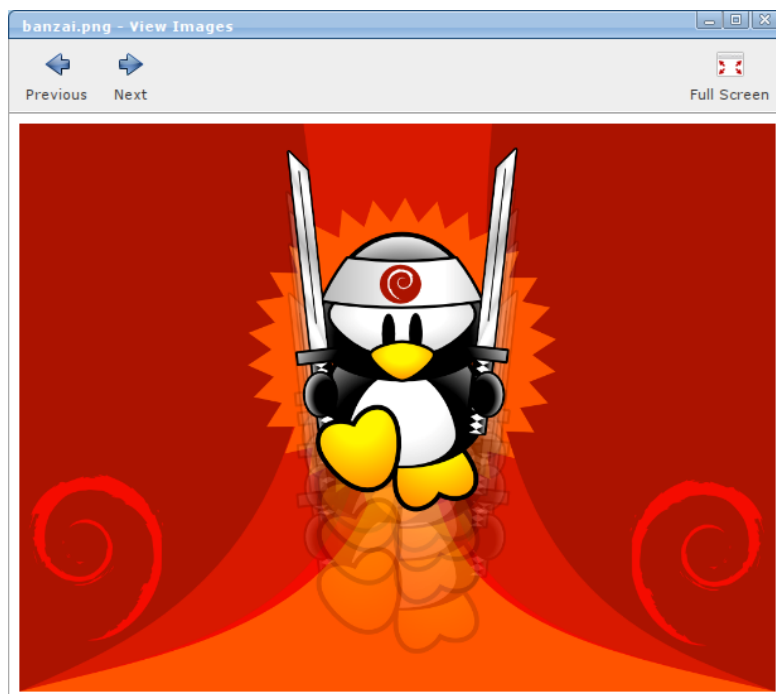
Algunes de les limitacions imposades a l'usuari es detallen a continuació:

- Manca d'informació sobre la fotografia mostrada, com pot ser la data de creació, resolució, format de la imatge, etc.
- Impossibilitat de visualitzar il·lustracions de directoris diferents a l'actual sense haver de tancar l'aplicació, obrir el navegador d'arxius per desplaçar-se al directori desitjat i visualitzar les imatges en qüestió.

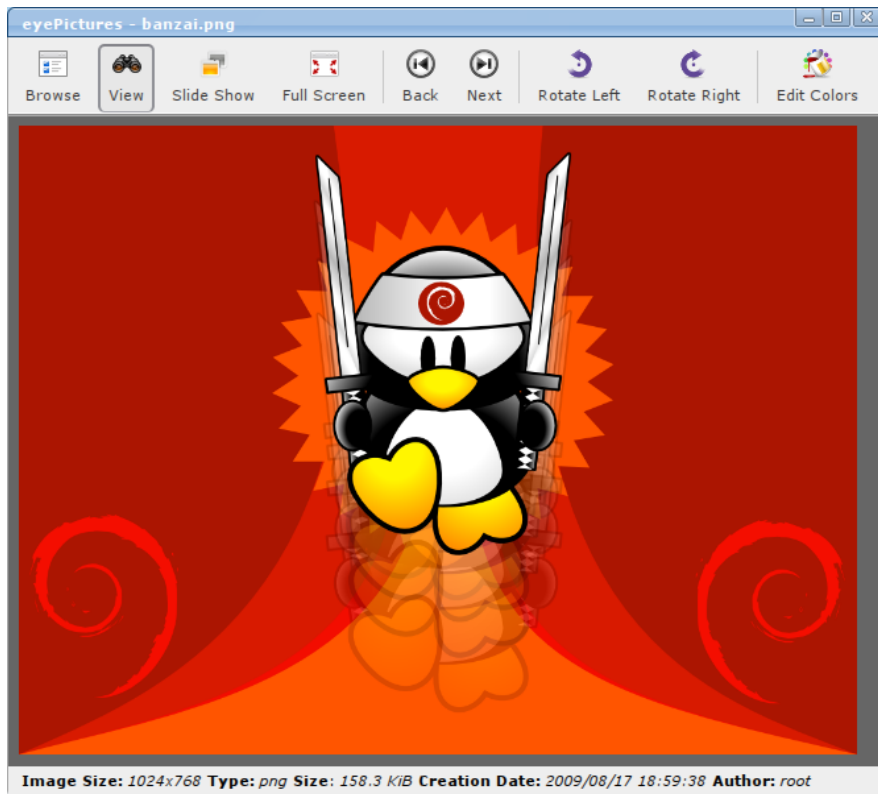
- Incapacitat de rotar imatges, el qual nega a l'usuari una experiència agradable durant la visualització d'aquestes.
- Omissió de les comprovacions de requisits mínims necessaris, com és el cas de l'existència de llibreries per al tractament d'imatges, i posteriors notificacions d'usuari necessàries.

Per tant, com a resposta a totes aquestes deficiències la proposta que es detalla en el transcurs del document intenta solucionar totes aquestes mancances anteriorment mostrades i també incorporar funcionalitats extres que permetin a l'usuari gaudir d'una major comoditat en la visualització d'aquest tipus de contingut.

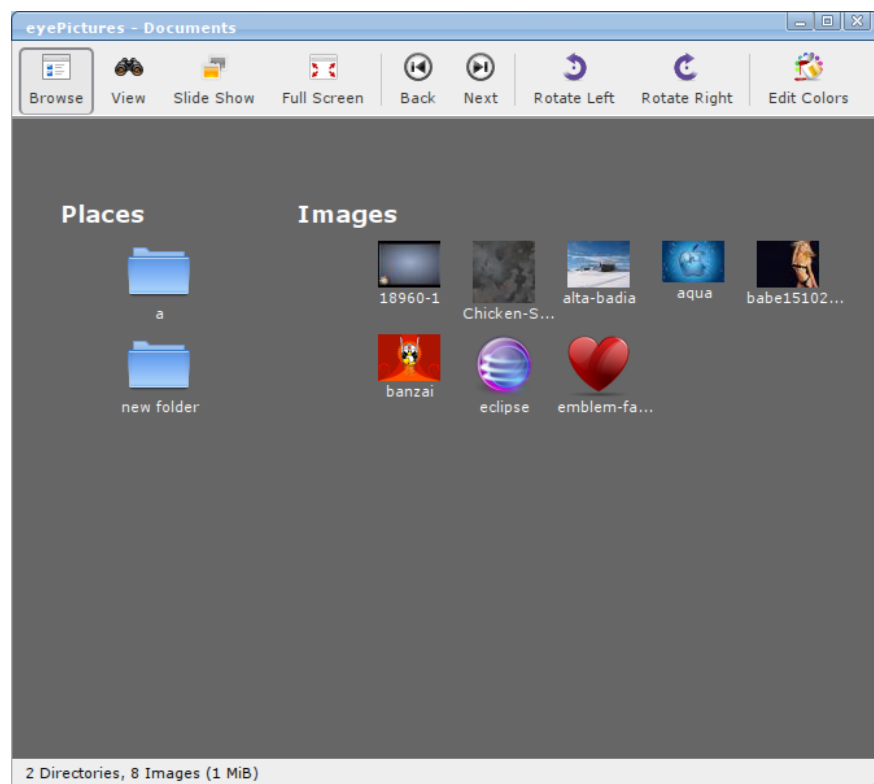
A continuació es poden observar un conjunt d'il·lustracions que permeten apreciar les carències i les diferències existents entre el visor d'imatges per defecte (imatge 10) i *eyePictures* (imatges 11 i 12).



Imatge 10 - Visor d'imatges que incorpora eyeOS per defecte



Imatge 11 - eyePictures en mode visor d'imatges



Imatge 12 - eyePictures en mode explorador d'arxius

1.6.2 Descripció del sistema a realitzar

L'aplicació a desenvolupar ha de constar d'una interfície gràfica senzilla però al mateix temps completa, amigable i intuïtiva, i que no descuidi els detalls estètics com són modernes icones, tant per a la barra d'eines com en el mostreig dels directoris, i efectes de redimensionament.

Per al desenvolupament de l'esmentada interfície d'usuari s'ha fet ús del "toolkit" de desenvolupament que incorpora eyeOS, per a una senzilla i ràpida implementació, i creació d'objectes gràfics. També del llenguatge de programació web *PHP* acompanyat de *JavaScript* per a manipular la informació mostrada a l'usuari.

S'ha programat també un motor capaç de tractar amb els formats gràfics actuals més utilitzats mitjançant l'ús de la llibreria *GD*, per a *PHP*, que incorpora un conjunt de mètodes per al tractament d'imatges.

1.7 Planificació

Tal com s'explica a l'apartat 3 (pàgina 32), la metodologia de planificació, desenvolupament i manteniment que s'utilitzarà per a la implementació d'eyePictures és la MÈTRICA 3²⁴, proposada pel Ministeri d'Administracions Públiques d'Espanya.

D'aquesta forma l'etapa de desenvolupament del projecte queda dividida en quatre fases principals:

- Investigació i anàlisi (90 hores). Aquesta és una de les etapes més feixugues, ja que consisteix en l'estudi de tecnologies amb les que normalment no s'ha treballat amb anterioritat. En el cas que ens ocupa, les hores dedicades en aquest apartat s'han invertit en l'estudi de la plataforma eyeOS i el conjunt de tecnologies utilitzades per al seu funcionament. Per a obtenir aquest tipus de documentació s'ha visitat la web oficial d'eyeOS²⁵ i la seva pàgina *wiki*²⁶. Un cop reunida i compresa

²⁴ <http://www.softqanetwork.com/2009/06/%C2%BFque-es-la-metodologia-metrica-3>

²⁵ <http://eyeos.org/>

²⁶ <http://wiki.eyeos.org>


la informació relativa al seu funcionament, s'ha fet un estudi sobre el mètode de desenvolupament d'aplicacions per a eyeOS i dels llenguatges de programació amb els que es permet interactuar.

- Disseny (35 hores). Aquesta secció compren tot allò relacionat amb el disseny tècnic de l'aplicació. En aquest cas, les hores han estat destinades al disseny de la interfície gràfica d'usuari i de la composició tècnica de la pròpia aplicació, com és la definició del contingut de cada fitxer i la seva divisió en funció de les tasques a desenvolupar.
- Codificació (145 hores). Segurament es tracta de la fase més estressant del projecte, ja que sovint apareixen imprevistos i situacions de complicada solució no contemplades en la fase de disseny degut a la seva espontaneïtat.
- Proves (20 hores). Aquest punt consisteix en l'execució de les proves necessàries per a la detecció i posterior correcció d'errors de programació o de possibles comportaments anòmals de l'aplicació. Aquest apartat s'ha dut a terme en el transcurs del desenvolupament del programa (etapa de codificació) i s'ha intensificat un cop acabat, per garantir un comportament fiable i homogeni davant una gran varietat d'escenaris. Cal remarcar que aquesta és una aplicació destinada a una gran varietat d'usuaris finals, pot ser descarregada des de la web d'aplicacions d'eyeOS <http://eyeos-apps.org>.
- Elaboració de la documentació (60 hores). Es tracta de la fase final del projecte, però no per això la més curta i menys important. Requereix una inversió important en quant a dedicació i organització de la informació recaptada fins al moment.

1.7.1 Planificació inicial - Tasques

A continuació es mostra una taula amb el llistat de tasques i sub-tasques que formen part del desenvolupament del projecte. En dita taula es pot distingir entre:

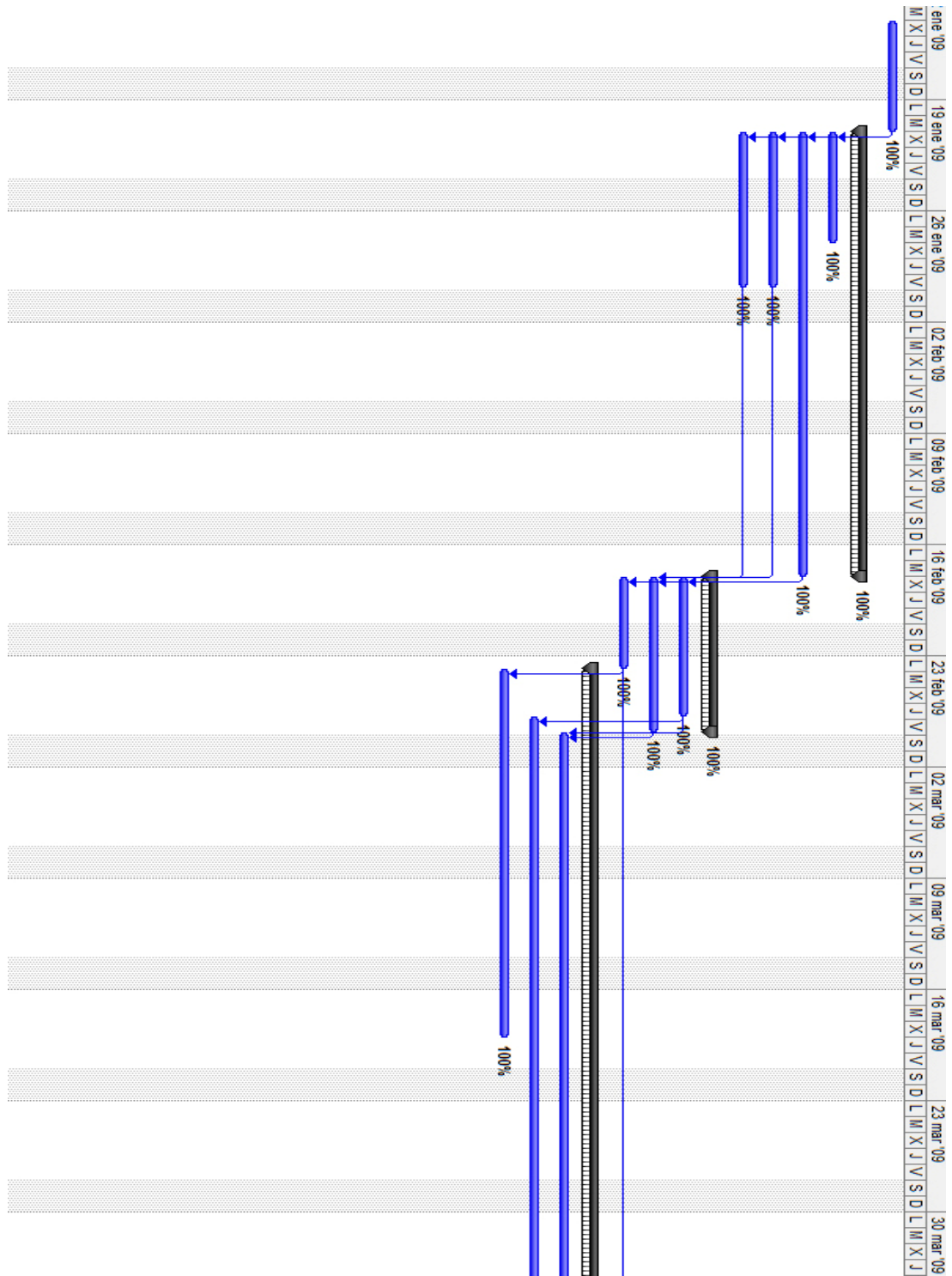
- Duració: duració real de les diferents tasques.
- Inici: data d'inici de cadascuna de les parts.
- Fi: data d'inici de cadascuna de les parts del projecte.
- Predecessores: Tasques de finalització prèvia obligatòriament.
- Nom del recurs: Únicament es disposa d'un recurs de personal que s'encarrega de desenvolupar el projecte.

