

FUNDAMENTACIÓN DEL USO DE SOFTWARE LIBRE EN LA UNIVERSIDAD PÚBLICA. ENSEÑANDO MATEMÁTICA CON HERRAMIENTAS ALTERNATIVAS

A. Pardini

*UID IMApEC – Universidad Nacional de La Plata. Departamento de Fisicomatemática.
pardo.bssso @ gmail.com*

Resumen

En el presente trabajo se caracterizará al Software Libre, sus ventajas técnicas, morales y pedagógicas por sobre sus contrapartes privativas. Es mi convicción que la Universidad pública debe enseñar métodos de trabajo y no a utilizar una herramienta en particular; principalmente porque los programas de mayor difusión en el ámbito académico no permiten su distribución ni el estudio de su funcionamiento interno. Se presentarán los principales reemplazos a MATLAB y Maple (SAGE, Python+NumPy y Maxima) y se dará una introducción a estos con ejemplos reales.

Palabras clave: Matemática. Educación asistida por ordenador. Técnicas alternativas.

INTRODUCCIÓN

El avance en la comunidad científica siempre fue acompañado por la libre circulación de las ideas y el conocimiento; entonces, ¿por qué educamos usando herramientas informáticas cuyo modelo de distribución no solo fomenta todo lo contrario sino que tampoco nos permite estudiar su funcionamiento? El ámbito académico supuestamente tiene que favorecer la creatividad, innovación, aprendizaje, cooperación y libre flujo de ideas, sin embargo si enseñamos utilizando herramientas privativas estamos atentando contra estos ideales. El modelo de software privativo trae consigo varias desventajas. Una es su licenciamiento, leonino a veces, que nos impide distribuirlo; la universidad puede costear el pago de licencias pero un alumno no. Esto conlleva a la piratería, dado que se exige el uso de un programa específico, el alumno debe tenerlo para realizar las actividades fuera del horario de taller; la única forma de lograr esto para casi la totalidad del alumnado consiste en obtener copias no legítimas. Otra desventaja del software privativo, la mas importante desde la perspectiva del educador, es que NO PODEMOS ver (ni estudiar) cómo funciona. Estamos trabajando con 'cajas negras', nos esta vedado saber cómo y por qué hacen lo que hacen. ¿Qué clase de profesionales formamos si les enseñamos a confiar en herramientas que no pueden estudiar ni modificar para que se adapten a sus necesidades? Aquí es donde entra el software libre. No sólo simboliza un ahorro económico, también representa el retorno del control hacia los usuarios. No es una herramienta que alquilamos (es lo que hacemos al pagar la licencia todos los años) y

se nos restringe su uso, es **NUESTRA HERRAMIENTA** y podemos hacer lo que queramos con ella: estudiarla, modificarla, mejorarla, combinarla con otras y fundamentalmente compartirla.

DESARROLLO

EL SOFTWARE LIBRE

Por definición (OSI / FSF[1,2]) un producto de software se considera libre si le brinda a sus usuarios cuatro derechos fundamentales:

1. - La posibilidad de usar el software para cualquier fin y bajo cualquier circunstancia. No posee cláusulas del estilo "Usted no puede utilizar este programa en Cuba".
2. - Uno puede estudiar sin ningún tipo de restricción el funcionamiento del mismo (para eso se necesita tener acceso al código fuente).
3. - Puede ser modificado.
4. - Pueden distribuirse sin ningún tipo de impedimento, el programa original así como también las modificaciones.

VENTAJAS DE USAR SOFTWARE LIBRE EN LA EDUCACIÓN (construyamos el auto en lugar de reinventar la rueda)

Desarrollo comunitario y abierto

Como punto de partida, el software libre (SL en adelante) se desarrolla con un modelo totalmente distinto al que se impuso como 'tradicional'. El software no es ya un producto terminado que una corporación nos entrega y como usuarios nos limitamos a aceptarlo tal como está y usarlo sin plantearnos la posibilidad de amoldarlo a nuestras necesidades o de influir en su desarrollo futuro. Los proyectos de SL se crean y desarrollan en comunidad, cualquiera puede contribuir sin necesidad de ser programador, la experiencia de los usuarios y sus sugerencias tienen influencia decisiva en el rumbo del proyecto. El alumnado no solo puede aprender estudiando el funcionamiento de programas que se utilizan a diario en líneas de producción sino que también pueden reutilizar partes de estos para sus propios proyectos. Tanto alumnos como profesores pueden modificar y extender el software para cubrir sus necesidades y pueden corregirlo de ser necesario.

