

Filosofia del Software i Intel·ligència Artificial

ORIOL VENTOSA

MARC FERRÉ

GONZALO PALACIOS

POL GÓMEZ

Índex

1	Introducció	3
2	El Software	4
2.1	Què és?	4
2.2	Com es crea?	4
3	El Software Lliure	6
3.1	Què és?	6
3.2	Qui el fa?	6
3.3	Història	7
3.4	Ús actual	8
3.5	Avantatges i inconvenients	8
4	El Software Propietari	10
4.1	Què és?	10
4.2	Qui el fa?	11
4.3	Ús actual	11
4.4	Avantatges i inconvenients	13
5	Llicències de Software	15
5.1	Definicions	15
5.2	Tipus de llicències software	15
5.3	Decidir la llicència	17
6	Moviments de Software Lliure	18
6.1	Organitzacions Defensores del Programari Lliure	18
6.2	Casos d'èxit de Software Lliure	19
7	Intel·ligència Artificial	21
7.1	Què és?	21
7.2	Principals utilitats	21

8	Present i futur IA	22
8.1	Present de la IA	22
8.2	Futur de la IA	23
8.3	Velocitat d'evolució	23
9	La historia de la IA	25
9.1	Introducció	25
9.2	Evolució de la IA	26
9.3	El test de Turing	27

Capítol 1

Introducció

Aquest projecte de recerca es divideix en dues parts; els *conceptes* i els *procediments*.

El *software*, les seves filosofies i llicències formaran el primer i més teòric apartat. S'explicarà què és el software, i es separarà en *propietari* i *lliure*, dues formes de veure la creació i distribució del mateix. A més, s'introduirà l'*intel·ligència artificial* (IA), la seva història, el seu present i el seu hipotètic futur, com a preludi pels procediments del projecte.

La segona part, sobre *intel·ligència artificial*, s'explicarà el procés que s'ha seguit per a crear un seguit de *demostracions*, que apliquen un dels mètodes més coneguts de la IA; les *xarxes neuronals*. A través de demostracions gràfiques es podrà entendre millor en concepte de les ANN (*Artificial Neural Networks*).

Amb aquest projecte es pretén trobar un punt en comú entre les matemàtiques, la intel·ligència artificial, i el software (específicament, lliure).

S'ha treballat de forma altament col·laborativa, a través d'una plataforma en línia anomenada GitHub, que implementant un sistema de control de versions anomenat Git, descentralitza el projecte, i permet el lliure accés a col·laboradors externs i, especialment, a membres del mateix grup. Permet mantenir en sincronització el projecte i fusionar canvis entre els integrants del grup.

Capítol 2

El Software

2.1 Què és?

El *software* o *programari* és 'el conjunt dels programes informàtics, procediments i documentació que fan alguna tasca en un ordinador'. Per tant, qualsevol eina que s'executa en el nostre ordinador, que té una tasca definida, es pot definir com a programari. Casualment, la major part d'eines que executem diàriament en els nostres ordinadors, duen a terme alguna tasca (navegar internet, crear un document, consultar el correu electrònic, etc). Per tant, un ordinador sense programari, serveix de ben poc.

Paint, Photoshop, GIMP, Steam, Firefox, Dropbox, Office, Avast... tots aquests noms fan referència a programes, software. És sense dubte complicat imaginar un ordinador sense cap d'aquests elements instal·lats, veritat?

2.2 Com es crea?

El programari, encara que sembli redundant, es programa. Els *programadors* són enginyers que desenvolupen aquest programari. Per a crear software, s'utilitzen *llenguatges de programació*, que n'hi han molts (*C, Python, Lisp, Haskell...*) i cadascun amb el seu propòsit diferent. Quan s'ha completat el disseny d'un programa (es a dir, s'ha programat el *codi font* ¹), aquest es *compila* a *codi de màquina* ², que l'ordinador llegeix i executa.

Si es detecta un mal funcionament en algun moment en l'execució del programa, anomenat *bug*, els programadors revisen el codi font, fan les correccions necessàries, compilen el programa, i el distribueixen altre cop.

¹Instruccions que se li donen al llenguatge de programació

²Text indesxifrabable per al cervell humà, però que els sistemes operatius entenen

Quan l'usuari adquireix el programari en format de codi de màquina, o *binari*, només el pot executar. És molt complicat desxifrar el codi original a partir del producte final, però és una pràctica que solen realitzar les comunitats de *crackers*, i s'anomena *enginyeria inversa*.

Però no només els desenvolupadors de programari són qui compilen el codi font; depenent de la forma de distribució del software, l'usuari final pot compilar el software, i fins i tot, fer-hi modificacions. Però, que l'usuari tingui aquesta oportunitat o no, dependrà de si el software que ha adquirit es regeix per una llicència *lliure* o *propietària*.

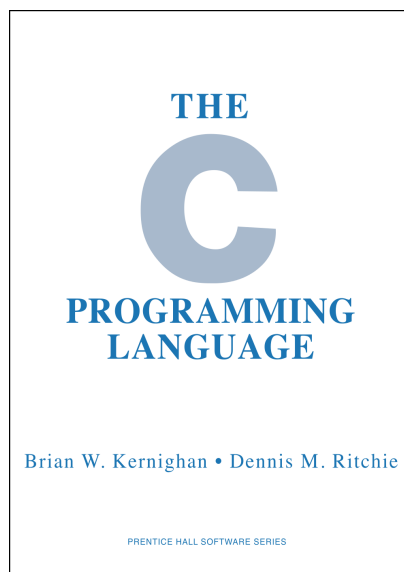


Figura 2.1: Portada del llegendari llibre que explica programació amb el llenguatge *C*, de *Kernighan* i *Ritchie*, els dos desenvolupadors principals.

Capítol 3

El Software Lliure

3.1 Què és?

El software lliure (de l'anglès "*free software*") és tot aquell software publicat sota llicències que respecten el concepte de *llibertat*. Degut a que la definició (en relació al software) de llibertat és molt àmplia, i es podria dedicar inacabable espai a definir-la, utilitzarem la definició més acceptada, la que ha popularitzat la Free Software Foundation (FSF); un programa és lliure si al adquirir-lo, l'usuari pot fer-lo servir, copiar-lo, estudiar-lo, modificar-lo, i redistribuir-lo lliurement de diferents formes. [24]

Que el software sigui lliure, no implica que el seu cost sigui zero (encara que molt software lliure, també sigui gratuït) [23]. Per tant, jo puc crear software lliure i distribuir els arxius binaris a 500,000 €, mentre mantingui un lliure accés al codi font. El desenvolupament de software lliure es sol finançar a través de donacions voluntàries, i la major part de projectes dediquen aquests diners recaptats al manteniment dels serveis que el software que distribueixen necessita (cost dels servidors i manteniment web, etc).