		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	<input checked="" type="checkbox"/>	Anàlisis	10 horas	mié 14/01/09	mar 20/01/09		Projecte
2	<input checked="" type="checkbox"/>	<input type="checkbox"/> Investigació	20 días	mié 21/01/09	mar 17/02/09		Projecte
3	<input checked="" type="checkbox"/>	Sistema eyeOS	10 horas	mié 21/01/09	mar 27/01/09	1	Projecte
4	<input checked="" type="checkbox"/>	Toolkit eyeOS	40 horas	mié 21/01/09	mar 17/02/09	1	Projecte
5	<input checked="" type="checkbox"/>	Llibreria GD	15 horas	mié 21/01/09	vie 30/01/09	1	Projecte
6	<input checked="" type="checkbox"/>	JavaScript i PHP	15 horas	mié 21/01/09	vie 30/01/09	1	Projecte
7	<input checked="" type="checkbox"/>	<input type="checkbox"/> Disseny	7,5 días	mié 18/02/09	vie 27/02/09		Projecte
8	<input checked="" type="checkbox"/>	Característiques d'eyef	13 horas	mié 18/02/09	jue 26/02/09	4	Projecte
9	<input checked="" type="checkbox"/>	Llibreria d'aplicació	15 horas	mié 18/02/09	vie 27/02/09	4;5;6	Projecte
10	<input checked="" type="checkbox"/>	Interfície gràfica	7 horas	mié 18/02/09	lun 23/02/09	4	Projecte
11	<input checked="" type="checkbox"/>	<input type="checkbox"/> Codificació	33 días	lun 23/02/09	jue 09/04/09		Projecte
12	<input checked="" type="checkbox"/>	Biblioteca d'aplicació	50 horas	vie 27/02/09	vie 03/04/09	8;9	Projecte
13	<input checked="" type="checkbox"/>	Esdeveniments	60 horas	jue 26/02/09	jue 09/04/09	8	Projecte
14	<input checked="" type="checkbox"/>	Interfície gràfica	35 horas	lun 23/02/09	mié 18/03/09	10	Projecte
15	<input checked="" type="checkbox"/>	Proves	20 horas	jue 09/04/09	jue 23/04/09	12;13;10	Projecte
16	<input checked="" type="checkbox"/>	Documentació	60 horas	jue 23/04/09	jue 04/06/09	15	Projecte

Imatge 13 - Taula d'organització inicial del projecte

7.1.2 Planificació inicial - Diagrama de Gantt

Acompanyant la informació anterior s'adjunta el diagrama de Gantt del projecte si s'hagués seguit la planificació inicial. Aquest mostra la distribució del projecte en el temps de forma gràfica, juntament amb el camí crític.



Imatge 14 - Diagrama de Gantt, primera divisió

7.1.3 Planificació final - Tasques

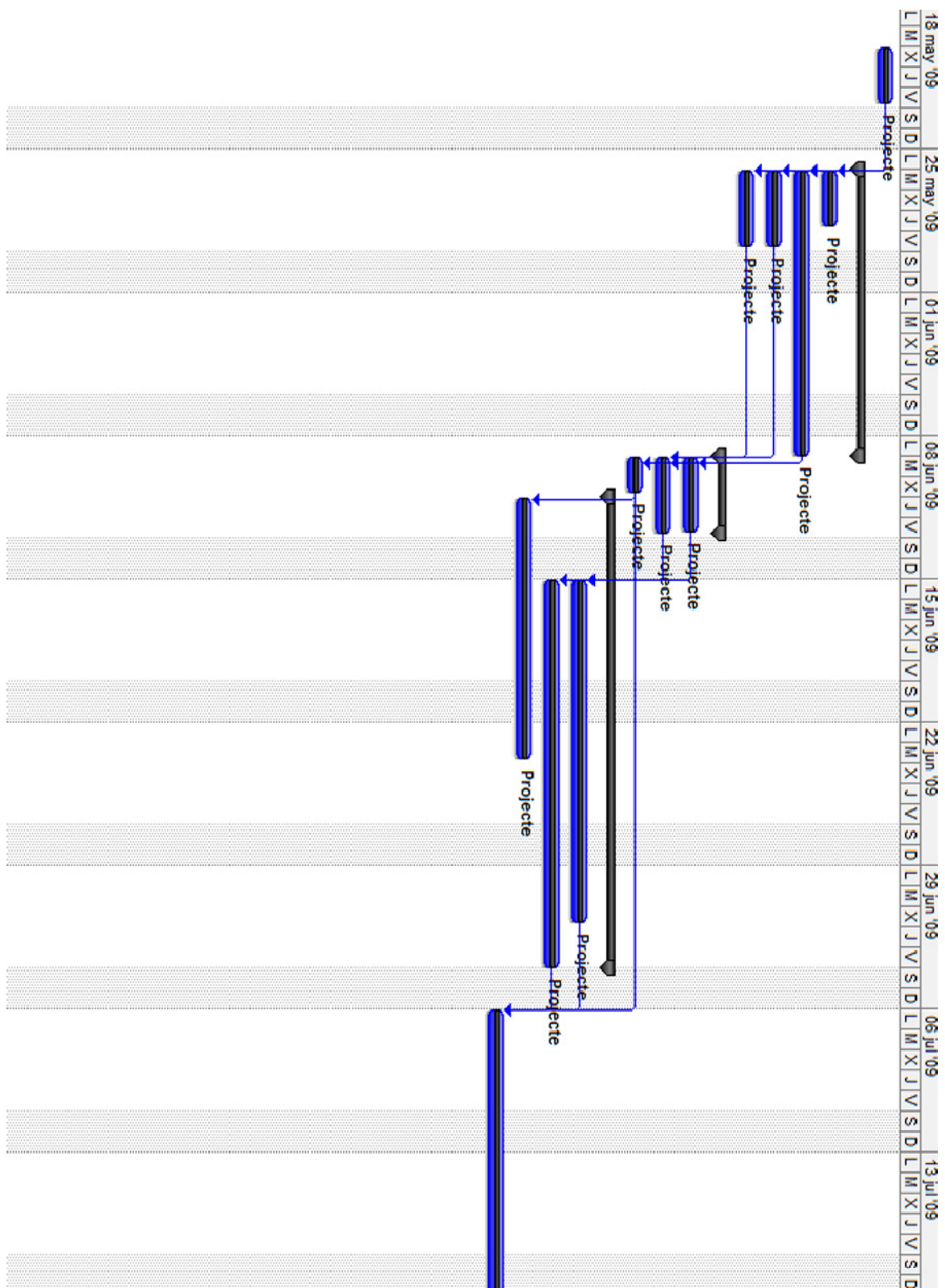
Degut a la manca de temps durant el transcurs del cicle escolar, la planificació per al desenvolupament del projecte no s'ha seguit de la forma esperada per lo que els seus horaris han estat modificats. A continuació es mostra una taula amb la llista de tasques desenvolupades amb les dates d'inici i fi reals.

	📌	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
1	✓	Anàlisis	10 horas	mié 20/05/09	vie 22/05/09		Projecte
2	✓	Investigació	10 días	mar 26/05/09	lun 08/06/09		Projecte
3	✓	Sistema eyeOS	10 horas	mar 26/05/09	jue 28/05/09	1	Projecte
4	✓	Toolkit eyeOS	40 horas	mar 26/05/09	lun 08/06/09	1	Projecte
5	✓	Llibreria GD	15 horas	mar 26/05/09	vie 29/05/09	1	Projecte
6	✓	JavaScript i PHP	15 horas	mar 26/05/09	vie 29/05/09	1	Projecte
7	✓	Disseny	3,75 días	mar 09/06/09	vie 12/06/09		Projecte
8	✓	Característiques d'eyeOS	13 horas	mar 09/06/09	vie 12/06/09	4	Projecte
9	✓	Llibreria d'aplicació	15 horas	mar 09/06/09	vie 12/06/09	4;5;6	Projecte
10	✓	Interfície gràfica	7 horas	mar 09/06/09	mié 10/06/09	4	Projecte
11	✓	Codificació	17 días	jue 11/06/09	vie 03/07/09		Projecte
12	✓	Biblioteca d'aplicació	50 horas	lun 15/06/09	mié 01/07/09	8;9	Projecte
13	✓	Esdeveniments	60 horas	lun 15/06/09	vie 03/07/09	8	Projecte
14	✓	Interfície gràfica	35 horas	jue 11/06/09	mar 23/06/09	10	Projecte
15	✓	Proves	60 horas	lun 06/07/09	vie 24/07/09	12;13;10	Projecte
16	✓	Documentació	70 horas	lun 27/07/09	mié 19/08/09	15	Projecte

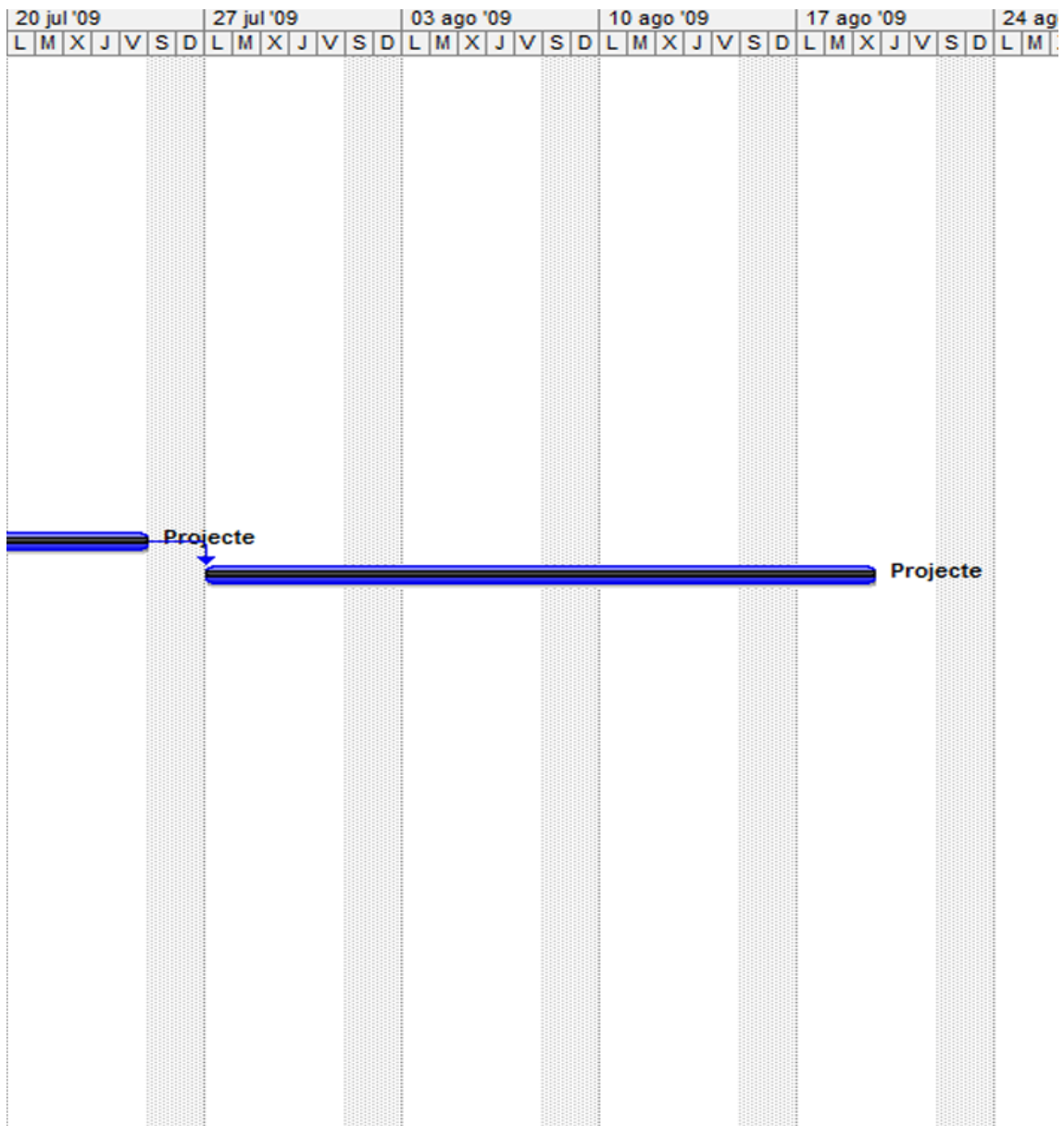
Imatge 16 - Taula d'organització final del projecte

7.1.4 Planificació final - Diagrama de Gantt

Tal com s'ha fet amb la planificació inicial del desenvolupament del projecte, a continuació es complementa la llista de tasques mitjançant el diagrama de Gantt de la distribució temporal final del projecte (*a posteriori*).



Imatge 17 - Diagrama de Gantt, primera divisió



Imatge 18 - Diagrama de Gantt, segona divisió

L'increment d'hores en la realització del projecte, concretament en el desenvolupament de la documentació, respecte la planificació inicial ve donat principalment per la voluntat de complementar l'aplicació amb una documentació que analitza i detalla de forma exhaustiva cadascun dels punts que conformen el procés de creació d'*eyePictures*. És per aquest motiu, que amb l'objectiu de realitzar una documentació a l'altura de l'aplicació s'han destinat més hores de les inicialment establertes.