Independencia económica y técnica

El SL es accesible a todos por naturaleza. Por un lado no se plantea la situación de utilizar copias ilegítimas al no poder costear las licencias y por el otro uno no depende de las decisiones de la empresa; si ésta deja de existir, dar soporte al producto o simplemente no considera que nuestras necesidades sean afines a las de la mayoría de la comunidad podemos modificar nosotros mismos el programa o contratar a un programador local para que realice la tarea.

Libertad de distribución

El profesor puede sin ningún impedimento legal realizar una compilación de los programas y ejercicios utilizados en su curso y repartirlo entre sus estudiantes para que los usen en la cantidad de equipos que deseen. Compárese con la mayoría de las licencias, las cuales desde ya prohíben la copia de los programas, ponen un tope a la cantidad de instalaciones y tienen duración limitada.

EL SL. EN EL AULA: REEMPLAZANDO MATLAB Y MAPLE

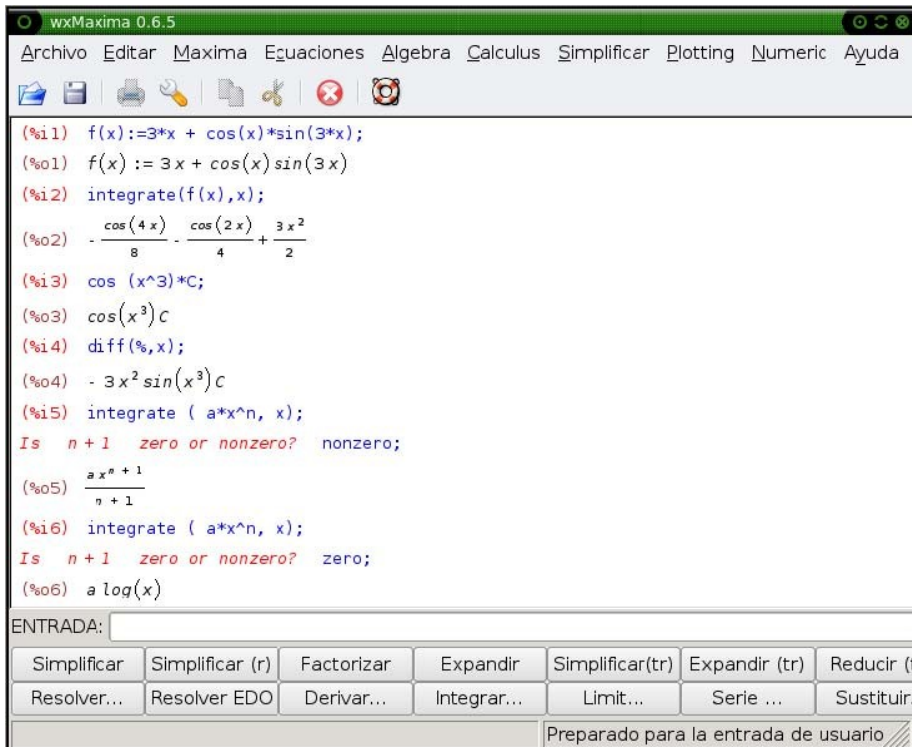
Luego de enunciadas las ventajas del SL pasaré a dar una introducción a los programas que ya están en condiciones de ser utilizados en el aula.

Los paquetes elegidos son Maxima, SAGE y Python con las librerías SciPy, matplotlib y PyLab.

MAXIMA[3]:

Es un sistema para la manipulación de expresiones numéricas y simbólicas. Entre otras tareas permite realizar derivadas, integrales, series de Taylor, transformadas de Laplace, resolver ecuaciones diferenciales y sistemas, calcular límites, trabajar con vectores, listas, matrices y tensores. Tiene capacidad para trabajar con números enteros y de punto flotante con precisión arbitraria. Realiza gráficas en dos y tres dimensiones y permite exportar los resultados a Tex. Maxima es la continuación libre del proyecto Macsyma, iniciado en los 60 en el MIT, el cual fue el inspirador de Mathcad y Maple entre otros paquetes similares. Posee varias interfaces, en este caso mostraremos wxMaxima[4] por ser la más consistente en GNU/Linux y Windows pero las versiones para TeXmacs[5] y Kayali[6] también merecen consideración. Hay además, interfaces web, que permiten trabajar desde cualquier sitio con acceso a internet. Personalmente la que más me gustó es WebMathematics Interactive[7].