3.2 Qui el fa?

La major part de software lliure no és realitzat per companyies multimilionàries; en canvi, petits i mitjans grups de desenvolupadors són qui donen vida a la gran comunitat de software lliure. Tot i això, la major part de projectes de software lliure relativament importants són suportats i actualitzats per empreses que no es dediquen exclusivament a *crear* software lliure, però que n'utilitzen, i això les motiva a millorar-lo (per exemple, *HP, IBM, Intel, Oracle, Samsung, AMD, Google, Cisco, Toyota, Panasonic, Adobe, Dell, Epson, LG, Nokia, Nissan, Nvidia, Sony, Siemens, Toshiba, Yahoo...*[14]).

Definir la importància del software lliure basat en el nombre d'empreses que hi col·laboren no és la forma més correcta de fer-ho; hi han milers de programadors que actualitzen programari lliure cada dia, molts d'ells de forma altruista, i això no incrementa el nombre d'empreses.

La comunitat de software lliure és extremadament viva, i omple tots els tipus de programari que un usuari pugui desitjar. Per exemple; el projecte *Linux* accepta setmanalment un mitja de 750 col·laboracions en el seu codi font, i té més de 3,500 col·laboradors actius [40]. El reproductor de vídeo i audio *VLC* (un projecte més petit, que Linux, evidentment), té una mitja de 200 col·laboracions setmanals [17]. Dia a dia, desenvolupadors individuals o grups reduïts utilitzen diferents plataformes a internet per a iniciar els seus projectes de software lliure, i permetre la col·laboració amb molts més programadors.

3.3 Història

Des dels anys 50 fins als 70, no hi havien grans corporacions que llicenciessin software, i es solia compartir de forma lliure entre programadors, i distribuir de forma integrada en el *hardware* (es a dir, els ordinadors). Una vegada entrats en els 70, la indústria del software va començar a mostrar la seva capacitat econòmica, i es va començar a vendre programari per separat. [21]

En *Richard Matthew Stallman*, va anunciar el projecte *GNU* (*GNU no és Unix!*, conjunt d'utilitats per a iniciar un sistema operatiu lliure), argumentant que s'havia cansat dels efectes del canvi en la cultura de la indústria informàtica i els seus usuaris. La FSF (*Free Software Foundation*) va ser fundada l'Octubre de 1984, i va desenvolupar una de les definicions de *software lliure* més acceptades, i el concepte de *copyleft*, dissenyat per a assegurar la llibertat de software per a tothom. [8].

A partir d'aquell moment, la comunitat de software lliure va començar a créixer.



Figura 3.1: El logotip de la FSF.

3.4 Us actual

En l'actualitat, hi ha molt software lliure en circulació activa, i una gran comunitat de desenvolupadors darrere d'ell, però no és tant popular com el software propietari (que no vol dir que no sigui utilitzat tant).

No només els usuaris particulars tenen a l'abast (i utilitzen) software lliure: molts governs han fet (o estan fent) el traspàs al software d'aquest tipus (Kerala, a la Índia, Munich, a Alemanya, Veneçuela, Malàisia, Perú, Equador, i altres). [18]

3.5 Avantatges i inconvenients

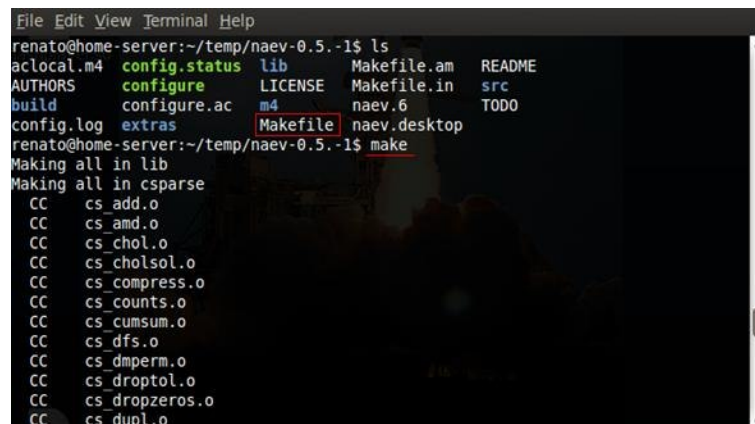
El software lliure té una quantitat important d'avantatges [2]:

1. *Econòmic* - molta part del FOSS (software de codi obert i lliure) és gratuït, o té un preu molt baix. Les petites empreses es poden beneficiar d'això, i expandir la seva infraestructura informàtica sense gastar milers d'euros en software propietari.
2. *Llibertat d'ús i distribució* - es pot instal·lar software lliure sense limitacions per culpa de llicències d'un sol ús, com passa amb els sistemes operatius i 'suites' ofimàtiques.
3. *Independència tecnològica* - l'accés al codi font permet desenvolupar nous productes amb una base sòlida de software, o ajustar-lo a les nostres necessitats. Quan s'utilitza software lliure, no s'ha de patir per les decisions de l'entitat creadora, ja que sempre es tindrà accés a versions més antigues, o en casos més avançats, al codi font.
4. *Sistemes sense 'backdoors'* - tenir accés al codi impossibilita la implementació de *espies* en el codi font d'un programa.
5. *Correcció més ràpida i eficient d'errors* - la comunitat activa de desenvolupadors de software lliure realitzen actualitzacions constants, i errors o 'bugs' són arreglats molt més ràpid que en el software propietari.

De totes formes, també té desavantatges [29]:

1. *Falta de garantia* - el software lliure no ofereix garanties; si es trenca, no és culpa de ningú excepte teva, si es que has fet alguna cosa malament.