2. Anàlisi de l'entorn

2.1 Historia

Cap a mitjans d'agost de 2005 un petit grup de joves emprenedors originaris de Barcelona comença a desenvolupar el que fins llavors era una idea sorgida gràcies a serveis d'emmagatzemament d'imatges digitals com *Flickr* o serveis d'emmagatzemament d'enllaços com *Del.licious.us*. La primera idea que sorgeix és la de crear un sistema operatiu web, tècnicament no ho és, amb nucli i "toolkit" propi el qual s'ofereixi lliurement a través d'Internet per que qualsevol persona pugui instal·lar-lo en els seus servidor i/o intranets. És llavors quan comença un rigorós treball de desenvolupament que dona com a resultat el que avui dia es coneix com *eyeOS*, un projecte català de programari lliure que ha acumulat 30.000 descàrregues en la seva versió per a servidors i 56.000 en la versió executable per xarxes d'àrea local amb *MS Windows*.

El març de 2006 decideixen posar en marxa un servei en línia gratuït, obrint així *eyeos.info*, basat en el seu programari original que ofereix un ús similar al d'un *Sistema operatiu* legítim al mateix temps que es brinda als usuaris la possibilitat d'emmagatzemar les seves dades de forma il·limitada.

Des de llavors, *eyeOS* no ha parat de créixer fins al punt que no només han creat projectes per a escoles i la Generalitat de Catalunya sinó que altres empreses han agafat el seu producte modificat per oferir els seus propis serveis.

2.1 eyeOS

eyeOS és un escriptori virtual multi plataforma, lliure i gratuït, que pretén emular l'aspecte que qualsevol sistema operatiu avui dia ofereix. Al ser un producte gratuït pot ser descarregat de forma directa des de la mateixa web d'*eyeOS* o bé es pot obtenir com a còpia de qualsevol de les branques que el repositori públic ofereix. *eyeOS* està format per una estructura de sistema Web 2.0,

desenvolupada en *HTML*, *PHP*, *JavaScript* i *AJAX*²⁷, que s'encarrega del correcte funcionament del propi sistema. Aquest ve acompanyat per un paquet d'aplicacions bàsic que es compon d'un conjunt d'aplicacions de tipus ofimàtiques com pot ser un potent processador de textos a l'estil *MS Word*, un conjunt d'aplicacions multimèdia, calendari, gestor d'arxius, un paquet d'entreteniment entre el que destaca un joc d'escacs, sistema de missatgeria intern i un senzill sistema d'instal·lació d'aplicacions externes. El paquet s'allibera sota la llicència lliure *AGPL (Affero General Public Licence)*, que permet la modificació del propi eyeOS per a poder comercialitzar-lo.

eyeOS suposa una revolució i un avenç al que es coneix com computació dins el "núvol", cosa en el que les potents empreses a nivell tecnològic estan començant a mostrar el seu interès fins al punt que ja no només podem trobar aplicacions web orientades a substituir les nostres aplicacions preferides d'ofimàtica, sinó que gegants empreses com el propi *Google* estan començant a interessar-se en el desenvolupament de sistemes similars com el ja conegut *Google Chrome OS*. Com ja s'ha comentat, la computació en el "núvol" s'entén com aquelles aplicacions que viuen a Internet i per tant permeten que l'usuari emmagatzemi les seves dades en elles deixant-les accessibles de forma immediata des de qualsevol punt del món sempre i quant es disposi d'una connexió al gran "núvol".

La principal virtut d'eyeOS és que es tracta d'un sistema, actualment, únic que no requereix la instal·lació de cap tipus d'utilitat externa per a poder ser utilitzat. Els seus únics requisits són disposar d'un ordinador amb connexió a la xarxa i navegador web.

2.3 Entorn

Al tractar-se d'un producte multi plataforma, eyeOS és capaç de funcionar en qualsevol sistema operatiu que compleixi una sèrie de requisits mínims tant de maquinari com de programari.

²⁷ *Asynchronous JavaScript And Xml (AJAX)*, un conjunt de tecnologies que permeten actualitzar continguts web sense haver de tornar a carregar la pàgina completa.

Els requisits a nivell de maquinari s'han de dividir en dos besants, requisits com a client d'eyeOS i requisits com a servidor. Com s'ha comentat anteriorment, els requisits per a poder accedir als serveis d'eyeOS com a client són mínims i només és necessari estar proveït d'un ordinador de sobretaula amb connexió a Internet. Per altra banda, si el que es vol és tenir accés a eyeOS mitjançant un servidor propi els requisits necessaris varien en funció del volum de clients als quals es vulgui donar accés.

Per a un servidor privat del que no s'espera tenir un volum de peticions gaire alt podríem definir els seus requisits mínims amb la següent llista:

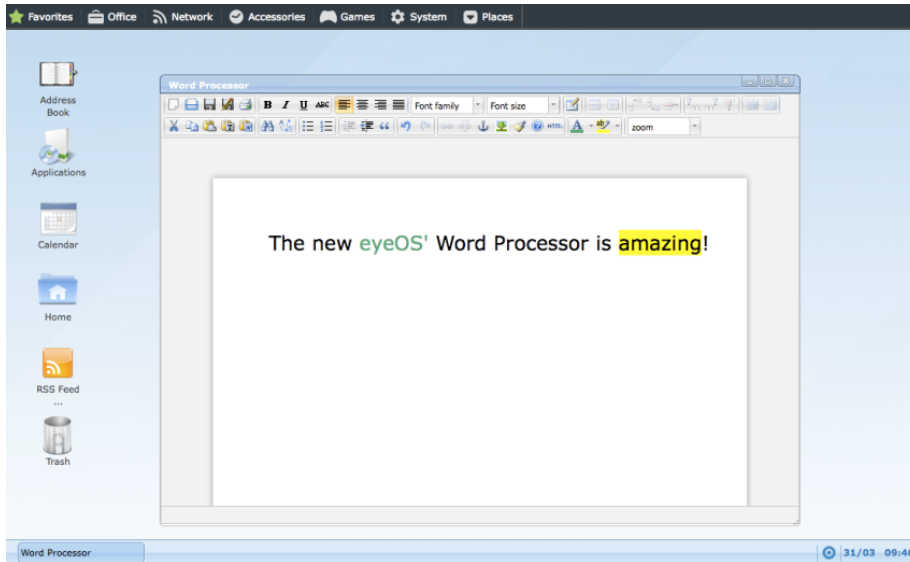
- Placa Asus, o equivalent, amb micro controladors Intel d'altas o mitjanes prestacions.
- Intel Quad Core 2.66GHz amb dissipador a 7.200 R.P.M.
- Dos discs durs Serial Ata II a 7.200 R.P.M.
- 4GB de memòria RAM DDR2.
- Tarja gràfica amb micro controlador Nvidia.

Mentre que les necessitats pel que fa a nivell de maquinari queden dividits en funció dels requeriments de l'usuari, a nivell de programari no hi ha cap tipus de distinció. L'aplicació busca un comportament homogeni en qualsevol sistema pel que garanteix un correcte funcionament independentment de la plataforma sempre i quant disposi dels serveis necessaris.

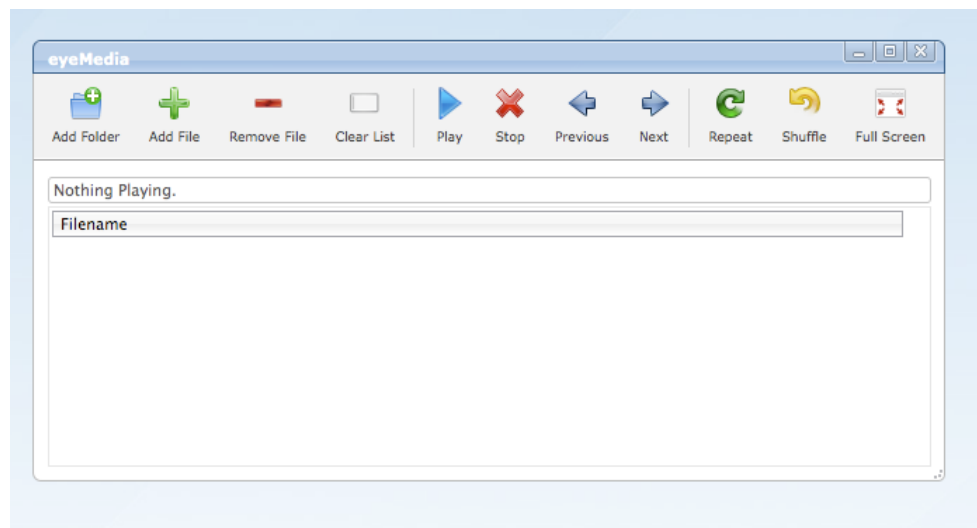
Aquests serveis bàsics per al correcte funcionament d'eyeOS són:

- Servidor web amb suport habilitat per al llenguatge *PHP* i algunes llibreries externes com ara l'*XML*.
- Suport per a *OpenOffice.org*, brindant així l'accés a contingut ofimàtic com són les fulles de càlcul, processador de texts i presentacions.

- Opcionalment es pot disposar d'un client de sistema de control de versions per a poder tenir la possibilitat de gaudir de les actualitzacions per a l'aplicació.



Imatge 19 - Escritori d'eyeOS amb el processador de textos en execució



Imatge 20 - Reproductor d'àudio d'eyeOS

3. Organització del projecte

Com a metodologia de planificació, desenvolupament i manteniment s'utilitzarà la MÈTRICA 3, pressuposada pel Ministeri d'Administracions Públiques d'Espanya.

ETAPES i METODOLOGÍA DE DESENVOLUPAMENT

1. Obtenir una llista de problemes, deficiències i millores del sistema actual.
2. Establir els requeriments funcionals del sistema.
3. Establir els requeriments no funcionals del sistema.
4. Estudi d'alternatives.
5. Planificació de les etapes de desenvolupament del projecte.
6. Recol·lecta de les eines i recursos programari per les tasques a realitzar.
7. Disseny lògic del projecte.
8. Codificació del disseny lògic.
9. Proves i posterior implementació.

PAUTES PER AL DESENVOLUPAMENT DE LA DOCUMENTACIÓ

1. Document de disseny d'alt nivell.
2. Document d'anàlisi de requeriments.
3. Document de planificació del projecte. Es proposa utilitzar alguna tècnica de planificació i control com és *MS Project*.
4. Anàlisi del funcionament del programari.

3.1 Especificació de requeriments

Un cop compresos els problemes i solucions, avantatges i les limitacions que implica el desenvolupament d'eines per a un entorn allotjat a Internet, entra en joc l'anàlisi de requeriments funcionals (aquells que descriuen el comportament desitjat del programari) i els no funcionals (restriccions imposades pel client o pel mateix problema i que afecten al disseny).

3.1.1 Requeriments funcionals

A continuació es llista i detalla el conjunt de requeriments funcionals que han anat sorgint durant el desenvolupament de l'aplicació. Cal remarcar que el còmput inicial de requeriments ha anat creixent durant el desenvolupament del programa.

El primer requeriment a tenir en compte és el de dissenyar una interfície gràfica senzilla i intuïtiva, que al mateix temps sigui eficaç i capaç de dotar a l'usuari de totes les eines necessàries per aconseguir una experiència agradable pel que a la visualització de fotografies respecte.

Per altra banda, un dels requeriments més importants, juntament amb l'esmentat anteriorment, és el del desenvolupament dels diferents modes de funcionament dels que disposa el programa (navegador d'arxius, visor d'imatges i presentació de diapositives) de forma que siguin usables i no interfereixin entre si.

- Mode navegador: Ha de permetre navegar pel sistema d'arxius, entrar i sortir dels directoris, a més de mostrar les imatges que aquests contenen de forma clara i ordenada, sense que aquests es barregin entre si. Per a complementar aquestes funcions, s'ha dissenyat una zona del navegador anomenada barra d'estat, on es combina el mostreig de la informació relativa a la ruta actual (nombre de carpetes, d'imatges i mida d'aquestes) amb la meta-informació d'imatges i la del contingut de directoris.

A banda de ser capaç d'oferir les funcionalitats mencionades anteriorment, ha de brindar la possibilitat d'interaccionar amb la resta de modes de funcionament.

- Mode visor d'imatges: Ha de possibilitar la visualització dels diferents formats d'imatge (jpeg, bmp, png i gif), avançar o retrocedir en el mostreig d'aquestes i rotar-les en qualsevol direcció. També aprofita l'espai que ens brinda la barra d'estat per a mostrar la meta-informació de la imatge en qüestió juntament amb el nom d'aquesta, cosa que fa aprofitant l'espai del gesto de finestres. A més d'oferir aquestes característiques ha de permetre entrar en el mode edició de colors, accessible únicament en situació visor d'imatges activat, i també canviar a qualsevol dels altres modes.
- Mode reproductor de diapositives: Permet a l'usuari mostrar les il·lustracions de forma automatitzada, avançant automàticament en la llista d'imatges amb un interval de 5 segons. Per descomptat, aquest mode és capaç d'interactuar amb la resta i parar la seva reproducció tant en el moment que l'usuari així ho decideix com quan decideix utilitzar qualsevol de les altres funcions de les que disposa l'aplicació.
- Mode editor de colors: Aquest mode disposa de 9 filtres fotogràfics professionals accessibles gràcies a la llibreria de *PHP* que té per nom *GD*. Aquest mode és només accessible amb el visor d'imatges habilitat i permet aplicar els filtres en qualsevol ordre i de forma recursiva tants cops com l'usuari desitgi, sempre reflexant els canvis de forma immediata. Per altra banda sempre podrà tornar a la imatge original, donat que els canvis només s'emmagatzemaran quan l'usuari ho demani.

3.1.2 Requeriments no funcionals

En aquest àmbit, s'ha de fer especial menció a les dificultats i limitacions imposades per les restriccions de disseny, les quals han fet del desenvolupament del programari un treball llarg i feixuc.

3.1.3 Requeriments de rendiment

Al ser una aplicació emmagatzemada en un servidor i a la qual s'accedeix a través del "núvol" s'ha hagut de tenir en compte aspectes com la velocitat d'execució i de resposta davant les interaccions amb l'usuari.

L'aplicació està pensada per a poder ser executada i accedida a través de qualsevol ordinador domèstic actual, sense que aquest requereixi unes característiques no convencionals.

3.1.4 Requeriments de disseny

El programa està desenvolupat de forma que es pugui executar amb total normalitat en els navegadors amb més índex de popularitat de l'actualitat que són: *Internet Explorer*, *Mozilla Firefox* i *Safari*. A més hauria de ser capaç de poder ser executat des de qualsevol altre navegador d'Internet però degut a la bifurcació d'estàndards això és quelcom que no es pot garantir ara per ara.

