

Sol·licitud de serveis informàtics per la Facultat de Ciències Biològiques

4 de maig de 2023

1 Introducció

Actualment la Facultat de Ciències Biològiques compta amb sis aules d'informàtica a la planta baixa de l'edifici B. Les aules estan equipades amb ordinadors que s'inicien en un entorn de Linux i poden executar aplicacions de MS Windows en un entorn virtual. Abans de l'inici de cada quadrimestre, el professorat sol·licita a les persones administradores quines aplicacions necessita instal·lades a l'entorn de MS Windows.

Aquesta manera d'oferir recursos informàtics a l'alumnat ha funcionat adequadament durant molts anys en què el paradigma d'**alfabetització informàtica** era el de poder utilitzar individualment programes informàtics privatis amb una interfície gràfica d'usuària. Amb l'expansió del programari lliure, la popularització d'alguns llenguatges de programació i l'adopció de tecnologies Web 2.0, les necessitats docents i les expectatives dels i de les estudiants han canviat. Als paràgrafs següents argumentem que el sistema de gestió de les aules d'informàtica ha quedat obsolet i necessita ser complementat, si no substituït, per recursos informàtics diferents.

2 Limitacions de les aules d'informàtica

2.1 Incapacitat d'administració del programari

Només podem sol·licitar la instal·lació o l'actualització de programari als ordinadors de les aules d'informàtica en dos moments a l'any: abans de l'inici de cada quadrimestre. Aquesta rigidesa té conseqüències negatives:

1. Els i les estudiants perden l'oportunitat d'aprendre a instal·lar el programari.

2. Perdem l'oportunitat d'utilitzar versions més recents dels programes amb un cicle ràpid d'actualitzacions.
3. No podem variar el programa d'activitats ni improvisar-ne de noves, si no és utilitzant els mateixos programes que ja hi ha instal·lats.

2.2 Disponibilitat estrictament local del programari

S'entén que un control rígid del programari instal·lat garanteix el respecte de llicències d'ús privatives. Siga com siga, cada vegada és més freqüent que els i les estudiants porten a l'aula d'informàtica els seus propis ordinadors portàtils, amb l'esperança de poder realitzar la pràctica en el seu propi ordinador. No hi ha dubte dels enormes beneficis didàctics de treballar amb l'ordinador propi, tant des de l'aula com des de casa. Actualment, aquesta possibilitat és terriblement incòmoda i frustrant, perquè els ordinadors particulars porten una diversitat de versions de diferents sistemes operatius, on la instal·lació del programari necessari pot incórrer en problemes de llicència o d'incompatibilitats. Quan, a pesar de tot, s'aconsegueix realitzar la pràctica amb l'ordinador propi, ens trobem amb la paradoxal situació d'estar infrautilitzant els recursos de la Universitat. En definitiva, les aules ofereixen un entorn homogeni i convenient, però al preu de no poder-se utilitzar més que allà mateix. Aquesta obvietat resulta ser un hàndicap important de cara a l'aprenentatge, i fa que l'opció cada vegada més freqüent d'utilitzar ordinadors propis siga preferible.

2.3 Obstrucció de l'ús de l'entorn Linux

Alguns programes instal·lats a l'entorn MS Windows podrien ser executats igualment en l'entorn Linux. Aquesta possibilitat és molt desitjable, perquè MS Windows s'executa en un entorn virtual dins de Linux, en una finestra, el marc de la qual ocupa un marge de la pantalla. Això redueix significativament la superfície útil de la pantalla i afecta la qualitat de les visualitzacions.

Altres programes senzillament només existeixen per a Linux. En aquest cas, no tenim opció d'utilitzar-los a les aules d'informàtica, tot i existir els recursos necessaris. Fins ara ha sigut impossible sol·licitar la instal·lació de programari docent en l'entorn Linux de les aules d'informàtica. Els administradors de les aules d'informàtica només atenen les peticions d'instal·lació de programari en MS Windows.

Certament, la persona usuària sempre té la possibilitat d'instal·lar programari dins la seua pròpia carpeta, on gaudeix de permissos d'escriptura. Per exemple, no és impossible instal·lar el gestor de paquets `conda`, i amb

ell instal·lar qualsevol altre programa en una carpeta local. Tanmateix, el procés pot prendre molts minuts de classe, i s'hauria de repetir a cada sessió, perquè les modificacions a l'entorn d'usuària no són permanents.