Integrando y derivando:



The screenshot shows the wxMaxima 0.6.5 interface. The menu bar includes Archivo, Editar, Maxima, Ecuaciones, Algebra, Calculus, Simplificar, Plotting, Numeric, and Ayuda. The toolbar contains icons for file operations and editing. The main window displays the following commands and results:

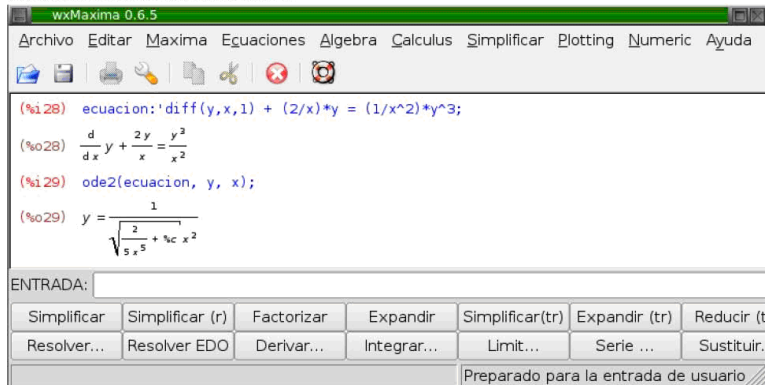
```
(%i1) f(x):=3*x + cos(x)*sin(3*x);
(%o1) f(x) := 3x + cos(x) sin(3x)
(%i2) integrate(f(x),x);
(%o2) 
$$-\frac{\cos(4x)}{8} - \frac{\cos(2x)}{4} + \frac{3x^2}{2}$$

(%i3) cos(x^3)*C;
(%o3)  $\cos(x^3)C$ 
(%i4) diff(%o3,x);
(%o4)  $-3x^2 \sin(x^3)C$ 
(%i5) integrate(a*x^n, x);
Is n+1 zero or nonzero? nonzero;
(%o5)  $\frac{ax^n + 1}{n + 1}$ 
(%i6) integrate(a*x^n, x);
Is n+1 zero or nonzero? zero;
(%o6)  $a \log(x)$ 
```

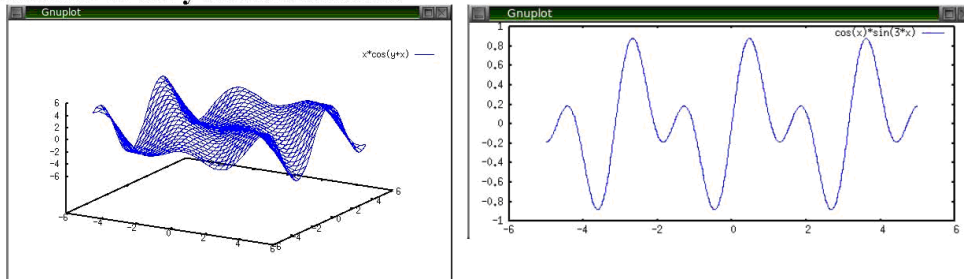
Below the main window is an "ENTRADA:" field and a grid of buttons: Simplificar, Simplificar (r), Factorizar, Expandir, Simplificar(tr), Expandir (tr), Reducir (t), Resolver..., Resolver EDO, Derivar..., Integrar..., Limit..., Serie ..., and Sustituir. The status bar at the bottom right reads "Preparado para la entrada de usuario".