2. *Difícil d'adquirir* - hi ha una certa quantitat de software lliure que és més complicat de descarregar i instal·lar en comparació amb el seu germà propietari. Es deu, en molts casos, en que els autors es centren en el *codi* del programa, i no tenen els recursos suficients per assegurar la disponibilitat de paquets compilats per a cada plataforma.
3. *Acabat final* - molt software lliure ofereix un aspecte gràfic o final poc atractiu; no hi han equips dedicats íntegrament al desenvolupament de l'interfície gràfica, i es fa el que es pot, amb els recursos que es tenen.
4. *Entreteniment* - els títols AAA (jocs amb pressupost molt elevat) no són lliures. Falta molt de temps per a que l'usuari comú pugui observar l'impressionant codi font de títols com *Battlefield* o *Far Cry*, ja que el mercat és més lucratiu que ètic, i s'hi troben molts inconvenients en alliberar el codi font d'un *engine* (motor gràfic).



```
File Edit View Terminal Help
renato@home-server:~/temp/naev-0.5.-1$ ls
aclocal.m4  config.status  lib          Makefile.am  README
AUTHORS    configure      LICENSE      Makefile.in  src
build       configure.ac   m4          naev.6       TODO
config.log  extras        Makefile     naev.desktop

renato@home-server:~/temp/naev-0.5.-1$ make
Making all in lib
Making all in csparse
CC      cs_add.o
CC      cs_amd.o
CC      cs_chol.o
CC      cs_cholsol.o
CC      cs_compress.o
CC      cs_counts.o
CC      cs_cumsum.o
CC      cs_dfs.o
CC      cs_dmperm.o
CC      cs_droptol.o
CC      cs_dropzeros.o
CC      cs_dupl.o
```

Figura 3.2: Procés de compilació de Linux (wonderhowto.com).

Capítol 4

El Software Propietari

4.1 Què és?

Anomenem software propietari a tot aquell programa publicat sota llicències que reserven un o tots els drets d'ús, còpia, modificació i distribució al fabricant qui, pagant, concedeix un ús del programa executable al titular de la llicència.

Per tant, el software *propietari* o *privatiu* obstrueix la llibertat de l'usuari final, que quan ha adquirit el programa, té uns drets limitats i fortes obligacions, que solen incloure la impossibilitat d'adquirir i modificar el codi font del producte que ell mateix ha comprat, tant com la prohibició total o parcial de la redistribució del programa. [1]

Per a entendre millor el software propietari, posem un exemple pràctic: en Jaume crea un programa impressionant, amb l'ajut de diferents col·laboradors (que ho fan com a hobby), i distribueix el programa a través d'una llicència permissiva. Amb el temps, més gent col·labora i cada vegada, aquest programa fet amb l'esforç de centenars de voluntaris, adquireix més importància en el món del *software*. En Jaume, però, segueix volent mantenir la filosofia lliure del software que la comunitat ha creat, i per tant manté la llicència permissiva.

Un dia, una companyia, agafa tot el codi d'en Jaume, el modifica lleugerament, i el redistribueix amb una llicència propietària. Ara, totes les millores que es podrien haver incorporat al codi d'en Jaume a benefici de la comunitat, estan en mans d'una gran corporació, que no les vol compartir. Degut a la llicència que en Jaume ha fet servir, no té dret a res, i la comunitat que ha treballat durant tot aquest temps, es quedarà sense les millores que la corporació ha desenvolupat, sobre la seva base.

Aquesta situació es pot evitar o repetir, tot depèn de la llicència que en

Jaume esculli al crear el seu següent programa.

4.2 Qui el fa?

El principal desenvolupador de software propietari a nivell mundial és *Microsoft*, encara que hi ha moltes més empreses que també en creen i distribueixen, com *Apple*, *Oracle*, *Adobe*, *VMware*, *SAP*, *Symantec*... [20]

4.3 Ús actual

Avui en dia molta part del software utilitzat per la majoria de població, és propietari.

Aquesta gran extensió del seu ús és degut a l'inversió milionària al màrket-ing, i a pactes amb productors de sistemes operatius i proveïdors d'Internet, que acorden la prèvia instal·lació de software propietari als ordinadors. Junt amb la falta d'informació del que comporta l'ús de programes propietaris, i la falta de preocupació de la integritat dels nostres drets virtuals, son els motius principals de l'expansió del software propietari.

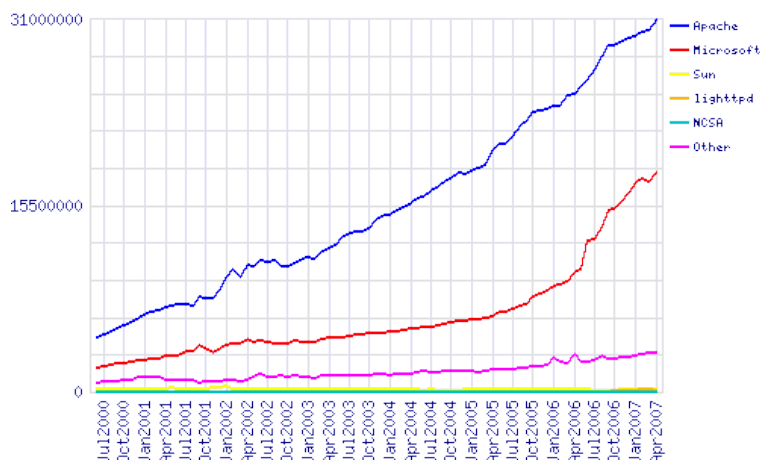


Figura 4.1: Ús de software en servidors web [43]

El gràfic 4.1 mostra la distribució (o *market share*) de diferents companyies de software en l'àmbit dels servidors web ¹. *Apache*[16] ha mantingut sempre la seva posició, seguit per *Microsoft* i altres companyies. En aquest

¹Ordinadors que funcionen el màxim de temps possible i comparteixen l'accés a una direcció web. Per exemple, *Google* té molts servidors que permeten l'accés públic als seus serveis.