Considerem que a les aules d'informàtica de la Facultat de Biologia és necessari poder treballar i ensenyar en un entorn Linux.

2.4 Alienació de l'estudiant

En enfrontar-se a un sistema operatiu hermètic, sobre el qual la persona usuària no té cap control, l'actitud esdevé la d'un client cap a un proveïdor, no la d'un estudiant universitari. Des d'un punt de vista didàctic, l'entorn MS Windows i totes les aplicacions amb interfície gràfica d'usuària són nefastes pels motius següents:

1. Cancel·len la curiositat, en tant que el programari funciona com una caixa negra, que respon als *clicks* sense revelar-ne el com. Com per art de màgia.
2. Afegeix una càrrega cognitiva completament inútil: aprendre l'ús d'una interfície gràfica concreta (on són els menús, quins botons fan què, com es configura, etc.) no té cap altra aplicació en absolut.
3. Les interfícies gràfiques d'usuària són llenguatges analògics, no generatius. Igual que si ens comunicàrem per gestos, el que podem comunicar a un ordinador amb una interfície gràfica és extremadament limitat.
4. Manipular un ordinador amb interfícies gràfiques generalment implica renunciar a la reproduïbilitat: no guardem registre d'on i quan hem fet *click*. I per tant, reproduir un resultat o explicar com l'hem obtingut esdevé sovint una tasca impossible. Res més lluny del que hauria de ser l'ensenyança universitària.

3 Proposta de serveis informàtics

La proposta gira entorn a un canvi de paradigma, en què el protagonisme passa del programari a la persona usuària. En primer lloc, cal posar en valor els **llenguatges de programació interpretats** com a mitjà (més que com a fi) de qualsevol aprenentatge. Els llenguatges de programació interpretats es caracteritzen per la possibilitat d'especificar les ordres en un format de text pla, que pot ser interpretat i executat en qualsevol maquinari. El fet de ser un **estàndard** universal, fa del text pla un vehicle fantàstic de tota classe

d'instruccions i comunicacions. Quan ens comuniquem amb els ordinadors a través de text pla, és a dir, a través d'una línia de comandaments o d'un *script*, tenim tots els avantatges que les interfícies gràfiques ens neguen:

1. Les ordres o comandaments són transparents: generen un resultat i alhora mostren com ho fan. No hi ha, per tant, barreres a la curiositat.
2. La càrrega cognitiva d'aprendre a programar és certament molt major que la d'aprendre a fer *clicks*, però no és inútil en absolut, sinó que fins i tot un coneixement molt superficial pot capacitar per fer moltes coses, en qualsevol assignatura i en la vida professional.
3. Un llenguatge de programació és difícil d'aprendre perquè la relació entre el significat i el significant és arbitrària, exactament igual que en un llenguatge natural. I això és, precisament, el que fa que el llenguatge siga *generatiu*: les possibilitats de creació de significats són infinites. La programació és una habilitat sobre la qual es pot *construir*, mentre que l'ús d'una interfície gràfica, no.
4. Les ordres especificades en text pla són fàcilment guardades en documents o *scripts*, que registren i permeten documentar tot allò que li demanem a l'ordinador que faci. És a dir, l'ús d'un llenguatge de programació pràcticament ens *obliga* a registrar la manera en què hem obtingut un resultat i ens facilita enormement compartir i col·laborar amb altres en l'obtenció de resultats, qualsevol que siga la naturalesa de la tasca.

Amb l'objectiu de beneficiar la comunitat universitària de tots aquests avantatges, a continuació proposem la implementació de nous serveis informàtics.

3.1 Facilitar l'ús de l'entorn Linux

3.1.1 De manera local

En primer lloc, el canvi més immediat que es pot implementar per començar a solucionar el problema és oferir la possibilitat d'instal·lar programari en l'entorn de Linux. Si més no, de moment, en les mateixes finestres temporals en què se'ns permet sol·licitar la instal·lació de programari en MS Windows.

Gran part del problema se solucionaria si a l'entorn Linux tiguérem instal·lada alguna cosa més que l'estrictament necessari. Algunes eines de programari lliure són d'aplicació molt general en qualsevol assignatura, com per exemple: *git*, *L^AT_EX*, *R*, *Python* i *conda*, per esmentar-ne uns pocs.