Per altra banda s'ha posat especial èmfasi en l'estabilitat, el que significa que al tractar-se d'una aplicació que es pot descarregar i instal·lar en qualsevol computadora amb diferents versions de programari, s'ha treballat per a que tingui un funcionament correcte, lliure d'errors i amb una lògica que el faci prou intuïtiu per a que qualsevol usuari sigui capaç d'utilitzar-lo sense cap tipus de problema.

3.1.5 Objectius de disseny

El programari ha d'estar escrit per a permetre futures actualitzacions i/o modificacions de forma que sigui possible incloure noves característiques juntament amb una senzilla correcció d'errors.

3.2 Eines de desenvolupament

Per tal d'obtenir el prototip que es presenta com a projecte de final de carrera, s'ha hagut de fer ús d'un conjunt d'eines de desenvolupament les quals es descriuen a continuació.

3.2.1 GNU/Gentoo Linux

GNU/Gentoo Linux es considera un "sabor" especial de *GNU/Linux* el qual pot ser optimitzat, personalitzat i configurat per a qualsevol tipus d'aplicació i/o necessitat.

Quan es parla de *Linux*, s'ha de tenir clar, que es referencia de forma exclusiva el nucli (*kernel*) dels *sistema operatiu*, però és utilitzat de forma comuna per descriure un *Sistema operatiu* complet tipus *Unix*, amb la peculiaritat que *Linux* utilitza primordialment filosofies i metodologies lliures. Aquest és també conegut com *GNU/Linux*, i està format per la combinació del nucli *Linux* amb biblioteques i eines del projecte *GNU* juntament amb molts altres projectes i grups de programari, lliure o propietari. El nucli no forma part del projecte oficial *GNU*. Aquest compte amb el seu propi nucli anomenat *Hurd* i és distribuït sota la llicència *GPL*²⁸.

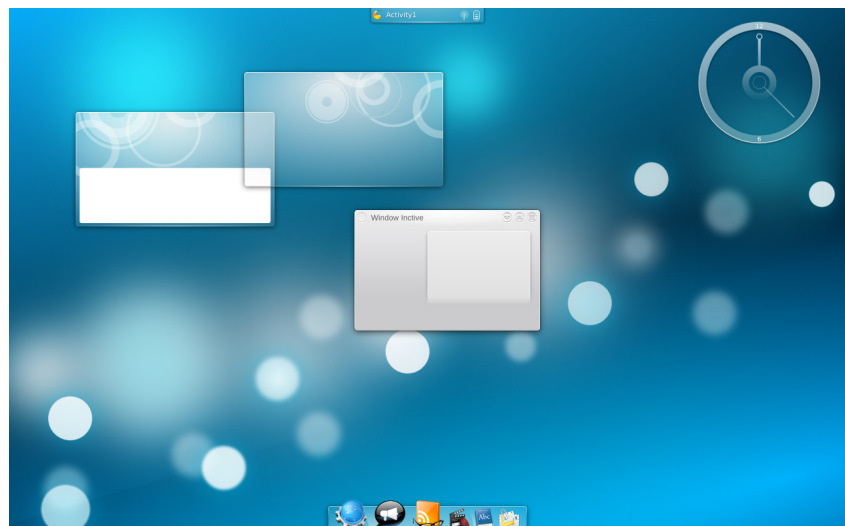
Linux apareix a principis de la dècada dels noranta, concretament al 1991. Es tracta d'una versió lliure de *Unix* desenvolupada originalment per un estudiant d'informàtica de la Universitat de Helsinki, Finlàndia, anomenat *Linus Benedict*

²⁸ **General Public License (GPL)**, és un tipus de llicència per a programari que permet la còpia, distribució (comercial o no) i modificació del codi font, sempre que qualsevol modificació continuï sent distribuïda sota la pròpia llicència GPL. La llicència GPL no permet la distribució de programes executables sense el codi font corresponent o en la seva variant, mitjançant una fórmula d'obtenció gratuïta.

Torvalds el qual comença el seu desenvolupament per afició sense imaginar el futur que li esperaria. Basat originalment en *Minix*, un clon de *Unix* distribuït amb el seu codi font, i desenvolupat pràcticament en la seva totalitat en els llenguatges *C#* i *C++* juntament amb algunes extensions de *GNU C* és avui dia un dels sistemes operatius més utilitzats. Suporta pràcticament totes les arquitectures actuals, s'adapta a funcionalitats tant diferents com les d'escriptori i servidor oferint els conjunts d'eines més potents del moment en qualsevol de les seves versions.

Segons paraules textuais del seu creador, *GNU/Gentoo Linux* neix partint del temps lliure. Temps dedicat a explorar, descobrir i experimentar. Així és com *Daniel Robbins*, el seu creador, executa les seves primeres passes en el món de *GNU/Linux*. La primera distribució²⁹ que va utilitzar va ser *GNU/Debian Linux* amb la qual es va dedicar, únicament, a configurar algunes aplicacions abans de trobar-se amb les avantatges i desavantatges del sistema. Per aquest motiu va decidir canviar a *GNU/Stampede Linux* on ràpidament es va integrar com a desenvolupador fins al moment en que va decidir començar a treballar en el seu propi gestor de paquets. Després d'un breu període de temps decidí deixar l'equip de desenvolupadors per crear la seva pròpia distribució, així va néixer *Enoch* que més tard rebria el nom de *GNU/Gentoo Linux*.

Unes de les característiques principals de *Gentoo* radica en l'ús d'un sistema de gestió de paquets amb reminiscències del model de "ports" de



Imatge 21 - Escriptori KDE4 a GNU/Gentoo Linux

²⁹ Conegut com l'acció o efecte de distribuir, es coneix en la informàtica com el mateix conjunt d'eines que formen un sistema GNU/Linux creat per diferents empreses.

*BSD*³⁰. Amb *Gentoo* és possible crear un sistema *GNU/Linux* totalment funcional únicament partint del codi font, el qual des d'un inici permet la seva compilació mitjançant paràmetres d'optimització a gust de l'usuari. *Gentoo* també brinda el més complet control sobre els paquets a instal·lar, d'aquesta forma *Gentoo* permet de forma única un control d'instal·lació excepcional que s'adapta a tot tipus d'usuaris i objectius. És per el seu desenvolupament excepcional, la seva adaptabilitat i l'alt nivell, pel que a coneixements respecte, que la comunitat d'usuaris posseeix que fan a única aquesta distribució.

Per tots aquests motius, i d'altres, que s'ha decidit utilitzar aquesta distribució, com a *Sistema operatiu*, per al desenvolupament d'*eyePictures*.

3.2.2 Subversion

Subversion és un programari de sistema de control de versions dissenyat única i específicament per suplir i substituir les carències l'antic *CVS*. La seva funció principal és la de mantenir versions actuals dels arxius juntament amb un històric d'aquests. D'aquesta forma els arxius poden ser modificats, actualitzats o retornats a la seva versió antiga per múltiples persones germanitzant cert ordre en totes aquestes modificacions i permetent un accés públic a tots aquests canvis. D'aquesta forma, el també anomenat *svn*, permet la creació i posterior manteniment d'arxius de codi font, pàgines web, documentació, etc.

Les seves característiques principals son:

- Segueix l'històric de fitxers i directoris a través de còpies i reanomenaments.
- Realitza modificacions, incloent canvis a arxius simultàniament, de forma atòmica³¹.
- Creació de branques i etiquetes, operació molt eficient que permet tenir diferents versions d'un o mes arxius de forma paral·lela.

³⁰ **Berkely Software Distribution (BSD)**, són les sigles que s'utilitzen per designar un sistema operatiu desenvolupat per la universitat de *Berkely* basat en *Unix*.

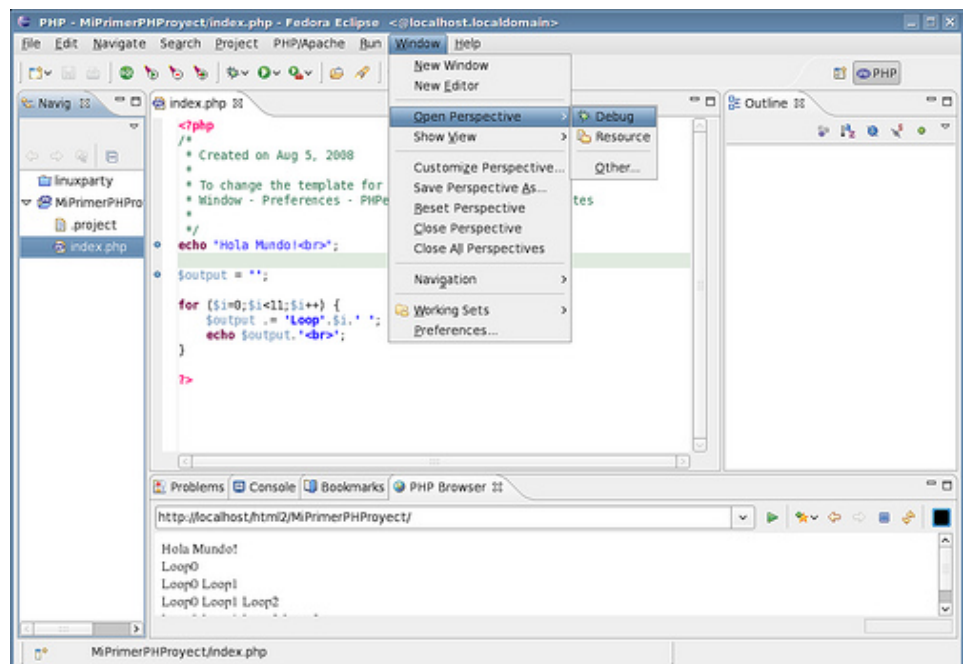
³¹ Fa referència a aquelles operacions capaces de dur-se a terme sense passes intermèdies.

- Al enviar o rebre modificacions d'arxius treballa únicament amb les diferències entre aquests.
- Permet l'accés a través d'Apache, utilitzant *WebDAV/DeltaV*, de forma transparent.
- Maneig eficient de binaris.
- Permet el bloqueig d'arxius de forma selectiva, útil per a que arxius binaris no siguin editats per més d'una persona al mateix temps.

3.2.3 Eclipse Platform

Al contrari del que molta gent pensa, la plataforma *Eclipse* és una eina per tot, i per res en concret. És una eina dissenyada i pensada per a la construcció d'entorns de desenvolupament (*IDE*) que puguin ser utilitzats per a la construcció d'aplicacions web, aplicacions *Java* de tot tipus, programes *C++* i *Enterprise JavaBeans (EJBS)*.

L'entorn *Eclipse* permet ampliar les seves funcionalitats mitjançant components integrables anomenats *plugins*, els quals estan escrits en *Java*. La plataforma *Eclipse* té una interfície de desenvolupament



Imatge 22 - Entorn de desenvolupament Eclipse Platform amb PHPclipse habilitat

construïda sobre un entorn de treball que proporciona tota la estructura i que presenta una interfície d'usuari. D'aquesta forma es pot dir que *Eclipse* no proporciona un entorn de treball sinó que és la base d'un conjunt de *plugins* que permeten el desenvolupament de futures aplicacions. Per tant, per al desenvolupament d'aquesta aplicació s'ha utilitzat *Eclipse Platform*, en la versió *Ganymede*, amb un conjunt de *plugins* que han permès el desenvolupament de l'aplicació en si.

3.2.4 PHPclipse

PHPclipse és un *plugin* per a *Eclipse Platform* que exten les seves funcionalitats proporcionant un entorn de desenvolupament adequat per al llenguatge *PHP*. Algunes de les seves característiques principals són:

- Habilita la creació de projectes i arxius *PHP*.
- Acoloreix el codi per a una ràpida identificació i lectura de les seves parts.
- Auto completament de codi.
- Ofereix i habilita la creació de plantilles de codi *PHP*.
- Ofereix ajuda de programació en temps real.

3.2.5 Subclipse

Subclipse és un *plugin* per a *Eclipse Platform* que integra dins l'entorn de desenvolupament les funcionalitats d'un client de control de versions, de forma que es poden modificar els arxius, ja sigui mitjançant actualitzacions, enviant els canvis fets fins al moment o tornant a versions anteriors, des de la mateixa interfície gràfica evitant així l'accés a l'entorn de comandes.

3.2.6 Mozilla Firefox

Mozilla Firefox és un navegador d'Internet, lliure i de còdi obert, descendent de *Mozilla Application Suite*. Per tant, desenvolupat per *Mozilla*, la seva pròpia fundació i un gran nombre de voluntaris.

El projecte *Firefox* va començar com una branca experimental del projecte *Mozilla*, a càrrec de *Dave Hyatt* i *Blake Ross*. La idea inicial era la d'eliminar totes les funcionalitats, que fins llavors incorporava *Netscape*, que no fossin pròpiament necessàries en un navegador web. D'aquesta manera apareix al mercat el projecte *Mozilla/Browser* sota el nom de *Phoenix*. Amb el temps *Phoenix* va continuar el seu desenvolupament adoptant el nom de *Firebird*, degut a assumptes legals.

Actualment el projecte es coneix amb el nom de *Firefox*, es tracta d'un projecte multi plataforma distribuït sota la triple llicència *GPL/LGPL/MPL*. Compte amb el 22.7% del mercat total de navegadors web, per sota d'*Internet Explorer*. Per al renderitzat de les pàgines web, *Firefox* fa ús del motor *Gecko*³² que implementa un conjunt d'estàndards web actuals a més d'altres, els quals estan enfocats a anticipar-se a possibles futures adicions .

Algunes de les seves principals característiques són:

- Ser el primer navegador en implementar la navegació per pestanyes.
- Incloure corrector ortogràfic.
- Disposar de busca interactiva progressiva.
- Marcadors dinàmics.
- Integració de motors d busca, com el popular *Google*.
- Adició de programes de terceres persones mitjançant extensions.
- Alt índex de personalització i optimització adaptant-se a objectius.

³² *Gecko*, és un motor de renderitzat, originalment desenvolupat per *Netscape* i escrit en *C++*.

3.2.7 Firebug

Firebug és una extensió de *Firefox* creada i dissenyada especialment per desenvolupadors i programadors web. Consta d'un petit paquet d'utilitats amb el qual es pot analitzar (revisar la velocitat de càrrega i d'estructura *DOM*³³), editar, monitoritzar i depurar el codi font *CSS*, *HTML* i *JavaScript* d'una pàgina web de forma instantània.