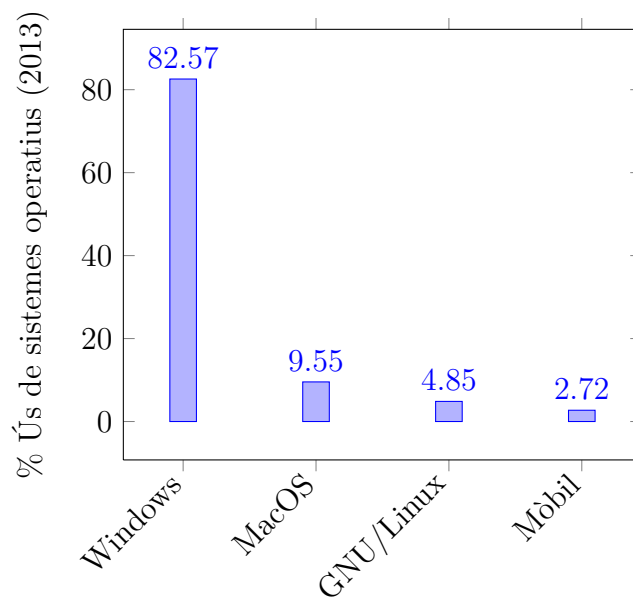


Figura 4.2: Ús de sistemes operatius per a particulars [41]

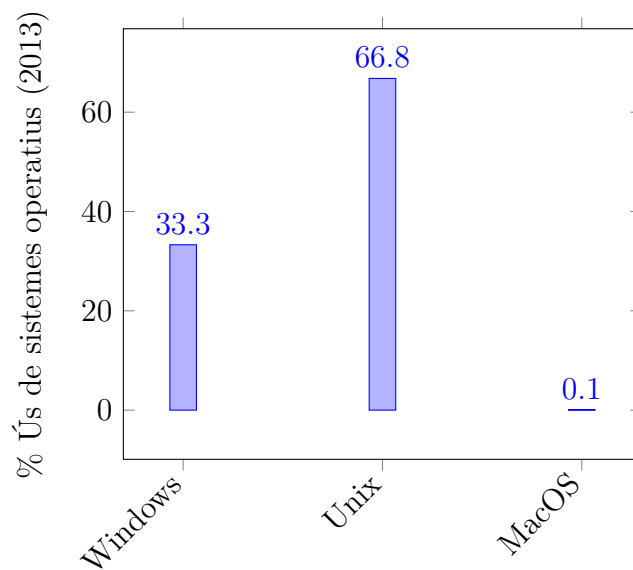


Figura 4.3: Ús de sistemes operatius per a servidors [42]

cas, el software lliure (de la mà de la llicència *Apache 2.0*[15]) és prevalent de llarg.

El gràfic 4.2 mostra el market share de *sistemes operatius* que utilitzen

els particulars (qualsevol persona). *Microsoft Windows* és el guanyador indiscutible, amb *MacOS* molt per darrere i els sistemes operatius *GNU/Linux* (l'alternativa lliure) amb menys del 5% del share.

El gràfic 4.3 mostra el market share de sistemes operatius que utilitzen els servidors i, al igual que en els programes que utilitzen els servidors, el software lliure guanya.

D'aquests documents podem extreure una conclusió ben simple; la població general, no especialitzada, utilitza generalment software propietari (que, casualment, els hi ve pre-instal·lat als ordinadors). En canvi, els usuaris especialitzats (mantenidors de servidors, en aquest cas), es passen a les alternatives de software lliure, que amb el temps han demostrat més viabilitat i seguretat per al ús que s'els hi ha de donar. Per tant, el software propietari és majoritari per *conveniència*, no per qualitat.

4.4 Avantatges i inconvenients

Els avantatges principals del software propietari són ben simples; solen ser avantatges per part de l'empresa creadora, ja que generen beneficis més elevats que el software lliure (que es sol distribuir de forma gratuïta) [29].

Per part de l'usuari, en podem extreure alguns:

- *Atenció al client*: molt software lliure no ofereix ni garanties, ni atenció al client 'professional'. La major part d'empreses de software propietari es veuen obligades a oferir-ne, ja que si no, el seu producte (generalment, que comporta un cost econòmic) perd reputació i clientela.
- *Fàcil adquisició*: avantatge per a alguns, inconvenient per a uns altres (no volen software pre-instal·lat als seus ordinadors).

En quant a inconvenients, en podem trobar bastants:

- *Falta de suport multi-plataforma*: molt software privatiu es centra en les plataformes principals (Windows i MacOS), i s'oblida de les minoritàries.
- *Impossibilitat de modificació*: un programa propietari té un error amb solució simple, i tu tens coneixements de programació? Oblida't d'arreglar-ho, hauràs d'esperar a una actualització oficial.
- *Impossibilitat de còpia*: en molts casos, si vols instal·lar un mateix software a diferents ordinadors de la teva propietat, hauràs de comprar la mateixa llicència varies vegades.

- *Falta de seguretat personal:* no pots saber si el software recull dades del teu ordinador, ni a qui les comparteix.
- *Dependència a l'empresa:* quan compres un programa, quedes subjugat a les decisions que l'empresa faci sobre el mateix.

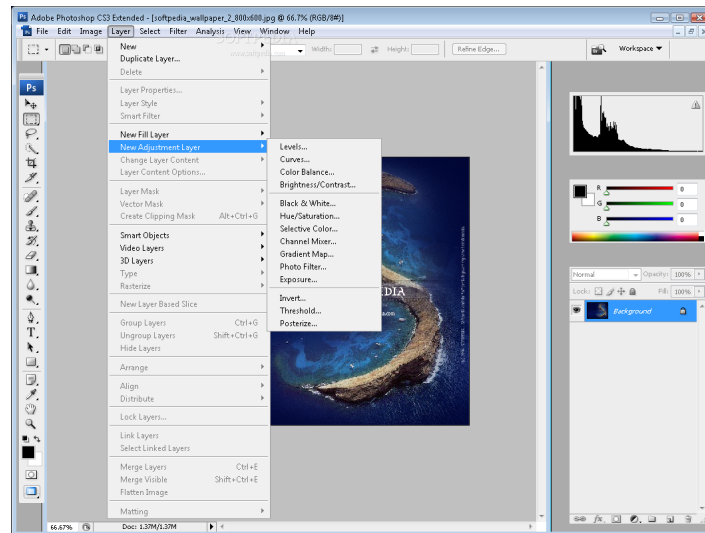


Figura 4.4: Captura del programa propietari *Photoshop CS3*.

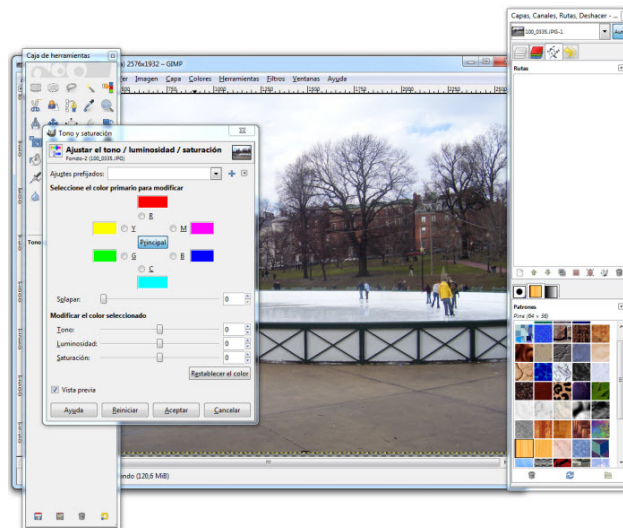


Figura 4.5: Captura de l'alternativa lliure de Photoshop, *GIMP* (GNU Image Manipulation Program)

Capítol 5

Llicències de Software

5.1 Definicions

Abans de parlar sobre els tipus de llicències software s'haurien de conèixer alguns conceptes bàsics.

