

Intérprete de órdenes (shell) GNU/Linux. Lenguaje de control. Comandos combinados. Shell Scripts. Sistemas de Archivos. Procesos, sincronización e intercomunicación. Manejo de Memoria. Primitivas. Administración de usuarios, recursos y servicios básicos. Herramientas: Debian (Huayra / Ubuntu), bash y Python.

Redes de computadoras (GNU/Linux): Modelo de Referencia de Interconexión de Sistemas Abiertos (OSI). Ethernet y WiFi (estándares IEEE). Pila de Protocolos TCP/IP. Sockets. Aplicaciones de internet (sistema de nombres de dominio, terminales remotas, transferencias de archivos, correo electrónico, web, etc.). Direccionamiento y Enrutamiento. Configuración y administración básica de servidores. Herramientas: Debian y Python (entre otras).

Ingeniería del Software Libre: Introducción a la Ingeniería de Software. Gestión de proyectos (PM). Metodologías Ágiles (Scrum). Procesos de Software Personal (PSP). Diseño Estructurado y Orientado a Objetos (DFD, UML). Métricas. Control de Calidad y Pruebas. Mejora continua. Sistemas de Control de Versiones Distribuidos (DVCS). Aspectos Legales y Económicos del Software Libre. Gestión del Software. Documentación. Portales y Comunidades Virtuales (SourceForge, GoogleCode, GitHub).. Herramientas: Python, Mercurial (entre otras).

Desarrollo de Aplicaciones Web (PHP + MySQL, Python -web2py- + PostgreSQL): Introducción a las aplicaciones web. Diseño de páginas web (HTML5 + CSS). Contenido dinámico. CGI. Desarrollo web con PHP y MySQL (Ejemplos Prácticos). Introducción a la programación web avanzada. Tecnologías y lenguajes de cliente y servidor. AJAX. Frameworks modernos: web2py (MVC). Aplicaciones de Internet Enriquecidas. Servicios Web XML / JSON.

Distribución de la Carga horaria estimada por cuatrimestre:

1°	Alg. y Programación I (128hs: 64hs teoría + 64hs práctica)	Bases de Datos (64hs)	Redes de Comp. (64hs)	Seminario de Sw. Libre (64hs)
2°	Programación II (128hs: 64hs teoría + 64hs práctica)	Sistemas Operativos (64hs)	Desarrollo Web (64hs)	Ing. del Sw. Libre (64hs)

Diseño Curricular

El modelo curricular estará enfocado en el educando³⁰, basado en estudios, trabajos, investigaciones y proyectos individuales e independientes (método de proyectos). El objetivo es que el alumno aprende haciendo, construyendo, resolviendo problemas.

Si bien hay una división de los contenidos en distintos cursos, no son materias ni disciplinas aisladas, para evitar temas demasiado abstractos desconectados de la realidad del alumnado (a la usanza de “escuela tradicional” solo con conocimientos acumulativos lógicamente estructurados).

Esto no implica que se busquen materias o disciplinas fusionadas (que podría ser aquellas que incluirían varias materias tradicionales, por ej. “Administración de Servidores GNU/Linux” de otros planes de estudio) y tampoco se organiza por temas

³⁰ TRALDI, Lady Lina (1984) Currículo. San Pablo, Brasil: Ed. Atlas.

o tópicos (con un foco meramente técnico, por ej. “Técnicas avanzadas de programación en Android” o “Desarrollo de aplicaciones empresariales para internet usando Jboss Seam 2.2.2” en otras diplomaturas), ya que dichos enfoques se alejaría de los estándares sugeridos en nuestro país por el Ministerio de Educación y la Red de Universidades Nacionales con Carreras en Informática.

En conclusión, para lograr una unidad conceptual y no producir un aislamiento a los alumnos (analizado en secciones anteriores), se ha buscado un diseño curricular más orientado a un núcleo básico o “core” integrado, con una formación común y campos del conocimiento de las TICs lógicamente estructurados correlacionados, con transversalidad en torno al software libre, sus problemáticas y posibilidades.