Firebug no és un simple inspector *DOM*, sinó que també permet emmagatzemar els canvis realitzats. Consta d'una interfície gràfica molt intuïtiva la qual es serveix de pestanyes específiques per a l'anàlisi de cada element (consola, *HTML*, *CSS*, *Script*, *DOM* i xarxa). *Firebug* és de distribució gratuïta, ve encapsulat en forma de *plugin* i permet la instal·lació d'extensions per ampliar el seu propi funcionament.



Imatge 23 - Integració de Firebug en Mozilla Firefox

³³ **Document Object Model (DOM)**, és essencialment una interfície de programació d'aplicacions que proporciona un conjunt estàndard d'objectes per representar documents *HTML* i *XML*. A través de *DOM* les aplicacions són capaces d'accedir i modificar estructures de documents *HTML* i *XML*.

3.2.8 Firecookie

Firecookie és una extensió de *Firebug* que permet administrar i analitzar les galetes de sessió des del mateix *Firefox*.

3.2.9 Apache

Apache és un servidor HTTP, per tant servidor web, multi plataforma i de codi obert. Com a proveïdor de serveis web s'encarrega d'atendre peticions de pàgines web i de respondre-les amb el contingut adequat. El seu desenvolupament basat en, el llavors, popular codi de *NCSA HTTPd 1.3* es va posar en marxa l'any 1995. Des de llavors *Apache* s'ha seguit desenvolupant de forma continuada, per la *Apache Software Foundation*, fins arribar a convertir-se en un dels estàndards de facto en quant a servidors web escau.

Tot i que existeixen múltiples aplicacions capaces de proporcionar aquest serveis, *Apache* compta amb una àmplia acceptació en el món informàtic, tanta que al 2005 va arribar a aconseguir la seva màxima quota de mercat comptant amb un 70% d'ús en el còmput mundial de servidors web en ús. Això demostra que es tracta d'un servidor d'altres prestacions amb un alt índex de flexibilitat, el qual permet adaptar-lo a tots els gustos i necessitats. Per a aconseguir-ho posa a disposició de l'usuari un conjunt de mòduls i eines capaces d'integrar-lo en qualsevol entorn i resoldre possibles conflictes amb relativa facilitat.

És per aquests motius esmentats anteriorment que s'ha triat *Apache* com una de les eines imprescindibles per al desenvolupament del projecte.

4. Fonaments teòrics

4.1 Llenguatges i tecnologies utilitzats

Per al desenvolupament d'*eyePictures* s'ha utilitzat un conjunt de diferents tecnologies les quals es detallen a continuació. Abans de continuar amb la documentació del conjunt de llenguatges utilitzats cal mencionar que degut a que l'aplicació desenvolupada s'incorpora a un producte madur amb un camí marcat, no han existit possibilitats reals de triar els llenguatges de desenvolupament. Deixant això a banda, cal remarcar que l'ús de cadascun dels llenguatges utilitzats continua sent un encert ja que es tracta de llenguatges de gran potencial i que brinden infinites possibilitats d'ús, son lliures, es troben sempre en constant desenvolupament i estan àmpliament estesos en el seus respectius sectors, tots ells enfocats a l'entorn de desenvolupament web.

4.1.1 PHP

Hipertext Preprocessor (PHP), és un llenguatge de programació interpretat normalment utilitzat per a implementació i generació de pàgines web dinàmiques. Es tracta d'un llenguatge que s'executa a la banda del servidor, per tant, és l'usuari qui realitza les peticions, ja sigui fent clic sobre algun enllaç o bé obrint una nova adreça web, i aquest respon a les peticions en format *HTML* per tal que navegador web pugui mostrar la pàgina al client. És aquest motiu el que impossibilita la visualització del codi *PHP*. Tot i que el seu ús està clarament orientat a la generació de webs dinàmiques, també pot ser usat per implementar interfícies de comandes i/o aplicacions executables amb interfície gràfica. Es tracta d'un llenguatge multi plataforma, ja que l'interpret que es troba al costat del servidor *HTML* es distribueix en diferents formats executables.

PHP tot i ser programari lliure, es distribueix sota la llicència *PHP*, la qual és incompatible amb la *GPL*.

Algunes de les seves característiques son:

- *PHP* és extremadament modularitzat. En tipus de dades, sintaxi i funcions s'assembla molt al llenguatge de programació *C#*. I des de la versió 5, la qual inclou major suport per a l'orientació a objectes, s'assembla també al *C++*.
- *PHP* pot ser inclòs dins el codi *HTML* indicant-ho a l'interpret mitjançant les etiquetes d'inici "<?" i fi "?>" o respectivament "<?php" i "?>". Per tant quan el servidor *HTML* interpreta codi situat entre les etiquetes d'inici i fi de *PHP* l'envia al interpret d'aquest, de forma que es genera en *HTML* "barrejat" amb el codi *PHP* inclòs.
- *PHP*, com a llenguatge nascut de i per Internet, proporciona un ampli conjunt de funcions tot tipus. Proporciona funcions de llibreria per a connectivitat de xarxa, desenvolupament de codi web, i interacció amb bases de dades. Alguns exemples de la varietat que ofereix en quant a connectivitat de bases de dades són: *Oracle*, *PostgreSQL*, *MySQL*, *SQLite*, *MSSQL*. També ofereix gran varietat de funcions de llibreria per a la interconnexió amb diferents protocols, a continuació es llisten alguns d'aquests: *MAP*, *SNMP*, *NNTP*, *POP3*, *HTTP*, *LDAP*, *XML-RPC*, etc.

4.1.2 Llibreria *GD*

Com s'ha comentat anteriorment, *PHP* està dissenyat amb l'objectiu de ser utilitzat per al desenvolupament web. Aquest motiu ha fet que per a la creació d'imatges, en miniatura o no, en el navegador s'hagi hagut de recórrer a l'ús de llibreries externes de *PHP*. Per a la creació i tractament d'il·lustracions, existeixen diferents llibreries, les principals són: *GD*, *Imagemagick*, *Gmagick* i *Exif*. En aquest cas s'ha optat per *GD* ja que és una llibreria que compte amb

alt índex de popularitat i ofereix un gran rendiment en tots els seus àmbits. *GD* ofereix suport per un ampli rang de formats d'imatge com ara: *JPEG, GIF, PNG, SWF, TIFF, JPEG2000, WBMP, i XPM*.

Com s'ha comentant anteriorment, aquesta llibreria no forma part de les funcions integrades a *PHP*, sinó que és una opció desenvolupada de forma independent (per *Thomas Boutell*) que els responsables dels servidors poden o no instal·lar de forma que *PHP* sigui capaç d'entendre i interpretar les seves funcions.

En la següent caixa de text es mostra l'ús de la llibreria *GD* per a la creació d'una imatge amb fons transparent.

```
$thumbnail = imagecreatetruecolor($width,$height);
imagealphablending($thumbnail,0);

imagefill($thumbnail,0,0,imagecolorallocatealpha($thumbnail,0,
0,0,255));
imagesavealpha($thumbnail,1);
```

4.1.3 JavaScript

JavaScript és un llenguatge de programació interpretat, per tant no necessita ser compilat. Utilitzat principalment en pàgines web s'executa directament en el navegador, al costat del client, i per tant no requereix cap tipus de consulta ni resposta per part del servidor *HTTP*. *JavaScript* és un llenguatge orientat a objectes, ja que disposa d'herència, i esdeveniments. Segueix el paradigma de programació basada en prototips. Qualsevol explorador d'Internet modern és capaç d'interpretar *JavaScript*. Per a fer-ho es proveeix a *JavaScript* d'una implementació del *DOM*.

A continuació es mostra l'execució de codi *JavaScript* dins de codi *PHP*, utilitzant la llibreria *eyeX* i el mètode *rawjs*.

```
eyeX('rawjs',array('js' => '
    if (slideShow == 1) {
        cronoid = setTimeout("sendMsg(' . $checknum .
',\"slideShowJS\",eyeParam(\"view\",1))",0);
    }
'));
```

4.1.4 CSS

Els fulls d'estil en cascada (*Cascading Style Sheets*), és un llenguatge formal usat per a definir la presentació d'un document estructurat, escrit en *HTML* o *XML*. Per tal de garantir una sèrie d'estàndards, *W3C*³⁴ s'encarrega de formular l'especificació dels fulls d'estil que servirà d'estàndard per als agents d'usuari o navegadors.

El paradigma de desenvolupament CSS consisteix en separar l'estructura d'un document de la seva presentació. Per exemple, l'element d'*HTML* `<H1>` indica que un bloc de text és un encapçalament i que és més important que un bloc etiquetat com `<H2>`. Versions més antigues d'*HTML* permetien atributs extra dins l'etiqueta oberta per donar-li format (com el color o la mida de font). No obstant això, cada etiqueta havia de disposar d'aquesta informació si es desitjava un disseny consistent per a una pàgina, i a més, una persona que llegís aquesta pàgina amb un navegador, perdia totalment el control sobre la visualització del text. Quan s'utilitza CSS, l'etiqueta `<H1>` no hauria de proporcionar informació sobre com serà visualitzat, només marca l'estructura del document. La informació d'estil separada en una fulla d'estil específica com s'ha de mostrar `<H1>`: color, font, alineació del text, grandària, i altres característiques no visuals com definir el volum d'un sintetitzador de veu, per exemple.

La informació d'estil pot ser adjuntada tant com un document separat o en el mateix document *HTML*. En aquest document, podrien definir-se estils generals en la capçalera del document o en cada etiqueta particular mitjançant l'atribut "style".

Els avantatges d'utilitzar CSS, o un altre llenguatge de maquetació d'estil, són:

- Control centralitzat de la presentació d'un portal web complet, cosa que permet agilitzar l'actualització i desenvolupament del mateix.

³⁴ **World Wide Web Consortium (W3C)**, és el consorci internacional que treballa per a desenvolupar i promoure estàndards per al Web. El dirigeix Tim Berners-Lee, creador del WWW i autor de les especificacions de l'URL, l'HTTP i l'HTML, que són les seves principals tecnologies d'aquesta xarxa.

- Els navegadors permeten als usuaris especificar el seu propi full d'estil local que serà aplicat a un lloc web remot, amb el que augmenta considerablement l'accessibilitat i la personalització. Per exemple, persones amb deficiències visuals poden configurar el seu propi full d'estil per a augmentar la grandària del text.
- Una pàgina pot disposar de diferents fulles d'estil segons el dispositiu que la mostri o fins i tot a elecció de l'usuari. D'aquesta forma és possible tenir fulls d'estil per a impressió, visualització en dispositius mòbils, ser "llegida" per un sintetitzador de veu, etc.
- El document *HTML* en si mateix és més clar d'entendre i s'aconsegueix reduir considerablement la seva grandària.

En el següent exemple es mostren una sèrie de propietats CSS juntament amb els seus valors, continguts en un vector i aplicats per mitjà del mètode `setCSS` del "*toolkit*" d'*eyeOS*.

```
$mainBox->setCSS(array(  
    'background-color' => '#666666',  
    'border-top' => '1px solid #aaaaaa'  
));
```

4.1.5 Toolkit d'*eyeOS*

eyeOS està creat sobre del seu propi "*toolkit*" el qual interactua amb el sistema per mitjà d'un *framework* propi. D'aquesta manera *eyeOS* proporciona un sistema de programació orientat a objectes molt intuïtiu, amb una gran varietat d'objectes, els quals reben el nom de widget³⁵, i amb un conjunt de llibreries i serveis que permeten la creació d'aplicacions interactuant amb llenguatges *CSS*, *JavaScript* i *PHP*.

³⁵ **Widget** és el nom que reben els objectes en el món del desenvolupament informàtic. Un clar exemple és l'objecte botó.

Una aplicació completament funcional a eyeOS, consta de com a mínim dos fitxers, *app.eyecode* i *events.eyecode*. El primer emmagatzema tot el contingut visual de l'aplicació, com podria ser el codi referent a la creació de finestres, botons, icones, etc. El segon emmagatzema tot allò relacionat amb la creació d'esdeveniments com per exemple la part del codi relativa a fer clic sobre un botó, al redimensionar l'aplicació, etc. Opcionalment les aplicacions d'eyeOS poden constar d'altres arxius específics com és l'arxiu *com.eyecode*, el qual s'encarrega de notificar a la pròpia aplicació de successos externs a aquesta. Per altra banda es pot incloure codi provinent d'altres arxius mitjançant la sentència *PHP "include"*.

El "*toolkit*" eyeOS està pròpiament format per un conjunt de *widgets*, els quals no són més que una sèrie de vectors amb propietats específiques que són executats mitjançant crides a classes *PHP* amb *AJAX*, de forma que és totalment transparent i instantani per l'usuari. Quan succeeix algun esdeveniment, el sistema s'encarrega de buscar el codi relatiu a dita circumstància provinent del fitxer *events.eyecode* i executar-lo sense refrescar el navegador.

eyeOS posa a disposició del programador un conjunt de llibreries, *widgets* i serveis que permeten la creació d'objectes i l'administració tant dels fitxers emmagatzemats al sistema d'arxius, anomenats fitxers reals, com la dels fitxers visualitzats per l'usuari i que reben el nom de fitxers abstractes o virtuals. Mitjançant aquest conjunt de serveis i llibreries el programador pot interactuar amb el sistema d'arxius, amb altres aplicacions o bé amb el propi sistema eyeOS.

A continuació es mostra una estructura típica per a la creació d'un *widget*. Com es pot veure no es més que un vector del tipus "índex" => "valor", on índex fa referència a una de les propietats disponibles.

```
$mainBox = new Simplebox(array(  
    'name' => 'eyeMainBox',  
    'father' => 'eyePictures_MainWnd_Content',  
    'x' => 0,  
    'y' => 59,  
    'width' => $window->width - 2,  
));  
$mainBox->show(0);
```

4.2 Formats d'imatge

A continuació es detallen els formats d'imatge amb els quals s'ha treballat per mitjà de la llibreria *GD* per permetre la seva manipulació i posterior visualització a través d'*eyePictures*.

4.2.1 JPEG

Joint Photographic Experts Group (JPEG), és un format d'imatge estàndard basat en un algorisme dissenyat per a comprimir imatges estacionàries amb 24 bits de profunditat o en escala de grisos. També pot ser, incorrectament, conegut com *JPG* a causa de l'extensió que té en sistemes operatius que només accepten tres lletres en la seva extensió de format d'arxius.