- *Llicències de software*: són un contracte entre el desenvolupador del software (sotmès a propietat intel·lectual i els drets d'autor), i l'usuari. En aquest contracte es defineixen els drets i deures de ambdues parts. El desenvolupador, o qui hagi cedit els drets d'explotació del producte, és la persona qui decideix quina llicència software usar per la distribució del programa.
- *Patent*: és el conjunt de drets exclusius concedits per un estat al creador o als creadors de un producte susceptible a ser explotat industrialment, per un període limitat de temps a canvi de la divulgació de la invenció. Vol dir bàsicament que tercers no facin ús de la tecnologia patentada. [38]
- *Drets d'autor o copyright*: és un conjunt de drets i normes [26] que tenen els autors de creacions de obres de qualsevol tipus, tant científica, tecnològica, didàctica...

5.2 Tipus de llicències software

Hi han molts tipus de llicències software, però les més utilitzades són aquestes quatre:

- *GPL*: prové de *GNU Public License* (GNU essent acrònim de *GNU no és Unix*). Aquesta llicència permet la còpia, la distribució, tant amb fins

comercial com no, i permet la modificació del codi només si es segueix utilitzant el mateix tipus de llicència GPL. No permet la distribució d'executables sense mostrar el codi font d'aquest. És la més usada en el món del software lliure, i garanteix a l'usuari final la llibertat de usar, estudiar, compartir i modificar el software amb el propòsit d'evitar que el software tingui una llicència de software privativa i protegir-lo dels intents d'apropiació que restringeixin les llibertats de l'usuari. Aquesta llicència va ser creada per *Richard Stallman*, fundador de la *Free Software Foundation* per el projecte del grup *GNU*. Segons aquest grup, quan es parla de que és *free* es refereixen a que és lliure, no gratuït. Això vol dir que tu tens la llibertat de compartir i de modificar les versions del programa perquè estiguin segurs de que és lliure per tots el usuaris. Si fos gratis en comptes de lliure voldria dir que tu pots fer us del programa però no tindries el codi font per modificar-lo ni la llibertat per compartir-ho [22] [39]

- *BSD*: o *Berkeley Software Distribution* és una llicència software més permissiva que GPL, ja que aquesta té menys restriccions en comparació a la anterior. La llicència BSD al contrari que la GPL permet l'ús del codi font en software no lliure. Aquesta llicència es podria definir com a molt liberal ja que no es fa responsable del que fas amb el teu software, o sigui que si per culpa teva es perden dades, es danyen ordinadors o obtens benefici per el teu producte, no et poden acusar. L'únic que has de tenir en compte per aquesta llicència és mantenir el document de llicència BSD. Molts sistemes operatius descendents de BSD són *SunOS*, *FreeBSD* i *MacOS X*, entre altres. [25] [39]
- *MIT*: La llicència MIT (Massachusetts Institute of Technology) té unes característiques molt similars a la llicència BSD: pots fer el que vulguis amb el teu software mentre tu adjuntis el copyright inicial. Una quantitat de packs de software utilitzen llicències MIT com ara el Projecte *Mono*, o *Ruby on Rails*, entre moltes altres.[31] [39]
- *WTFPL*: és la llicència més permissiva. Bàsicament, et permet fer el que vulguis amb el teu programa com el mateix nom de la llicència indica: *Do What The Fuck You Want To The Public Licence*. L'usuari pot fer el que vulgui amb el codi font i la llicència en sí, sense cap mena de restricció. Aquesta llicència és poc utilitzada, degut a la seva falta de restriccions, i el fet que no assegura la continuïtat de les llibertats que ella mateixa proporciona.[39]
- *MPL*: la *Mozilla Public License* és una llicència no lucrativa que et dona

una varietat explícita a mesura que mantens el programa Open Source (de codi font accessible per a tothom). Aquesta llicència no és molt estricta i només té uns requeriments molt senzills. Els programes que utilitzen aquesta llicència són bàsicament els de la *Mozilla Foundation*, per exemple el navegador *Firefox* o el client de correus *Thunderbird*, però també és usat per altres programes, com per la companyia *Adobe* en la seva línia de productes *Flex*, o per *LibreOffice*, popular suite d'ofimàtica. [39]

5.3 Decidir la llicència

Quan el desenvolupador (o companyia) ha de decidir quin tipus de llicència vol usar per el seu software ha de tenir en compte les seves motivacions: si aquest vol remuneració monetària usarà una llicència de software privativa en el que el seu producte no pugui ser compartit ni modificat o si vol que el seu producte estigui l'abast de la comunitat, en aquest cas haurà de decidir el grau de llibertat que vol que tingui l'usuari.

De totes formes, cal mencionar la possibilitat de obtenir remuneració econòmica amb llicències de software lliures: ja sigui a partir de donacions (el mètode més habitual), o amb la venda directa de l'executable.

Capítol 6

Moviments de Software Lliure

6.1 Organitzacions Defensores del Programari Lliure

El programari lliure es refereix a '4 Graus' de llibertat per als usuaris:

- *Grau 0* - llibertat per a *executar el programa*, per a qualsevol propòsit.
- *Grau 1* - llibertat d'*estudiar* el funcionament del programa, d'*adaptar-lo* a les necessitats pròpies del usuari i de *tenir accés* al codi font.
- *Grau 2* - llibertat de poder *distribuir* còpies amb la comunitat.
- *Grau 3* - llibertat de *configurar* de manera beneficiosa el programa, i tenir la possibilitat de distribuir còpies d'aquesta millora a una comunitat per a que es puguin beneficiar.