Correlatividades

Al no existir gradualidad (la mayoría de los contenidos pueden cursarse en paralelo en un año, sin una marcada secuencialidad temporal como otros tipos de estudios), se recomienda orientación y seguimiento personalizado a los alumnos mediante tutorías continuas (en especial al momento de matriculación), para contemplar las diferencias individuales en términos de capacidades, habilidades, progresos, ritmos, velocidades y madurez; sobre todo entendiendo que esta propuesta busca ser más accesible e incluir a estudiantes egresados del nivel secundario y/o de otras áreas diversas ajenas a la informática, sin un necesario trasfondo como el requerido en otras especializaciones y carreras de grado o posgrado.

Igualmente, para simplificar el acompañamiento y disminuir los tiempos/recursos necesarios, se han organizado los módulos (unidades y bibliografía) con un esquema de correlatividades sugeridas (ver cuadro **Distribución de la Carga horaria**)

En caso de que el alumno no respete esas correlatividades sugeridas, deberá profundizar los contenidos necesarios para cada curso de manera individual sin acompañamiento docente.

Evaluación:

Si bien la evaluación se dará de manera continua durante todos los cursos, se planificarán varias instancias de evaluación mediante Trabajos Prácticos, y un Examen al finalizar cada curso. Los alumnos que cumplan con las consignas hechas por el docente durante la etapa de seguimiento y evaluación continua (investigaciones, debates en el foro, etc.), accederán a dichas instancias de evaluación.

Las evaluaciones serán escritas, para dejar constancia del progreso de los estudiantes, con contenidos teóricos y prácticos, en las que se deba aplicar las metodologías estudiadas durante cada curso y poner de manifiesto la internalización de los conceptos de los distintos temas, adjuntando documentación sobre los trabajos prácticos pertinentes.

Dichos trabajos prácticos podrán ser desarrollados en grupos, pero el examen será presencial, individual y personal, con una primera parte escrita y una segunda parte oral, donde se evaluará el trabajo particular del alumno, incluyendo no solo el resultado, sino también el desarrollo y metodología empleada. En el examen oral, el alumno deberá defender un Trabajo Práctico Final que integrará todos los temas vistos en cada unidad / módulo / trabajo práctico.

Práctica Profesionalizante

En base al documento “[Programa de Educación Técnica: Las prácticas profesionalizantes](#)” del Ministerio de Educación / INET ³¹, entendiendo que si bien esta propuesta no es una carrera de grado o pregrado, es útil seguir los lineamientos allí expuestos para una correcta inserción laboral y vinculación educativa con el sistema socio-económico regional, por lo cual los alumnos deberán cumplir no menos de 200 hs reloj desarrollando un Proyecto Integrador Final de manera abierta y participativa, interactuando con la comunidad (retroalimentándose con comentarios y experiencias útiles en la futura vida profesional del estudiante).

El eje de las prácticas profesionalizantes son los cursos de “Seminario de Introducción al Software Libre” e “Ingeniería del Software Libre” que aportarán los contenidos y más de la mitad de las horas necesarias para llevar a cabo el proyecto integrador final. El resto de los cursos aportará una cantidad similar de horas aplicadas a dicho proyecto, a partir de los Trabajos Prácticos de cada asignatura, principalmente en “Desarrollo de Aplicaciones Web”, donde se orientará al alumno a producir un aporte original, real y útil, en colaboración de ONG, empresas, cooperativas u organismos gubernamentales.

En todos los casos, el alumno deberá acreditar las horas en prácticas externas para desarrollar el trabajo, convalidadas por un director o tutor de la respectiva entidad.

La Práctica Profesionalizante será evaluada mediante la defensa, por parte del alumno, de una tesina ante un tribunal formado especialmente a tal fin, por docentes de los distintos cursos de la diplomatura.

Herramientas abiertas, libres y gratuitas

Principalmente se utilizará GNU/Linux como sistema operativo, Python como lenguaje de programación y PostgreSQL como base de datos.

También se utilizarán (a modo comparativo/introductorio) los lenguajes de programación C, C++, PHP, bases de datos MySQL y posiblemente otras herramientas libres cuando los contenidos lo requieran.

Sistema Operativo GNU/Linux

El estudio de sistemas operativos Unix en ámbitos académicos es práctica frecuente de larga data, debido a que fueron pioneros en la mayor parte de los adelantos técnicos (manejo de archivos, procesos, memoria, comunicaciones, etc.), como también por su diseño simple y claro.