El *JPEG* és un algorisme de compressió d'imatge amb pèrdua d'informació. Això significa que en descomprimir la imatge no s'obté exactament la imatge anterior a la compressió. Una de les característiques que fan molt flexible el *JPEG* és la possibilitat d'ajustar el grau de compressió. Així doncs, el *JPEG* inclou diversos procediments de compressió d'imatge i pot considerar-se com un conjunt d'eines per a la compressió d'aquestes. Alguns esquemes del *JPEG* s'han fet molt populars i s'utilitzen en aplicacions informàtiques i d'electrònica de consum. Els mètodes proposats s'orienten a la codificació d'imatges multi nivell tant monocromes com de color. Per la codificació d'imatges binàries existeix l'estàndard *JBIG*.

4.2.2 BMP

L'anomena't *BMP* (en anglès, *Bit-map*), és l'estàndard per a sistemes operatius de *Microsoft* i basa el seu algorisme en mapes de bits on cada píxel té associat un nombre determinat de bits per a representar el color que conté. De fet, en funció d'això, podem distingir diversos formats: blanc i negre (1 bit), 16 colors (4 bits), 256 colors (8 bits), 16 milions de colors (24 bits), etc.

L'estructura interna dels fitxers BMP és la següent:

- Capçalera de l'arxiu
- Capçalera d'informació
- Paleta de colors (només per a arxius de 8 bits)
- Dades

Encara que aquest format permet la compressió, no se sol usar ja que hi ha altres tipus de fitxers gràfics que permeten una relació qualitat/compressió molt millor, com és el cas del ja explicat *JPEG* o del *PNG*.

El principal avantatge d'aquest tipus de fitxers és que la seva càrrega i lectura és extremadament ràpida, és capaç d'aconseguir una profunditat de color de 24 bits (aproximadament 16,7 milions de colors).

L'únic problema d'aquest format és el fet de no aparèixer habitualment en la seva versió comprimida, i per tant, la seva mida el fa inviable per al seu ús en entorns com per exemple Internet, on la mida de les imatges és fonamental.

4.2.3 GIF

Graphics Interchange Format (GIF) és un dels formats per a la compressió d'arxius d'imatge més populars a Internet, el qual va ser desenvolupat per *CompuServe*.

Es tracta, d'un esquema de compressió sense pèrdua per a reduir la grandària de la imatge, basat en una variant de l'algorisme *Lempel Ziv Welch* propietat de *Unisys Corporation*. Aquest sistema funciona especialment bé en imatges amb zones de color homogeni, és a dir, quants menys colors té la imatge millor funciona l'esquema de compressió.

Existeixen diverses versions, però les més emprades són el *GIF87A* i el *GIF89A*. El format inicial suportava fins a 256 colors (8 bits). Aquest estàndard fou revisat l'any 1989, donant com a resultat un de nou que permet emmagatzemar més d'una imatge, la qual cosa permet la creació d'animacions

senzilles. A més, permet definir un color com a transparent. El color de transparència pot originar problemes si s'utilitzen tècniques de suavitzat, ja que, a priori, la imatge no coneix el color que hi ha a sota i, per tant, no pot fer el suavitzat adequadament.

Un dels seus problemes afegits és que durant molt anys va ser un format patentat i qualsevol programador que volia fer-ne ús havia de pagar a la companyia creadora tant del format com de l'algorisme que empra.

4.2.4 PNG

Portable Network Graphics (PNG), és un tipus de format d'il·lustracions pensat especialment per a pàgines web, desenvolupat pel W3C amb la intenció de substituir el format *GIF*, el qual era subjecte a patents.

Les seves principals característiques són:

- Escala de grisos: 1, 2, 4, 8 16 bits
- Color real: 24, 48 bits
- Imatges basades en paleta: 1, 2, 4, 8 bits
- Transparència alfa: 256 (8 bits) o 65.636 (16 bits) nivells de transparència
- Transparència binària
- Entrellaçat 2D
- Correcció gamma
- Algorisme de compressió sense pèrdua i, a més a més, codi obert

Tot i que no ha sigut àmpliament acollit degut que ja hi havia d'altres formats fortament implantats, aporta uns avantatges notables sobre els altres dos formats més usats a Internet.

Vegem a continuació una taula comparativa:

	JPEG	GIF	PNG
Transparència	No	Sí	Sí, amb nivells de transparència (canal alfa).
Compressió	Amb pèrdua	Sense pèrdua	Sense pèrdua, i d'un 5 a un 25% menor que <i>GIF</i> .
Acceptació	Àmpliament acceptat	Àmpliament acceptat però en desús	Encara no el suporten correctament tots els navegadors.
Colors	24 bits	8 bits	24 bits
Animació	No	Sí	No, però hi ha un format basat en <i>PNG</i> que sí que ho suporta.

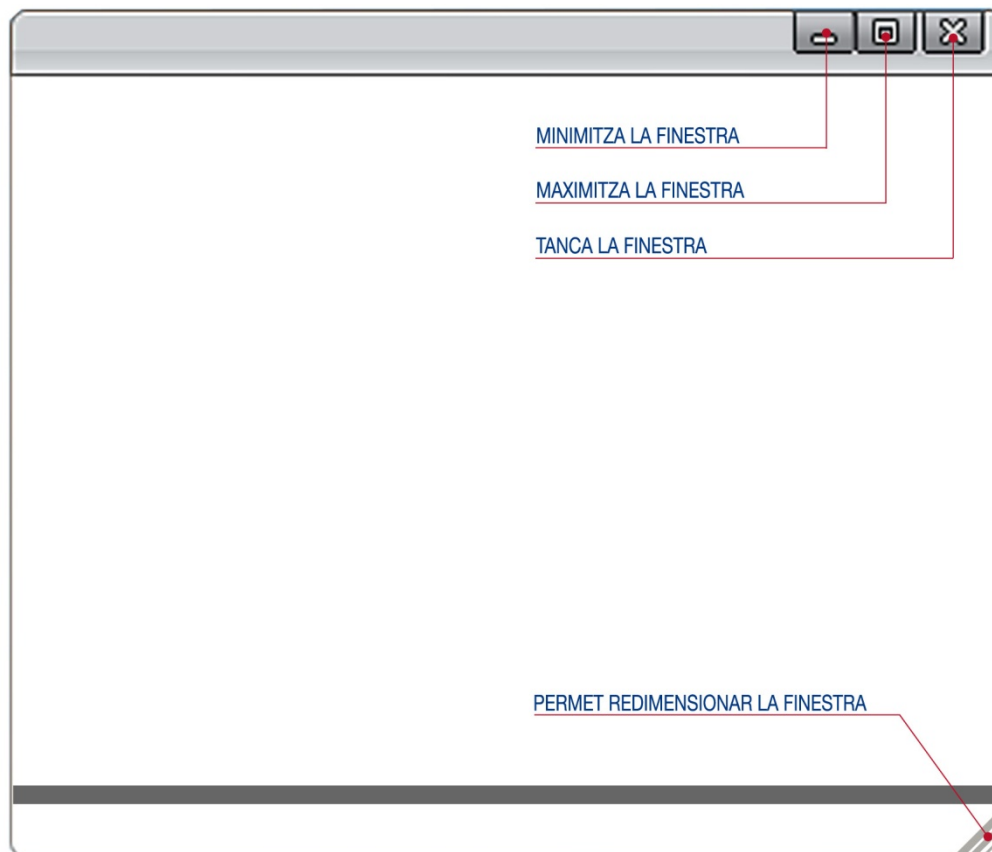
5. Anàlisi funcional

En aquest apartat es detalla el comportament de l'aplicació la qual es complementa per mitjà d'il·lustracions amb les seves respectives explicacions.

5.1 Presentació de l'aplicació

L'aplicació consta de dos modes principals de funcionament, explorador d'arxius i visor d'imatges. Ambdós funcionalitats comparteixen una sola interfície gràfica sobre la qual es modifica el seu contingut però es manté la mateixa lògica. Això converteix *eyePictures* en una aplicació molt intuïtiva i potent al mateix temps.

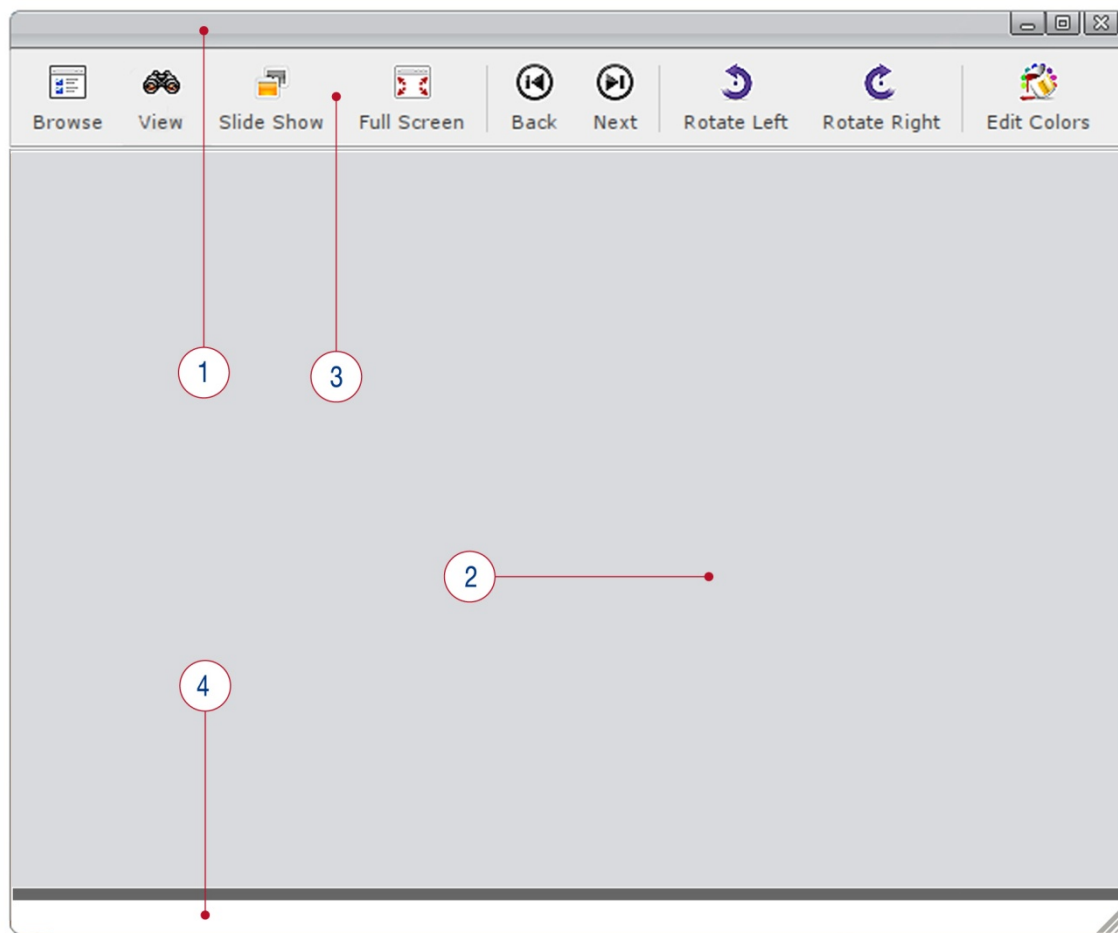
eyePictures està format per una finestra principal, la lògica de la qual es descriu mitjançant la imatge següent. Es tracta de la clàssica finestra tradicional que incorpora un gestor de finestres amb botons per minimitzar, maximitzar i tancar. Per altra banda, a la part inferior dreta es poden localitzar tres puntets que permeten a l'usuari redimensionar la finestra al seu gust.



Imatge 24 - Marc de finestra d'eyePictures

5.2 Composició i lògica

Sobre aquest marc de finestra, com mostra la següent il·lustració, es disposen un conjunt d'elements repartits tots ells sobre la mateixa interfície. Segurament, el que primer destaca és el seu contingut (2), immediatament es pot apreciar l'existència d'una llarga barra d'eines (3) amb modernes icones simbolitzant l'acció de cada botó. Si es mira a la part inferior es pot localitzar una barra d'estat (4) que proporciona certa informació, en temps real, a l'usuari. A més, s'incorpora el títol (1) d'aplicació dins el propi administrador de finestres. Aquest títol està format pel nom del propi programa acompanyat per la direcció de la ruta actual, si és que es troba en mode explorador d'arxius, o del nom complet de la fotografia, si està en ús el mode visor d'imatges.



Imatge 25 - Interfície gràfica d'eyePictures

El contingut de l'aplicació, com s'ha comentat amb anterioritat, pot variar en funció del mode que es trobi en ús. Quan s'executa el mode explorador d'arxius (il·lustració 26) es mostren imatges (sempre a la dreta dels directoris) i directoris (situats a la zona esquerra de la finestra) pertanyents a la ruta actual, juntament amb el nombre de directoris, imatges i la capacitat d'aquestes a la barra d'estat. També varia la informació mostrada a la barra d'estat segons el mode d'ús seleccionat. Si es treballa en mode navegador d'arxius es mostra el contingut del directori (il·lustració 27). En el mateix mode de treball i se situa el ratolí per sobre d'un directori aquest torna a mostrar el seu contingut (il·lustració 27) però si pel contrari se situa sobre d'una imatge es mostra el seu nom i mida (Imatge 28).