Les organitzacions més influents que s'encarreguen de defensar el software lliure són:

- *Electronic Frontier Foundation*: organització sense ànim de lucre basada en part en la primera enmenda de la Constitució dels Estats Units, que defensa la llibertat d'expressió, adaptant-la als *ciber-drets*, o *drets virtuals*. Formada en 1990 per *Mitch Kapor*, *John Gilmore* i *John Perry*, com a organització lliure volen educar a la premsa, els legisladors i el públic sobre les qüestions de llibertats civils relacionades amb les tecnologies, i actuar per a defensar-les. [46] [12]
- *Free Software Foundation*: Al igual que la *EFF*, és una organització sense ànim de lucre fundada per *Richard Matthew Stallman*. És, possiblement, la organització més influent de programari lliure, format

ahora per una comunitat ètica en tot el món dedicada exclusivament al software lliure i la protecció d'aquest, dividit en diverses etapes:

- Mantenir una definició universal de programari lliure.
- Mantenir una educació legal sobre el FOSS, celebrant sovint seminaris sobre aspectes legals de l'ús de la llicència *GPL* i oferint un servei de consulta per a advocats.
- Aconseguir que tothom tingui la possibilitat de *tindre control sobre la tecnologia* quotidiana, sense restriccions de caire governamental o corporatiu. A través del desenvolupament d'un sistema operatiu completament lliure, es vol aconseguir aquest objectiu. [36] [13]

6.2 Casos d'èxit de Software Lliure

- *GNU*: va ser creat per *Richard Matthew Stallman* el 1983, com a un sistema operatiu (kernel + utilitats) que es va posar en marxa per a les persones que treballaven i segueixen treballant juntes per la llibertat de tots els usuaris del programari per poder gaudir de tots els graus de llibertat d'una manera *total*. Les motivacions principals que van portar a Richard a dur a terme GNU estan recollides en un document escrit per ell anomenat *GNU Manifesto*. [28] [27] [48] [4]
- *Mozilla Corporation*: empresa filial propietària total de la *Fundació Mozilla*, sense ànim de lucre, coordinadora i responsable de l'integració d'aplicacions informàtiques tals com el conegut navegador web *Mozilla Firefox* o el client de correu electrònic *Mozilla ThunderBird* en el món informàtic. Aquests programes s'actualitzen diàriament per mitja de milers de programadors voluntaris que treballen juntament amb els de la corporació que es regeixen per uns principis que l'empresa té establerts. [5] [6] [7]
- *GNU/Linux*: és un sistema operatiu lliure, format pel *kernel*¹ Linux i les utilitats del projecte GNU. Dissenyat per milers de programadors d'arreu del món, segueix en desenvolupament sota la coordinació de *Linus Torvalds*. Cada dia s'actualitzen nous continguts, que milloren el funcionament del sistema operatiu. [33] [35]
- *Chromium*: és un projecte de navegador de codi obert que té com a objectiu construir una manera més segura, més ràpida i estable per a els

¹Programa que es comunica amb els components d'un ordinador i executa i administra els recursos del mateix.

usuaris d'experimentar amb internet. El navegador conté documents de disseny, informació de proves i altres continguts per ajudar a aprendre, construir i treballar amb el seu codi font. També es la base de *Google Chrome*, la versió propietària desenvolupada per *Google*. [45]

Capítol 7

Intel·ligència Artificial

7.1 Què és?

La intel·ligència artificial és aquella branca de la informàtica dedicada al desenvolupament d'algorismes per a aconseguir que una màquina prengui decisions racionals per si mateixa o que es comporti de forma similar a la intel·ligència humana.

En resumides comptes, segons la definició més estesa, que és la de l'informàtic i investigador cognitiu estatunidenc *John McCarthy*, és "*Fer que una màquina es comporti d'una manera que en un humà considerariem intel·ligent*".

En el que a robòtica es refereix la intel·ligència artificial consisteix a aplicar la definició anterior, és a dir, un ésser no viu amb una intel·ligència racional semblant a la humana, a una estructura que sol tenir una fisiologia semblant a la nostra i que es pot moure. [9] [19] [34]

7.2 Principals utilitats

La IA s'utilitza diàriament en àmbits extremadament diversos: diagnosi mèdica, comerç amb stock, control de robots, llei, traducció de textos, domòtica, etc.

Capítol 8

Present i futur IA

8.1 Present de la IA

Actualment, el camp de la IA s'ha especialitzat molt: *aprenentatge de màquina, robòtica aplicada, deducció, raonament i solució de problemes, processament de llenguatge natural, movió i manipulació, percepció, intel·ligència social, creativitat*, i l'objectiu final: *intel·ligència artificial general*, que combina tots els aspectes.

En el nostre dia a dia i han molts programes que utilitzen intel·ligència artificial (de forma més o menys proficient) sense que ens n'adonem, com per exemple, el *traductor de Google*: sembla molt fàcil buscar al diccionari les paraules que es demanen per a traduir, però una traducció literal mai és satisfactòria, per això es segueixen procediments intel·ligents que permeten realitzar traduccions naturals.

YouTube i *Google* utilitzen algorismes intel·ligents per a filtrar resultats de cerca segons els teus gustos. *eBay* i *Amazon* també ho fan, recomanant productes d'acord amb el teu historial de compra.

Pel que fa l'àmbit de la robòtica, s'han creat robots capaços de caminar de forma òptima i evitar caigudes causades per els diferents tipus de terreny intentant simular animals com ara gossos en córrer. També s'han creat humanoides, robots amb forma de persones, que són capaços de fer moviments humans, per exemple abraçar expressar emocions i manipular certs objectes. [10]

Un altre cas d'intel·ligència artificial és la que s'utilitza per complementar (o formar) un programa informàtic; el reconeixement per veu es un bon exemple. *Apple* utilitza la interpretació de la veu humana per a crear una aplicació que respon a qualsevol pregunta.

8.2 Futur de la IA

És difícil predir quin serà el futur de la intel·ligència artificial ja que tot són hipòtesis que probablement no seran certes, però ens podem fer una idea de quins objectius de futur pot tenir la intel·ligència artificial.

Un dels objectius que tenen la majoria de persones que treballen en la IA és aconseguir que aquesta intel·ligència deixi de ser tant específica i passi a ser global capaç d'aprendre qualsevol cosa, analitzar-la de manera intel·ligent i pensar bàsicament com les persones humanes. Aquest objectiu encara és difícil d'aconseguir perquè si volem que pensin com a éssers humans primer hem de saber perquè pensen com a éssers humans.