Ecuaciones diferenciales:

Ecuaciones diferenciales:



Gráficas en dos y tres dimensiones:



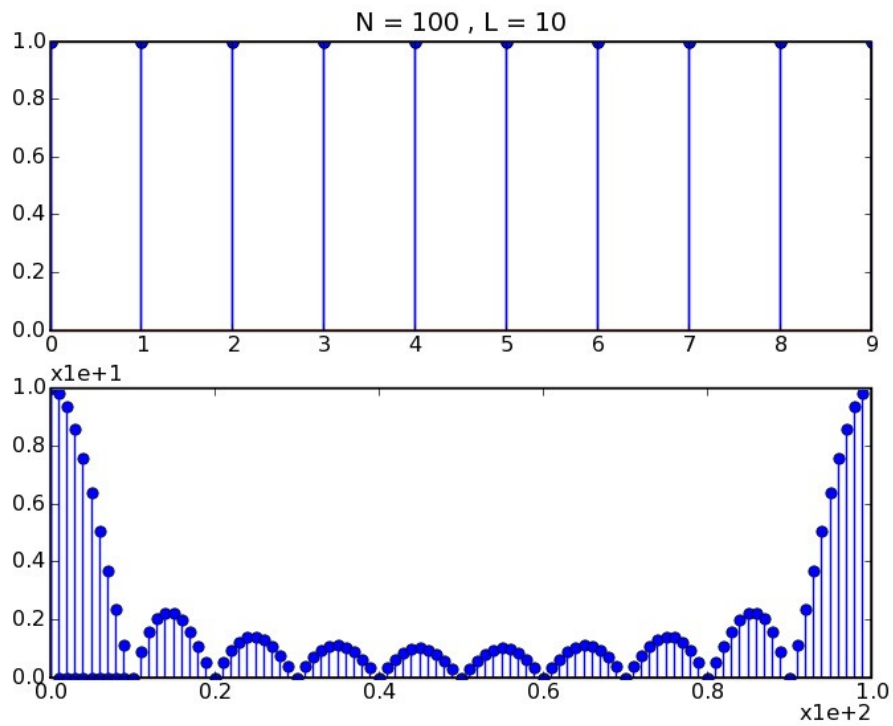
PYTHON + SCIPY + MATPLOTLIB/PYLAB:

Python[9] es un lenguaje interpretado de programación multiparadigma (orientado a objetos, procedimientos y funciones), muy portable (corre, por ejemplo, en GNU/Linux, MacOS, Windows, Palms y celulares Nokia) y de por si muy poderoso (SAGE esta construido con Python y estos ejemplos fueron probados allí). SciPy[13] es un módulo que permite trabajar con arreglos multidimensionales (también sparse), realizar integraciones, optimizaciones, análisis de Fourier e incluso integrar código en Fortran.

Matplotlib[14] es una librería para realizar graficas en 2 dimensiones, incluye pylab, una capa que la vuelve compatible con la sintaxis de MATLAB. Como ejemplo vamos a calcular y graficar una transformada discreta de Fourier con 100 puntos:

```
from pylab import * L = 10 x = ones( L ) # Figura 1: N = 100 L = 10 N = 100 L = 10 X =
fft(x, N ) figure(1) subplot(211) title('N = 100 , L = 10') horiz = arange(0,L,1) stem( horiz, x )
subplot(212) horiz = arange(0,N,1) stem( horiz, abs(X) ) savefig( 'figura1.png' ) show()
```

SAGE[8]: SAGE es un paquete pensado principalmente para la enseñanza e investigación de álgebra, geometría y teoría de números (en especial anillos [ring theory]). En su mayoría está programado en Python[9], lo cual trae ciertas ventajas:

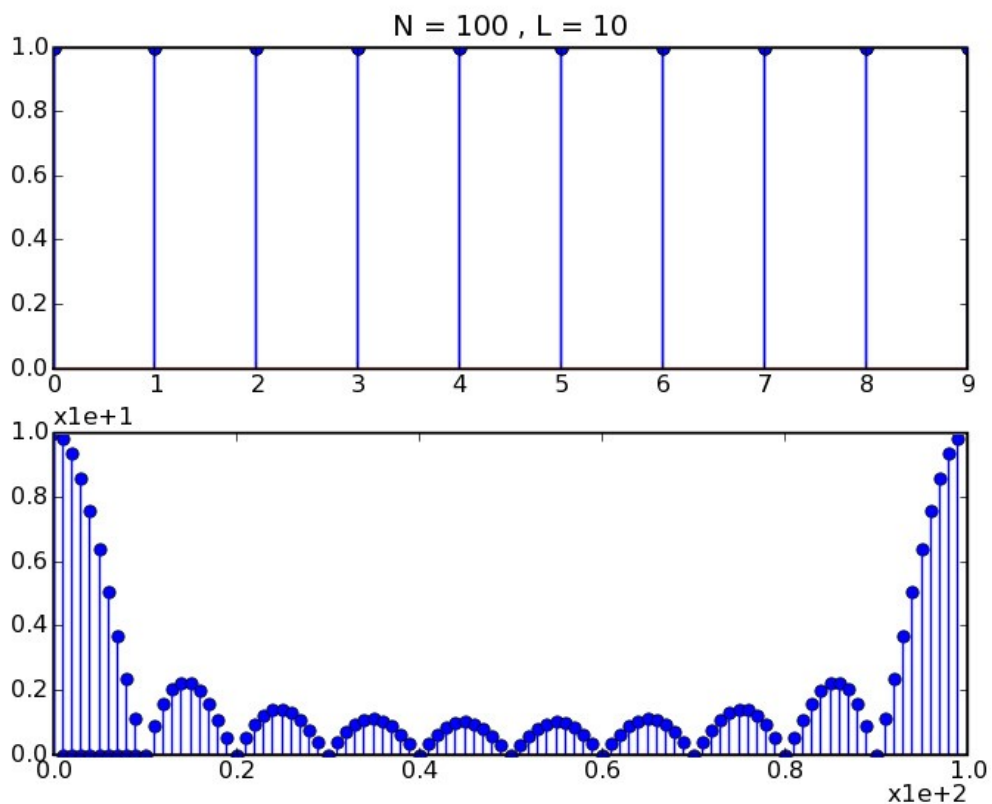


SAGE[8]:

SAGE es un paquete pensado principalmente para la enseñanza e investigación de álgebra, geometría y teoría de números (en especial anillos [ring theory]).

En su mayoría está programado en Python[9], lo cual trae ciertas ventajas:

1. - Es muy portable.



SAGE puede trabajar en modo batch (ejecutando un archivo con instrucciones), en forma interactiva desde una consola o por una interfaz web, la más indicada para un curso a distancia. Posee un muy buen soporte para integración con otros sistemas de álgebra tanto libres como propietarios, entre ellos Maxima[3], MATLAB, Singular[10], PARI/GP[11] o Magma[12].

La interfaz web permite que varias personas [usuarios] realicen documentos y publiquen sus trabajos en internet. Posee un sistema de control de revisiones y edición colaborativa de documentos (varias personas pueden trabajar sobre el mismo proyecto).

Las publicaciones no son estáticas, tanto un visitante (con una cuenta de invitado) como el propietario de un documento pueden interactuar con el mismo y modificarlo (sin embargo, el visitante solo puede editar una copia local).

Captura de un ejemplo, el trabajo se realizó en También podemos usar código Python para una pc con WindowsXP y SAGE corriendo en simplificar las cosas (al ser un lenguaje otra pc con Ubuntu gnu+linux multiparadigma podemos encapsular entidades matemáticas en un objeto) y utilizar su semántica para crear construcciones. Ejemplo:

```
#creo una lista de puntos (0<199)
L = [[cos(pi*i/100),sin(pi*i/100)] for i in range(200)]
#con esa lista creo un objeto polygon
p = polygon(L, rgbcolor=(3/8,3/4,1/2))
#y lo muestro, p.save(archivo) lo guarda p.show()
```

SDGE Notebook invitado | Home | Published | Log | Help | Sign out

Ejemplo 1
 last edited on September 19, 2007 12:57 AM by invitado

Save Save & close Discard changes

File... Action...
 Data... sage

Print Use Edit Text Revisions
 Share Publish

Ejemplo de una hoja en SAGE

Esta es una hoja de ejemplo, el texto se ingresa en formato HTML. Asimismo, puede incluirse código latex, $\frac{d^4 \sin(x^2)}{dx^4}$ se convierte en: $\frac{d^4 \sin(x^2)}{dx^4}$