Contingut i informació del navegador d'arxius



Imatge 26 - Navegador d'arxius amb barra d'estat



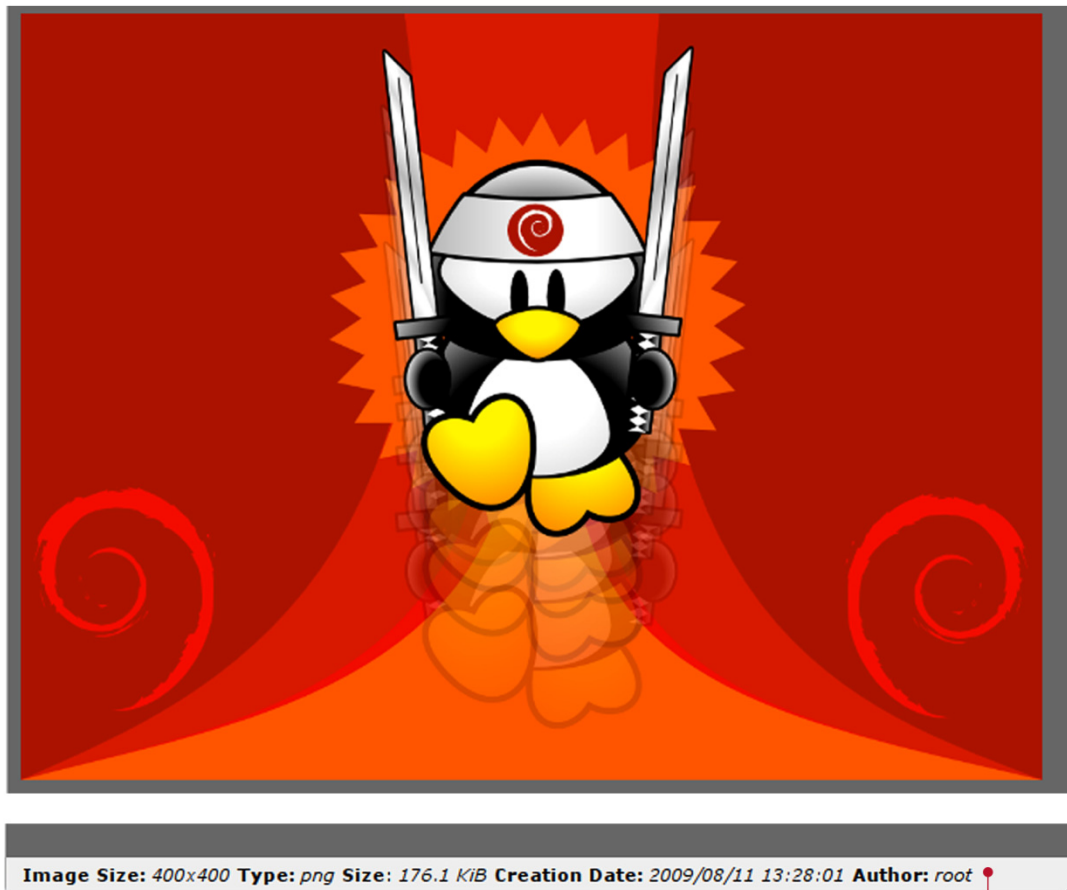
Imatge 27 - Barra d'estat, informació relativa a directoris



Imatge 28 - Barra d'estat, informació relativa a imatges

Treballant en mode visor d'imatges el contingut complet de l'aplicació consta únicament de la fotografia seleccionada . A la barra d'estat es pot consultar meta-informació de la imatge en qüestió juntament amb informació relativa a aquesta.

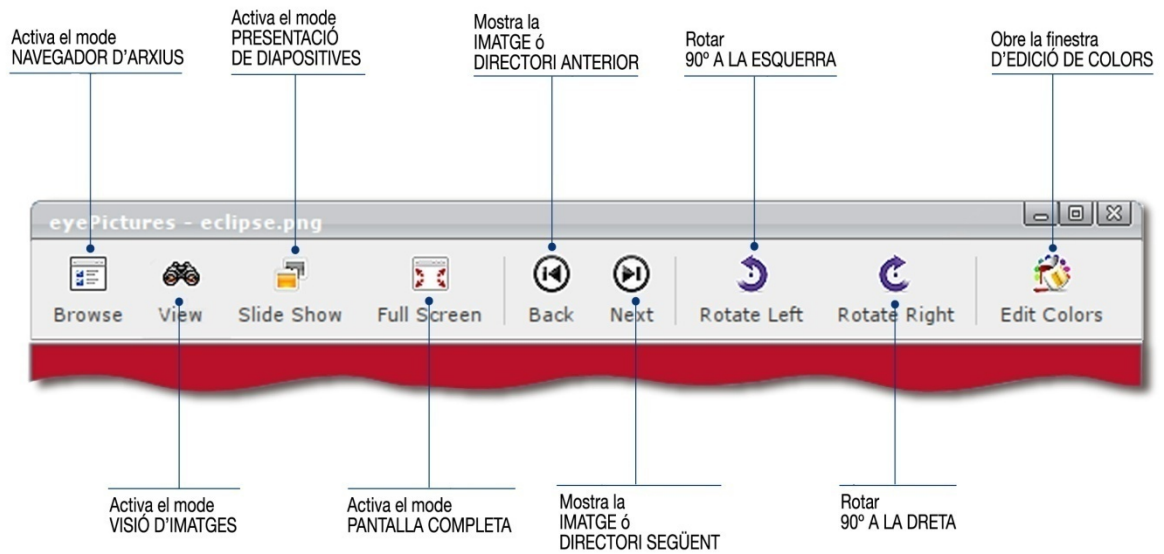
Contingut i informació del visor d'imatges



[MOSTRA META-INFORMACIÓ DE LA IMATGE](#)

Imatge 29 - Visor d'imatges amb barra d'estat

Com gairebé totes les aplicacions, *eyePictures* també disposa de barra d'eines per a complementar l'aplicació amb funcionalitats interessants i disminuir així la seva dificultat d'ús. Actualment la barra d'eines està formada per nou botons cadascun dels quals es troba acompanyat per una icona que permet a l'usuari intuir la seva funció. A més els botons estan separats per barres verticals permetent així la creació simbòlica de grups de botons segons la seva funció a desenvolupar. Finalment, cal remarcar que els modes presentació de diapositives i edició de colors es troben únicament disponibles des del mode visor d'imatges.



Imatge 30 - Barra d'eines d'eyePictures

6. Disseny tècnic i codificació

A continuació es descriuen els aspectes més rellevants del disseny tècnic i codificació d'*eyePictures*.

6.1 Arquitectura tècnica de l'aplicació

eyePictures està dissenyat de forma que la seva estructura es divideix en diferents capes o nivells. Cadascun d'aquests es situa sobre l'inferior de forma successiva, partint des de la primera capa que situada sobre el sistema eyeOS fins arribar a l'usuari per mitjà de la pròpia interfície gràfica. L'essència d'aquest mecanisme regeix en que cadascuna de les capes que componen l'aplicació delegui les tasques no pròpies en la capa inferior.

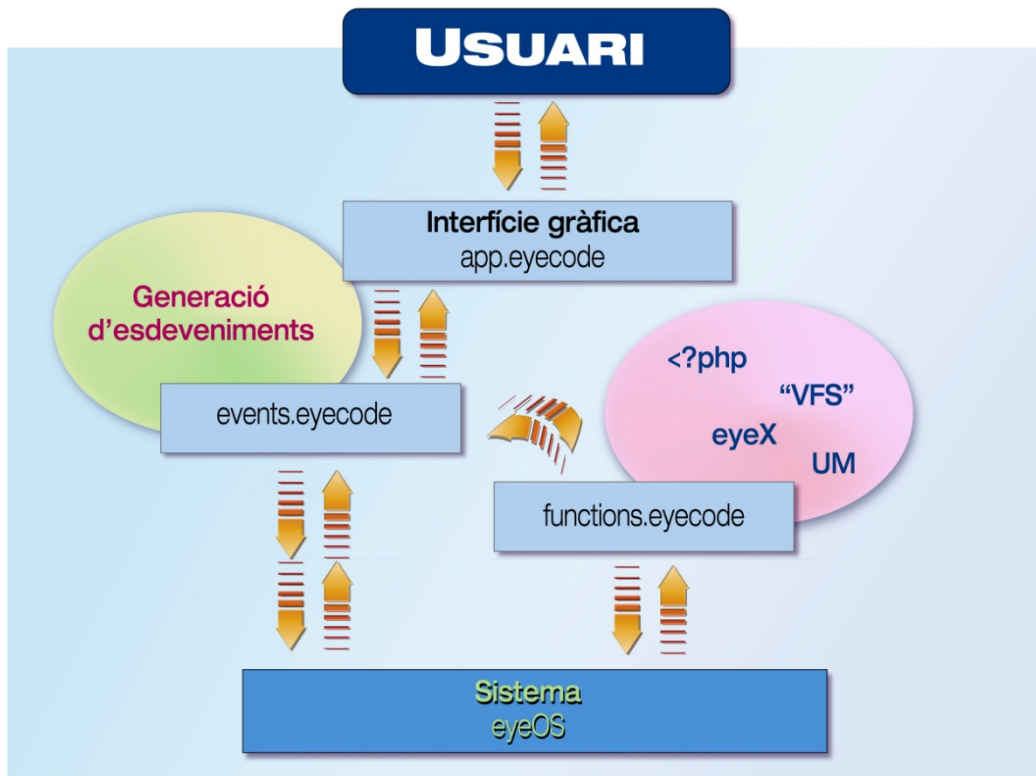
D'aquesta forma, el nivell superior està format per la interfície gràfica que s'encarrega de proporcionar a l'usuari un mètode d'interacció amb el propi sistema. És així com l'usuari genera esdeveniments, cada cop que interactua amb la interfície gràfica, els quals són subordinats a la capa inferior, encarrega de satisfer-los. Aquest ciment superior es compon únicament del fitxer *app.eyecode*, el qual conté el codi destinat a proporcionar la interfície d'usuari.

El sediment inferior a l'exposat anteriorment es compon de l'arxiu *events.eyecode* el qual conté tots els mètodes destinats a respondre els possibles esdeveniments generats pel client de l'aplicació. És mitjançant aquest codi i la seva interacció, únicament quan és necessari, amb el fitxer "*libs/functions.eyecode*" com cada acció per part de l'usuari obté una resposta immediata. Aquest nivell és també capaç de comunicar-se amb el sistema eyeOS sempre que és necessari.

Com a nivell inferior de l'aplicació tenim la llibreria "*libs/functions.eyecode*", la qual s'encarrega d'interactuar amb les funcions de tractament d'arxius i

directoris de *PHP*, amb la llibreria *GD* per a la creació d'imatges i previsualitzacions, i també amb les pròpies llibreries *PHP* del sistema eyeOS.

A continuació es mostra un gràfic on es pot apreciar cadascuna de les capes que componen l'aplicació, juntament amb la seva interacció, tant entre elles com amb el propi eyeOS.



Imatge 31 - Arquitectura d'eyePictures

6.2 Estructures de fitxers i directoris

L'aplicació es divideix en diferents directoris i fitxers, els quals es detallen a continuació:

Directori eyePictures

Es tracta del directori principal del programa i per tant conté la majoria de fitxers necessaris per a la seva execució. Quan aquesta s'executa eyeOS busca dita aplicació en la ruta "*directori_eyeOS/eyeOS/app/nomDeLaAplicació*".

Concretament busca el fitxer *app.eyecode* dins la ubicació especificada. En el cas d'*eyePictures* dins aquesta carpeta es poden trobar els següents arxius:

- *app.eyecode*: Emmagatzema el codi principal de l'aplicació, el codi que fa referència a la inicialització d'aquesta. El seu contingut consta de dues funcions:
 - *eyePictures_run()* → Conté tot el codi relacionat amb la inicialització de l'aplicació. En aquest cas conté el codi necessari però a al creació dels *widgets*, finestra principal i altres objectes, i per a la discriminació del mode a inicialitzar. Cal recordar que el programa pot funcionar en mode navegador d'arxius, visor d'imatges o presentació de diapositives.
 - *eyePictures_end()* → Conté el codi imprescindible per a la destrucció del programa. Bàsicament el que fa és deserialitzar l'aplicació.
- *events.eyecode*: Conté un extens conjunt de funcions la missió de les quals és la de respondre davant qualsevol dels possibles successos. Per a poder cridar a funcions de llibreria externes a possibles esdeveniments, com és el llistat del contingut de directoris, inclou per mitja de la sentència "*include_once*" el contingut de la llibreria "*libs/functions.eyecode*"

Les funcions que conté el fitxer *events.eyecode* són les següents:

- *eyePictures_on_Icon_Clicked()* → Event generat al fer clic sobre les icones que representen la presència de directoris o imatges dins les interfície gràfica. S'encarrega de diferenciar si l'element sobre el que s'ha fet clic és un directori o una il·lustració. En funció d'això crida la funció de llistat i mostreig de contingut de directoris o de mostreig d'imatges.

- *eyePictures_on_createThumbnail()* → Es tracta de la funció que s'encarrega de crear les miniatures d'imatge, normalment és cridada per la funció de llistat de directoris al detectar la presència de fotografies.
- *eyePictures_on_showImage()* → Conté el codi que permet la mostra d'imatges utilitzant a la llibreria *GD*, aquesta funció és només executada quan es fa clic sobre una imatge.
- *eyePictures_on_backSignal()* → S'encarrega de detectar en quin mode ens trobem i en funció d'això esbrina quin és el directori o imatge actual per a poder retrocedir.
- *eyePictures_on_backSignal()* → Té pràcticament el mateix objectiu que la funció anterior amb la excepció que aquesta refà la penúltima acció executada per l'usuari.
- *eyePictures_on_slideShowNextSignal()* → Quan l'usuari es troba en mode presentació de diapositives, s'encarrega de mostrar la següent imatge.
- *eyePictures_on_slideShowJS()* → Executa la funció de temporitzador *JavaScript*, la qual s'encarrega de cridar l'anterior funció passat el temps de mostreig de la diapositiva.

- *eyePictures_on_slideShow()* → Detecta el mode de treball actual abans d'entrar en mode reproductor de diapositives. Si l'usuari es troba en mode navegador d'arxius inicialitza la presentació començant per la primera imatge del directori actual. Pel contrari si l'usuari es troba en mode visor d'imatges inicialitza la reproducció prenent com a punt de partida la imatge en curs.

- *eyePictures_on_viewSignal()* → S'encarrega d'entrar en mode visor d'imatges un cop comprovat que aquest no es tracta del mode actual. Si el mode presentació de diapositives està en execució, l'atura.
- *eyePictures_on_browseSignal()* → S'encarrega d'entrar en mode navegador d'arxius un cop comprovat que aquest no és el mode actual. Si el mode presentació de diapositives està en execució, l'atura.

- *eyePictures_on_rotateRightSignal()* → Comprova si l'usuari es troba en mode visor o reproductor de fotografies i posteriorment cridar la funció de llibreria encarregada de rotar la imatge 270°.

- *eyePictures_on_rotateLeftSignal()* → Comprova si l'usuari es troba en mode visor o reproductor de fotografies i posteriorment cridar la funció de llibreria encarregada de rotar la imatge 90°.

- *eyePictures_on_Message()* → Té per objectiu de mantenir l'aplicació actualitzada sobre esdeveniments externs a aquesta,

necessaris per al seu redimensionament. Un clar exemple és la mida de finestra de l'explorador web.

- *eyePictures_on_winResize()* → La seva missió és la de fer els càlculs necessaris per al canvi de mida del programa. Els càlculs més comuns es tracten de la detecció de la mida relativa dels widgets a la finestra de l'aplicació.