3.1.2 De manera remota

Idealment, els i les estudiants haurien de poder accedir remotament, per `ssh`, a un servidor Linux. Fins i tot si l'accés es realitzara exclusivament a través del terminal, sense entorn gràfic, aquesta experiència seria molt valuosa per a l'aprenentatge. Si cada persona usuària tinguera el seu directori propi i els canvis foren permanents, el servidor podria utilitzar-se a manera de núvol. A més de poder accedir des de casa com des de l'aula d'informàtica, els i les estudiants tindrien facilitat per accedir a recursos comuns del servidor, com ara dades amb les quals realitzar exercicis pràctics.

Ens consta que alguns centres de la Universitat de València disposen ja d'un servidor de càlcul per a finalitats docents. Actualment les ciències biològiques tenen tanta necessitat de computació com qualsevol altra disciplina, o més. Per tant, sol·licitem la dedicació d'un servidor de càlcul a tasques docents de la facultat.

3.2 RStudio-server

L'ús d'*scripts* en la docència amb ordinadors és fonamental. Els *scripts* són documents vius, que poden ser editats durant una pràctica i poden convertir-se també en els apunts o anotacions amb què els i les estudiants conserven allò que han après. Actualment existeix una varietat de formats capaços de combinar en un mateix document les instruccions que donem a l'ordinador, els resultats d'aquestes instruccions si són imprimibles (figures, taules o resultats numèrics), i també les nostres anotacions o comentaris que documenten o expliquen l'anàlisi i els resultats. Una de les modalitats són els documents Rmarkdown, propis del llenguatge R però capaços d'interpretar ordres de Python, Bash, etc. Els documents Rmarkdown de text pla generen documents en format PDF o HTML en ser compilats.

El programa RStudio té una versió "servidor", RStudio-server, que permet l'ús compartit del servidor de càlcul a través d'una interfície típica de RStudio en una pestanya del navegador. Aquesta possibilitat no només facilita l'ús i l'edició documents Rmarkdown, sinó que permetria a tots els membres d'un grup executar les anàlisis o els seus *scripts* en un mateix ambient, amb els mateixos paquets instal·lats i amb accés a les mateixes dades.

3.3 Jupyter Hub

Una altra modalitat de documents executables són els *Jupyter notebooks* ("quaderns Jupyter" en avant), que s'editen i es visualitzen a través del navegador, a manera d'una aplicació web, i tenen l'avantatge de ser més dinàmics:

pots actualitzar seccions concretes del document sense haver de compilar-lo tot sencer. Els quaderns Jupyter accepten instruccions en multitud de llenguatges de programació, incloent Python, Julia, R, Bash, etc.

Aquests documents tenen potencials didàctics enormes [1]. Un dels seus avantatges és la facilitat amb què poden ser compartits. En estar basats en programari lliure i formats estàndard, són documents dissenyats per la difusió i la col·laboració. Actualment existeixen solucions de programari lliure per oferir de forma remota accés als entorns de computació virtuals on resideixen i poden ser actualitzats i executats aquests documents. El programari que fa possible compartir un conjunt de quaderns Jupyter amb una classe o un grup de persones col·laboradores és **Jupyter Hub** [2].

El que sol·licitem a la Universitat de València és que dedique els recursos de computació necessaris per instal·lar i executar Jupyter Hub, de tal manera que qualsevol membre de la comunitat universitària pugui connectar-se a les instàncies prèviament definides pel professorat. Com a exemple, es pot accedir a l'enllaç següent, que inicia un entorn de computació virtual accessible des del navegador i que conté un parell de quaderns Jupyter:

<https://mybinder.org/v2/gh/IgnasiLucas/Khi2/soca>

L'enllaç anterior ens condueix a una instància de Jupyter Hub que ens ofereix gratuïtament l'empresa MyBinder. Si la Universitat de València ofereix el mateix servei, gaudiríem d'una major estabilitat del servei (que a vegades se satura) i d'un accés privat amb identificació d'usuària. L'ús de les instàncies de MyBinder s'ha intentat en cursos anteriors, però sol presentar problemes de saturació del servei.

3.4 Servidor de git

El programa git és un sistema de control de versions. S'utilitza intensament en l'àmbit del desenvolupament de programari. Serveix principalment per mantenir fora de la vista però ben guardades les versions anteriors d'un conjunt de documents, en una carpeta de l'ordinador. Això ajuda enormement a tenir els arxius propis ben ordenats i evitar situacions com aquella popularitzada en la tira còmica "**Final.doc**". A més, git guarda registre de qui va introduir quins canvis i quan, la qual cosa facilita molt la col·laboració o l'edició conjunta de documents.