Integrando y derivando

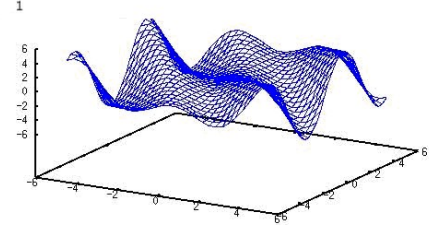
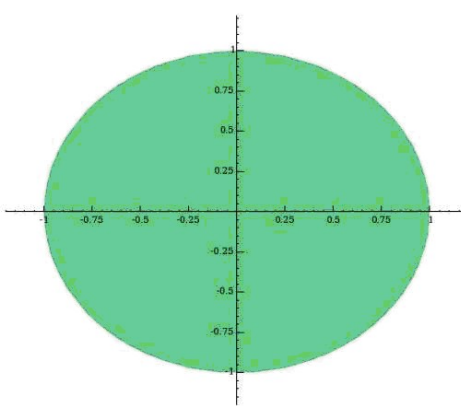
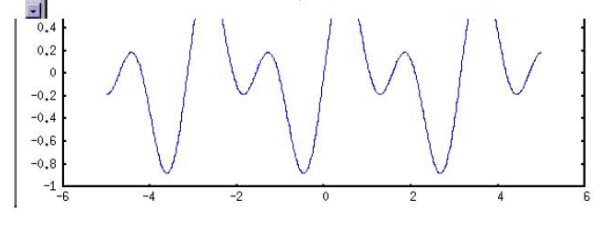
```
g = diff( x*sin(x^3), x ); g
      sin(x^3) + 3*x^3*cos(x^3)
```

Haciendo view(g) se ve mejor:
 $\sin(x^3) + 3 \cdot x^3 \cdot \cos(x^3)$

```
G = integral(g, x); view(G)
      x * sin(x^3)
```

```
integral( exp(-x), x, 0, oo )
```

1

DISCUSIÓN

Contamos hoy en día con herramientas alternativas libres de nivel más que suficiente para los cursos académicos, las principales barreras para su adopción son la desinformación y el rechazo al cambio. La comunidad del SL aún no se introdujo por completo en la educación pública [15] y menos en el ámbito universitario, sin embargo en forma extracurricular es algo común [16]. Como miembro activo trabajo para que esta situación se extienda también a las cursadas normales, para el 2008 tenemos pensado (junto con alumnos de la UBA[17] y el naciente Centro Platense de Referencia e Impulso de SL) hacer una propuesta formal para realizar en forma paralela todas las actividades exclusivamente con SL (incluyendo las resoluciones a guías de trabajo actuales y guías específicas).

SOBRE EL IMAPEC:

IMApEC (Investigación en Metodologías Alternativas para la Enseñanza de las Ciencias) es una unidad que busca la innovación en la enseñanza de las ciencias y la aplicación de nuevas tecnologías. Principalmente trata de optimizar los primeros niveles de los recursos humanos en la ciencia y la tecnología a través de:

1. - *Innovaciones de desarrollos en el área de la educación basada en diferentes investigaciones didácticas.*
2. - *Implementación de nuevas tecnologías de comunicación.*
3. - *Modelos de interacción que toman lugar en el aula usando diferentes herramientas didácticas, entre ellas la simulación por computadora. También creamos programas informáticos para la enseñanza de tópicos específicos.*

BIBLIOGRAFÍA

1. *FSF Free Software Foundation ; <http://www.fsf.org>*
2. *OSI Open Source Initiative ; <http://www.opensource.org>*
3. *Maxima, a computer algebra system ; <http://maxima.sourceforge.net>*
4. *wxMaxima, a cross-platform GUI for Maxima ; <http://wxmaxima.sourceforge.net>*
5. *GNU TeXmacs, free WYSIWYG editing platform with features for scientists ; <http://www.texmacs.org>*
6. *Kayali, a QT based Computer Algebra System ; <http://kayali.sourceforge.net>*
7. *WebMathematics Interactive ; <http://wmi.math.u-szeged.hu/wmi/math.php?skin=blue>*
8. *SAGE Mathematical Software, Version 2.8 ; <http://www.sagemath.org>*
9. *The Python Language ; <http://www.python.org>*
10. *Singular, a Computer Algebra System for Polynomial Computations ; <http://www.singular.uni-kl.de>*
11. *PARI/GP ; <http://pari.math.u-bordeaux.fr>*
12. *Magma ; <http://magma.maths.usyd.edu.au/magma>*
13. *SciPy, open-source software for mathematics, science and engineering ; <http://www.scipy.org>*

14. Matplotlib, a python 2D plotting library ; <http://matplotlib.sourceforge.net>
15. Gleducar, construcción cooperativa de conocimientos ; <http://wiki.gleducar.org.ar/wiki/Portada>
16. Curso de Métodos computacionales para científicos ; <http://curso.iflysib.unlp.edu.ar>
17. UBA, Universidad de Buenos Aires ; <http://www.uba.ar>