- *eyePictures_on_fullScreenSignal()* → El seu objectiu és el mateix que el de l'aplicació anterior però per en cas que l'aplicació entri en mode pantalla completa per mitjà de la icona de la barra d'eines.
- *eyePictures_on_colorEdit()* → Obre la finestra d'edició de colors per mitjà de filtres fotogràfics un cop assegurat que es troba en mode visor d'imatges i el reproductor de diapositives està aturat.

- *eyePictures_on_closeEditColors()* → Executa accions com alliberar el botó referent a l'edició de colors de la barra d'eines.

- *eyePictures_on_undoAll()* → Restaura la imatge original després de ser modificada.

- *eyePictures_on_saveFile()* → Guarda els canvis en la imatge sense sobre escriure la original.

- *eyePictures_on_Close()* → Mata el procés de la taula de processos del *kernel* i para l'execució del mode reproductor d'imatges si és necessari.

Les funcions que es llisten a continuació s'encarreguen únicament de cridar la funció de llibreria encarregada d'aplicar el filtre un cop especificat el paràmetre necessari per al reconeixement del filtre a executar.

- *eyePictures_on_setSepia()*
 - *eyePictures_on_setGrayScale()*
 - *eyePictures_on_setPencil()*
 - *eyePictures_on_setEmboss()*
 - *eyePictures_on_setDarken()*
 - *eyePictures_on_setBrighten()*
 - *eyePictures_on_setInvert()*
 - *eyePictures_on_setBlur()*
 - *eyePictures_on_setSmooth()*
-
- *com.eyecode*: Conté una única funció encarregada de notificar a l'aplicació quan aquesta es troba en mode de pantalla completa.

 - *info.xml*: Es tracta d'un fitxer amb estructura *XML* encarregat d'emmagatzemar la configuració necessària per a la seva instal·lació a *eyeOS*. Entre altres coses guarda el nom d'aplicació a mostrar, versió, descripció, llicència, categoria i icona de programa.

- *libs/functions.eyecode*: Conté el conjunt de funcions considerades part de la llibreria del programa.
- *eyePictures_cleanBox()* → Aquesta funció s'encarrega de netejar la finestra principal de l'aplicació cada cop que es fa un canvi de directori o de mode de funcionament.
- *eyePictures_listContent()* → Llista el contingut dels directoris i mostra aquest dins l'aplicació.
- *eyePictures_showImage()* → S'encarrega de crear l'objecte sobre el qual es pintarà la imatge. Calcula la mida del mateix i canvia el títol de la finestra d'acord amb el nom de la imatge.
- *eyePictures_rotateImage()* → Rota la imatge el nombre de graus especificat.
- *eyePictures_getExtension()* → Obté l'extensió de format de l'arxiu especificat
- *eyePictures_groupFiles()* → Crea un llistat i agrupa els arxius en funció de la seva extensió.
- *eyePictures_centerImage()* → Centra la imatge a mostrar a la interfície gràfica.

- *eyePictures_setDirStatusBarInfo()* → Afegeix la informació relativa al directori especificat a la barra d'estat inferior.

- *eyePictures_setPicStatusBarInfo()* → Fa el mateix que la funció anterior però mostrant la meta-informació de les imatges.

- *eyePictures_classifyDirContent()* → Distingeix entre imatges i directoris fent una llista separada per cadascun d'aquests i calcula la seva mida en bytes.

- *eyePictures_getFileSize()* → Converteix la mida especificada de bytes a kilobytes, megabytes, gigabytes o terabytes segons sigui necessari.

- *eyePictures_backUpFile()* → Emmagatzema la imatge original en un directori temporal abans d'aplicar els filtres seleccionats.

- *eyePictures_applyFilter()* → Prepara la imatge en funció del seu format per després aplicar el filtre per mitjà de la funció que s'explica a continuació.

- *eyePictures_selectFilter()* → Aplica el filtre seleccionat i el retorna a la funció anterior.

- *eyePictures_updateCurrentPicHidden()* → S'encarrega d'actualitzar un valor serialitzat, el qual s'utilitza per guardar i obtenir el nom i ruta de la imatge mostrada.
- *eyePictures_slideShow()* → Executa el codi *JavaScript* encarregat de fer el recompte de temps entre cada imatge a mostrar quan l'aplicació es troba em mode presentació de diapositives.
- *forms/form.eyecode*: Aquest fitxer conté el codi relatiu a la creació de la finestra d'edició de colors. Aquest fitxer és cridat per la funció "*eyePictures_slideShow()*".

La següent ruta conté altres imatges necessàries per al correcte funcionament del programa, però no per a la seva inicialització: "*directori_eyeOS/eyeOS/extern/apps/eyeX/themes/default*". Es tracta del directori on *eyeOS* busca les imatges. Com que les imatges han de poder ser llegides per tothom s'han de col·locar en rutes que no comprometin la seva seguretat. Dins aquesta ubicació es poden localitzar els següents fitxers d'imatge:

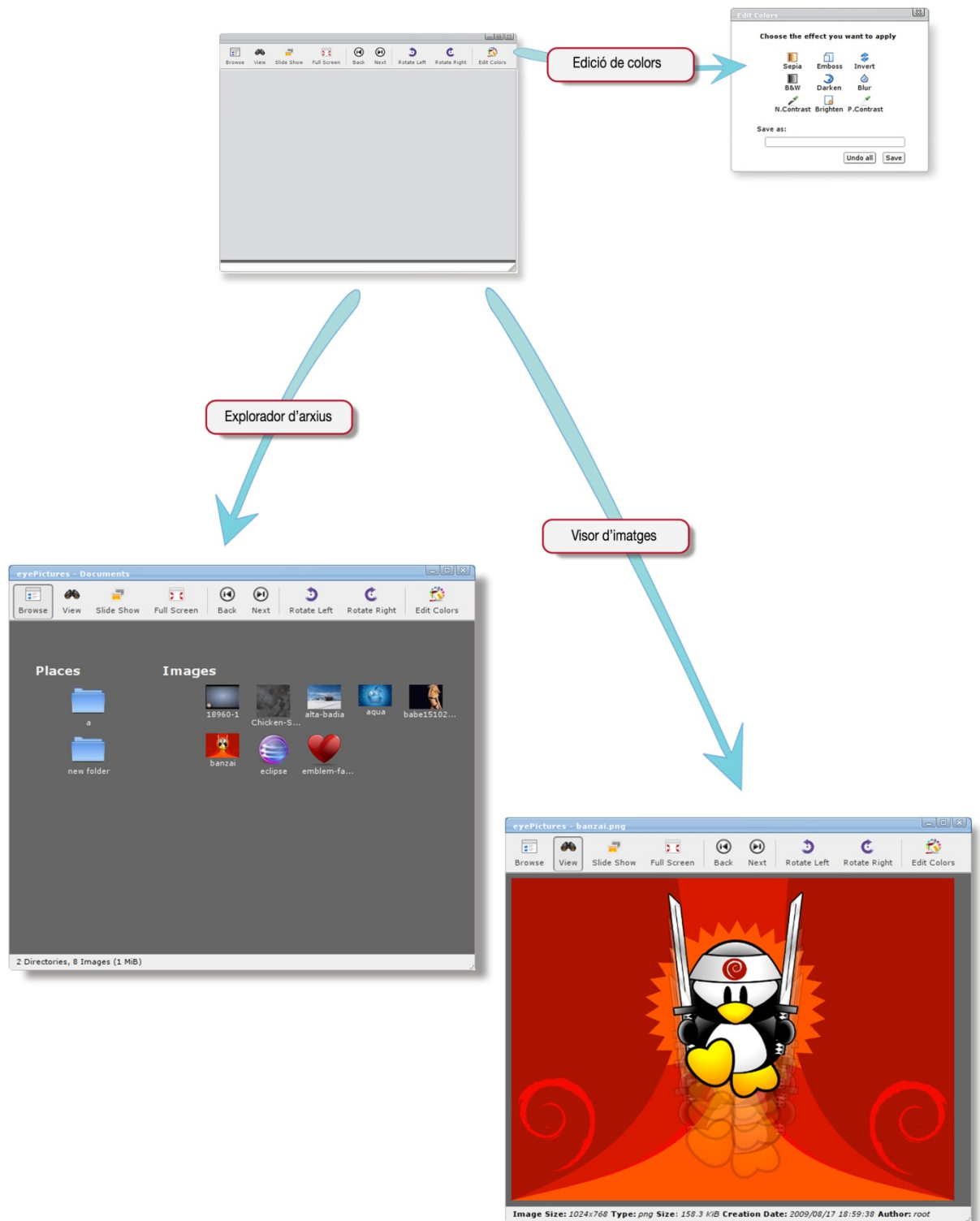
- "*icons/48x48/eyePictures.png*": Es tracta de la imatge que s'utilitza per executar *eyePictures* des del menú d'*eyeOS*.
- La següent ruta "*Images/apps/eyePictures/*", relativa a la ubicació anterior conté les dues estructures de directoris següents:
 - "*FX*": Conté les imatges utilitzades per a la finestra d'edició de colors.

- "*Toolbar*": Conté les imatges que es mostren a la barra d'eines de la interfície principal.

6.3 Disseny de la interfície gràfica

Com s'ha comentat anteriorment, la interfície gràfica d'usuari ha estat desenvolupada principalment per mitjà del "*toolkit*" d'eyeOS la qual s'ha acompanyat de funcionalitats extra per mitjà de *JavaScript* i *PHP*. En el moment de dissenyar la interfície s'ha prioritzat el disseny, la usabilitat i la inclusió de característiques que fan d'*eyePictures* una aplicació senzilla, intuïtiva, elegant i potent al mateix temps. Per a aconseguir-ho s'han hagut d'estudiar aspectes tant diferents com la disposició de cadascun dels seus elements fins als colors i icones a utilitzar.

Així es pot dir que finalment *eyePictures* compta amb la següent interfície gràfica d'usuari.



Imatge 32 - Interfície d'usuari d'eyePictures

7. Conclusions

7.1 Estat dels objectius

Els objectius inicials per aquest projecte consistien en el desenvolupament d'un visor d'imatges que combinés l'exploració de directoris locals juntament amb certes característiques pròpies d'un visor d'imatges professional. Aquestes característiques inicials consistien bàsicament en la possibilitat de rotar imatges cap a ambdues direccions, avançar i retrocedir en el mostreig d'imatges, i ser capaç de visualitzar les imatges a pantalla completa. Com s'ha pogut constatar totes aquestes característiques han estat implementades juntament amb altres.

Aquestes altres característiques que s'han afegit, permeten a *eyePictures* col·locar-se al mateix nivell que altres productes comercials:

- Mostreig tant d'informació de directoris com meta-informació o informació relativa a imatges.
- Capacitat de visualitzar conjunts de fotografies de forma totalment automatitzada.
- Capacitat d'editar la gamma de colors de les fotografies per mitja de nou filtres predeterminats.
- Creació d'un paquet auto-instal·lable per a la plataforma eyeOS.

Per tant, com a resum es pot dir que no només s'han complert els objectius inicials, sinó que s'ha ampliat la llista, amb un desenvolupat totalment satisfactori.

7.2 Possibles ampliacions.

Les ampliacions que podria incorporar amb el temps aquest projecte són nombroses, ja que es tracta d'una aplicació real, la qual ha de ser capaç de funcionar i integrar-se correctament en el programari eyeOS sense importar les modificacions o canvis de comportament que aquest últim vagi acumulant durant els seu cicle de vida.

Actualment, per exemple, es troba en desenvolupament una nova característica que permetria als usuaris configurar el temps d'espera entre cada fotografia quan l'aplicació funciona en mode presentació de diapositives. Igualment, s'està contemplant la possibilitat que *eyePictures* es posi de forma automàtica o no a pantalla completa quan el mode reproductor de diapositives s'habilita. Aquest tipus de configuració, en principi, anirà integrada en una altra aplicació anomenada *eyeControl*, i que ve instal·lada de forma predeterminada amb el paquet base d'eyeOS.

7.3 Valoració personal

La valoració global d'aquest projecte un cop finalitzat el seu desenvolupament és molt positiva, ja que gràcies a ell he pogut estar en contacte amb tecnologies i programari actual que s'utilitza en entorns de desenvolupament del món real. D'aquesta forma considero que he pogut créixer com a dissenyador i programador, i així agafar certa experiència en el que es considera el desenvolupament d'aplicacions professionals.

Per altra banda, la seva creació m'ha permès obtenir diferents punts de vista sobre la forma de crear programari i del procés de desenvolupament del que aquest consisteix.

Un altre punt molt interessant i de satisfacció personal és saber que el desenvolupament d'aquest programari ha sigut satisfactori i útil per a altres persones de forma que s'ha obert l'accés a *eyePictures* a tot el públic gràcies a

la pàgina de descàrrega d'aplicacions³⁶ d'eyeOS. Actualment compta amb 100 descàrregues.

³⁶ <http://eyeos-apps.org>

8. Referències bibliogràfiques

Material imprès

- Paul Hudson (2005): PHP in a nutshell: A desktop quick reference.

Llocs d'Internet

<http://eyeos.org>

Informació d'*eyeOS*. Setembre 2009.

<http://www.eyeos.info>

Compte d'usuari gratuïta per provar *eyeOS*. Setembre 2009.

<http://wiki.eyeos.org/>

Informació per a desenvolupadors d'*eyeOS*. Setembre 2009.

<http://doc.eyeos.org/>

Documentació per a desenvolupadors d'*eyeOS*. Setembre 2009.

<http://eyeos.org/ca/get/downloads/developermanual.pdf>

Manual del desenvolupador d'*eyeOS*. Setembre 2009.

<http://www.w3schools.com/>

Extensa informació relativa a tot tipus de llenguatges orientats a la web. Setembre 2009.

<http://www.php.net/>

Informació sobre *PHP*, funcions i característiques. Setembre 2009.

<http://eyeos-apps.org/>

Pàgina per obtenir aplicacions d'eyeOs. Setembre 2009.

<http://en.wikipedia.org>

Enciclopèdia gratuïta en línia. Setembre 2009.

<http://trycatch.lacoctelera.net/>

Bloc, extens reportatge dedicat al "*Cloud Computing*". Setembre 2009.

El sotasignant, Pau Peris Rodriguez,
professor/a de l'Escola Universitària d'Informàtica de la UAB,

CERTIFICA:

Que el treball al que correspon la present memòria
ha estat realitzat sota la seva direcció
per en Daniel Riera Terrén
I per a que consti firma la present.
Sabadell, Setembre de 2009

Signat: Pau Peris Rodriguez