Aquest concepte pot semblar estrany però el que pretenem dir és que hem d'entendre primer com funciona el nostre cervell per raonar, com reacciona en situacions que no coneix i quina és la seva reacció davant d'aquestes situacions. Per tant podem dir que la IA pot estar lligada a la neurociència i que, en part, depèn d'ella; el desenvolupament de la neurociència porta de la mà el desenvolupament de la intel·ligència artificial.

Un altre dels objectius de futur de la IA és el que fa referència a la intel·ligència específica i en aquest cas està més a l'abast dels programadors. Un dels molts projectes de aquesta intel·ligència és el de cotxes que siguin capaços de conduir sense la necessitat de un conductor. Aquest programa podria evitar molts accidents causats pel consum de drogues o d'alcohol, i la incapacitat de persones grans al volant, fent més segures les carreteres i evitant errors humans que a qualsevol persona podem passar-li. Aquest exemple és un dels múltiples objectius de la intel·ligència artificial específica i podríem estar-ne dient molts més ja que és un camp molt ampli i que encara es pot explotar.

8.3 Velocitat d'evolució

Es podria dir que hem avançat molt en el camp de la intel·ligència artificial en els últims anys, però té problemes per a continuar evolucionant. Això és donat per diversos factors [37]:

- *Neurociència*: com hem explicat en un dels apartats anteriors encara es té un desconeixement molt gran sobre el funcionament del cervell; això impedeix simular-lo, evidentment.
- *Desconeixent general*: hi ha un clar desconeixement general de la intel·ligència artificial, degut a que es tracta d'un camp en continu de-

senyolupament, amb molts poques *lleis* que es segueixen al peu de la lletra en qualsevol cas.

- *Falta d'inversió*: els governs d'avui en dia no consideren un factor important la intel·ligència artificial i aquest factor va lligat amb el factor anterior: el desconeixement.
- *Poder computacional*: si en una cosa coincideixen molts científics, es en que el cervell és una màquina de càlcul molt potent, que realitza interconnexions de forma molt ràpida, continua, i paral·lela. Els ordinadors han d'avançar en poder computacional per a poder establir-se al nivell del cervell humà, o s'han d'optimitzar els algorismes.

Capítol 9

La historia de la IA

9.1 Introducció

El concepte d'una *màquina pensant* es va introduir el 2500 a.C, quan els egipcis miraven a estàtues parlants en busca de consells místics. Segles després, durant el segle XV, els autòmats preferits de la societat eren els óssos que tocaven tambors, figuretes ballarines que apareixien cada vegada que un rellotge marcava l'hora. *Isaac Asimov*, un símbol en el camp de la robòtica, va ser escriptor, erudit i autor de les lleis de la mateixa. *Asimov* estava anyllum dels pensadors de l'època i va fer prediccions en les quals la "cibernètica" (a Asimov li agradava referir-se a la robòtica amb el nom de cibernètica), provocaria una revolució intel·lectual. Isaac Asimov va escriure en el pròleg de *Thinking by Machine: a Study of Cybernetics*, de *Pierre de Latil*:

La cibernètica no és merament una branca de la ciència, és una revolució intel·lectual que rivalitza en importància la Revolució Industrial. És possible que al igual que una màquina pot fer-se càrrec de les funcions rutinàries dels músculs humans, un altre pugui fer-se càrrec dels usos de rutinari de la ment humana?

Finalment el terme va ser establert el 1956, en la *Conferència de Dartmouth*, un congrés en el qual es van fer previsions triomfalistes a deu anys que mai es van complir, el que va provocar l'abandó gairebé total de les investigacions durant quinze anys. La *Conferència de Dartmouth* va intentar esbrinar com fabricar màquines que utilitzin el llenguatge, formin abstraccions i conceptes i siguin capaces d'auto-millorar-se; l'estudi va durar 2 mesos i estava format per 10 persones. Això sense contar que el matemàtic, lògic, científic de la computació, criptògraf i filòsof britànic *Alan Turing* ja havia dissenyat en el 1936-1937 la *Màquina de Turing* (model computacional que

va demostrar que no hi ha un mètode definit que es pugui aplicar a qualsevol sentència matemàtica). La pregunta bàsica que Turing va tractar de respondre era [32] [49]:

Poden les màquines pensar?

Els arguments a favor de Turing sobre la intel·ligència artificial, van iniciar un debat intens que va marcar clarament la primera etapa de interacció entre la intel·ligència artificial i la psicologia. De fet, se sap que anys enrere diferents filòsofs i matemàtics ja pensaven tant en la intel·ligència artificial com en el seu funcionament; per exemple, Aristòtil, va ser el primer en descriure de manera estructurada un conjunt de regles que descrivien el funcionament de la ment humana [47] [3].

9.2 Evolució de la IA

- *Aristòtil* (300 a.C): va ser el primer en descriure de manera estructurada un conjunt de regles que descrivien el funcionament de la ment humana
- *Ctesibi d'Alexandria* (250 a.C): va desenvolupar una màquina capaç de regular el flux d'aigua que actua modificant el comportament de la màquina.
- *Gottlob Frege* (1879): amplià la lògica booleana i establí la Lògica del Primer Ordre (sistema lògic-deductiu que restringeix quines són les expressions correctament formades). Aquesta ordre és tant summament important que té el poder expressiu suficient per definir pràcticament totes les matemàtiques.
- *Lee De Forest* (1903): inventa el tríode, un component electrònic usat per amplificar, commutar, o modificar una senyal elèctrica. [30]
- *Alan Turing* (1936-1937): considerat com el pare de la ciència informàtica, va formular el concepte de *algorisme*, i va inventar la *Màquina de Turing*. Va ajudar a Anglaterra contra els Alemanys en la Segona Guerra Mundial i va publicar un article on es va demostrar que existeixen problemes dels quals no es pot obtenir una solució; ni per mitjà humà ni per l'ús de computadores.
- *Warren McCulloch i Walter Pitts* (1943): van formular un model de neurones artificials sense considerar-se un treball del camp de la intel·ligència artificial degut a la inexistència d'aquesta en la època.[44]

9.3 El test de Turing

Alan Turing, apart de ser un inventor i persona d'èxit, va dissenyar un simple i lògic sistema durant 1950 capaç de *verificar si una maquina és o no intel·ligent*.