Tot i haver estat concebut per ajudar al desenvolupament de programari, resulta que git troba aplicació en qualsevol conjunt de documents que han de ser editats al llarg del temps, per una o més persones. En l'àmbit universitari, aquests documents poden ser: un treball de classe qualsevol, una tesi, o uns materials didàctics. Tot i que la corba d'aprenentatge no és suau, un coneixement bàsic de git permet gaudir de la major part dels seus beneficis.

Tot i que git pot executar-se localment en l'ordinador particular, gran part de la seua utilitat s'aprecia en els treballs col·laboratius. En aquest cas, és necessari un servidor de git remot, al qual totes les persones participants d'un projecte han de poder connectar-se, be per descarregar les actualitzacions o bé per proposar-ne de noves. El mateix programa ofereix mecanismes per evitar i gestionar els possibles conflictes entre versions diferents d'un mateix document. L'exemple més popular de servidor de git és la plataforma **GitHub**, que ofereix plans gratuïts per compartir carpetes de forma pública. A l'enllaç següent es pot accedir a una carpeta en GitHub que conté materials didàctics:

https://github.com/IgnasiLucas/Bioinformatica_33190

La Universitat de València hauria d'oferir el seu propi servidor de git, igual que ofereix el servei de núvol o de correu electrònic. Moltes altres universitats ho fan. D'aquesta manera, les carpetes podrien ser compartides més fàcilment només dins l'àmbit de la universitat, i sense haver de recórrer a serveis externs.

4 Conclusions

Les aules d'informàtica no cobreixen les necessitats d'aprenentatge de les generacions actuals d'estudiants. En qualsevol carrera professional és cada vegada més habitual haver de tenir nocions de programació. Tal com passa amb els llenguatges naturals, els llenguatges de programació s'aprenen millor en el context del seu ús, que com a objectiu de l'aprenentatge. Si a la Facultat de Ciències Biològiques disposem de les infraestructures sol·licitades, serà més fàcil que diferents assignatures es coordinen per utilitzar el mateix llenguatge de programació a les pràctiques amb ordinador. Així, sense haver d'impartir més que les nocions bàsiques, i sense pretendre que les persones graduades esdevinguen també expertes programadores, haver-los exposat sistemàticament a un llenguatge de programació repercutirà molt positivament en la seua preparació:

1. La familiaritat mínima amb una línia de comandaments desmitifica la tecnologia, proporciona seguretat i possibilita l'aprenentatge posterior.
2. Havent entés com funciona un llenguatge de programació és molt més fàcil aprendre'n un altre. És a dir, allò essencial no és triar el llenguatge del futur, sinó fer-los treballar amb un dels llenguatges disponibles.
3. L'edició i la gestió d'*scripts*, així com l'ús de git promou els bons hàbits de treball amb ordinador, essencials per preservar la integritat de les

dades, la reproduïbilitat dels resultats i la traçabilitat dels canvis i de l'autoria dels documents.

En definitiva, demanem actualitzar el concepte d'alfabetització informàtica per incloure nocions de programació. De la mateixa manera que tots aprenem a llegir i escriure, sense necessitat d'arribar a ser escriptors o escriptores professionals, igualment tots ens podem beneficiar de saber programar, sense necessitat d'arribar a ser enginyers o enginyeres informàtiques [3].

Per poder aconseguir-ho, necessitem que la Universitat actualitze els seus recursos informàtics. Ja no n'hi ha prou amb una aula d'informàtica que dóna accés a uns pocs programes privatis en un lloc i unes hores concretes. Aquest model caducat seria comparable a tenir un museu de llibres en lloc d'una biblioteca.

Referències

- [1] L. Barba, L. Barker, D. Blank, J. Brown, A. Downey, T. George, L. Heagy, K. Mandli, J. Moore, D. Lippert, K. Niemeyer, R. Watkins, R. West, E. Wickes, C. Willing, and M. Zingale, *Teaching and learning with Jupyter*. Bookdown, 2019.
- [2] Project Jupyter, “Jupyter hub.” <https://jupyter.org/hub>, 2023. Accessed on 2023-05-04.
- [3] A. Vee, *Coding Literacy. How computer programming is changing writing*. The MIT Press, 2017.