El test és duu a terme simplement situant un humà i una maquina separats per una paret. El test es basarà en l'humà, que haurà de anar formulant preguntes i la maquina les haurà de respondre. L'humà, al ser inconscient de que està parlant amb una maquina, haurà de jutjar si les respostes que rep són lògiques, o no. Si creu que ho són, es considerarà que la maquina en qüestió és intel·ligent. [11]

Bibliografia

- [1] FSF. Categories de privacitat. <http://www.gnu.org/philosophy/categories.ca.html>.
- [2] Anònim. 10 Ventajas del Software Libre y Propietario. <http://www.abadiadigital.com/diez-ventajas-del-software-libre-y-propietario/>.
- [3] Ataria. A brief history of the artificial intelligence. http://www.atariarchives.org/deli/artificial_intelligence.php.
- [4] Chromium. The chromium projects. <http://www.chromium.org/>.
- [5] Mozilla Corporation. About the mozilla corporation. <http://www.mozilla.org/en-US/foundation/moco/>.
- [6] Mozilla Corporation. The mozilla foundation. <http://www.mozilla.org/en-US/foundation/>.
- [7] Mozilla Corporation. The mozilla manifesto - mozilla. <http://www.mozilla.org/en-US/about/manifesto/>.
- [8] Miller K.W & Voas J. & Tom Costello. Free and Open Source Software. <http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5662568&url=http%3A%2F%2Fieeexplore.ieee.org%2Fiel5%2F6294%2F5662560%2F05662568.pdf%3Farnumber%3D5662568>, 2010.
- [9] Definicion.de. Inteligencia artificial. <http://www.definicion.de/inteligencia-artificial/>.
- [10] Boston Dynamics. Creacions de Boston Dynamics. https://www.youtube.com/results?search_query=boston+dynamics.
- [11] Enciclopedia. Test de turing. http://enciclopedia.us.es/index.php/Prueba_de_Turing.

- [12] Electronic Frontier Foundation. Electronic Frontier Foundation. <https://www.eff.org/about>.
- [13] Free Software Foundation. Free Software Foundation. <http://www.fsf.org/appeal/2009/freedom-is-the-goal>.
- [14] Linux Foundation. Linux foundation members. <http://www.linuxfoundation.org/about/members>.
- [15] The Apache Software Foundation. The apache license, version 2.0. <http://www.apache.org/licenses/LICENSE-2.0.html>.
- [16] The Apache Software Foundation. Web de la fundació de software apache. <http://httpd.apache.org/>.
- [17] Videolan Foundation. Vlc git repository. <https://github.com/torvalds/linux>.
- [18] Wikimedia Foundation. Free and Open Source Software - Adopters. https://en.wikipedia.org/wiki/Free_and_open-source_software#Adoption_by_governments.
- [19] Wikimedia Foundation. Inteligencia artificial. http://ca.wikipedia.org/wiki/Intel%C2%B7lig%C3%A8ncia_artificial.
- [20] Wikimedia Foundation. List of the largest software companies. http://en.wikipedia.org/wiki/List_of_the_largest_software_companies.
- [21] James W. McKie & Richard B. Franklin M. Fisher. *IBM and the U.S Data Processing Industry: an Economic History*. Praeger, 1983.
- [22] FSF. La llicència gpl. <http://www.gnu.org/copyleft/gpl.html>.
- [23] FSF. Vender software libre? <https://www.gnu.org/philosophy/selling.es.html>.
- [24] FSF. What is Free Software? <http://www.gnu.org/philosophy/free-sw.html>.
- [25] Wikipedia Fundation. Bsd. http://es.wikipedia.org/wiki/Berkeley_Software_Distribution.
- [26] Wikipedia Fundation. Copyright. http://es.wikipedia.org/wiki/Derecho_de_autor.

- [27] GNU. About the gnu operating system. <http://www.gnu.org/gnu/about-gnu.html>.
- [28] GNU. The gnu operating system. <http://www.gnu.org/gnu>.
- [29] Luis Lorenzo González. Software libre y software propietario, ventajas y desventajas. <http://www.gentegeek.com/sl-sp-ventajas-desventajas/>.
- [30] Ingeniatic. El triodo. <http://ingeniatic.euitt.upm.es/index.php/tecnologias/item/637-triodo>.
- [31] Open Source Initiative. La llicència mit. <http://opensource.org/licenses/MIT>.
- [32] Ivan. La maquina de turing. <http://www.slideshare.net/cchicaiza/mquina-de-turing>.
- [33] Linux. El rincon de linux. http://www.linux-es.org/sobre_linux.
- [34] Lina Marcela Cepeda Diaz & Grace Lennis Benavides. & Maria Cristina Lopez. Inteligencia artificial. <http://www.monografias.com/trabajos16/la-inteligencia-artificial/la-inteligencia-artificial.shtml>.
- [35] LynuxWorks. What's posix? <http://www.lynuxworks.com/products/posix/posix.php>.
- [36] GNU (GNU's not Unix). Gnu's not unix. <http://www.gnu.org/philosophy/philosophy.html>.
- [37] PBS off book. Explicació i opinió de la ia. <http://www.youtube.com/watch?v=53K1dMyslJg>.
- [38] Ramón M. Gómez Labrador (ramongomez@us.es). Definicions necessàries. <http://www.informatica.us.es/~ramon/articulos/LicenciasSoftware.pdf>.
- [39] TLDR. Explicació simplificada llicències. <http://www.tldrlegal.com/browse>.
- [40] Linus Torvalds. Linux github repository. <https://github.com/torvalds/linux>.
- [41] W3Schools. Os platform statistics. http://www.w3schools.com/browsers/browsers_os.asp.

- [42] W3Techs. Usage of operating systems for websites. http://w3techs.com/technologies/overview/operating_system/all.
- [43] David A. Wheeler. Why Free Open Source Software (FOSS)? http://www.dwheeler.com/oss_fs_why.html.
- [44] Wikipedia. Antecedents de la ia. http://es.wikipedia.org/wiki/Historia_de_la_inteligencia_artificial#Antecedentes.
- [45] Wikipedia. Chromium. <http://ca.wikipedia.org/wiki/Chromium>.
- [46] Wikipedia. Electronic Frontier Foundation. http://ca.wikipedia.org/wiki/Electronic_Frontier_Foundation.
- [47] Wikipedia. Historia de la ia. http://es.wikipedia.org/wiki/Historia_de_la_inteligencia_artificial.
- [48] Wikipedia. The manifest gnu. <http://www.gnu.org/gnu/about-gnu.html>.
- [49] How Stuff Works. What's a computer algorithm. <http://www.howstuffworks.com/question717.htm>.