En su momento, debido a restricciones de licenciamiento que surgieron con el software propietario y otras complejidades, en 1987 el profesor Andrew Tanenbaum desarrolló el sistema operativo MINIX de código abierto con un enfoque educacional para la enseñanza de diseño de sistemas operativos. En 1991 el estudiante universitario Linus Torvalds comenzó a trabajar en un reemplazo superador de MINIX que se convirtió en el núcleo LINUX, que complementó las herramientas del proyecto GNU iniciado en 1983 por Richard Stallman para crear un sistema operativo completamente libre, finalizando en lo que hoy se conoce como GNU/LINUX.

Este ambiente permite estudiar en profundidad los conceptos de sistemas operativos, lenguaje de control, herramientas y utilitarios, pudiendo incluso analizar

³¹ <http://www.entrieros.gov.ar/CGE/2010/tecnica/files/2011/03/Doc-de-Pr%C3%A1cticas-Profesionalizantes-3-2-2010.pdf>

su estructura, comportamiento y código de programación, modificar y adaptar para necesidades particulares, gracias a ser completamente software libre de código abierto, facilitando la administración y optimización para distintos usos. A su vez es la base fundamental para desarrollar sistemas basados en software libre.

Base de datos PostgreSQL

El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL es un derivado del paquete Postgres iniciado en 1986 en la Universidad de Berkeley, California, EEUU. Con más de dos décadas de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, perl, tcl y python).

PostgreSQL fue pionera en muchos de los conceptos de bases de datos relacionales orientadas a objetos que luego empezaron a estar disponibles en algunas bases de datos comerciales, ofrece soporte al lenguaje SQL:2003, integridad de transacciones, y extensibilidad de tipos de datos.

El proyecto Postgres, liderado por el Profesor Michael Stonebraker, fue patrocinado por diversos organismos oficiales u oficiosos de los EEUU: la Agencia de Proyectos de Investigación Avanzada de la Defensa de los EEUU (DARPA), la Oficina de Investigación de la Armada (ARO), la Fundación Nacional para la Ciencia (NSF), y ESL, Inc.

Entendemos que es el candidato ideal para la enseñanza del curso de Bases de Datos por su fuerte seguimiento a los estándares (lo que lo hace compatible con alternativas propietarias), cobertura completa de características avanzadas requeridas para el estudio de sistemas de bases de datos (fiabilidad, transacciones, consistencia, optimización, funciones, índices, disparadores, etc.), estructura concisa útil didácticamente y excelente diseño para seguridad, confiabilidad, integridad, rendimiento, extensibilidad (lo que también la convierte en un componente importante para el desarrollo de sistemas de software libre avanzados).

PostgreSQL es utilizado por gobiernos y empresas de primer nivel, y frecuentemente es comparado como la alternativa libre a la base de datos propietaria más reconocida. A mayo de 2008, unas de las bases de datos más grandes del mundo (administrada por Yahoo!) era soportada por una versión adaptada de PostgreSQL.

Lenguaje de Programación Python

Python es un lenguaje de programación multipropósito publicado inicialmente en 1991 por Guido van Rossum en el CWI (Centro para las Matemáticas y Ciencias de Computación en los Países Bajos).

Durante su estancia en CNRI (Corporación para Iniciativas de Investigación Nacional en EEUU), van Rossum lanzó el proyecto CP4E (Programación de computadoras para todos) auspiciado por DARPA (Agencia de Proyectos de Investigación Avanzada de la Defensa de EEUU), con el fin de hacer la programación más accesible. Python tuvo un papel crucial en este proceso: era idóneo debido a su orientación hacia una sintaxis limpia.

Su diseño multiparadigma, su extensa biblioteca estándar (con gran cantidad de módulos y funciones para las más diversas tareas) y su extensibilidad con módulos personalizados permite cubrir los requerimientos de programación prácticamente de la totalidad de los cursos, con la ventaja de que su código abierto permite el estudio y modificación de dichas funciones.

Python también es reconocido en el ámbito de aplicaciones científicas, desarrollo multimedia y juegos, aplicaciones web, entre otros, y principalmente por su estructura didáctica simple y clara es utilizado para la enseñanza secundaria, terciaria y universitaria alrededor del mundo. Su estructura didáctica e intérprete interactivo permite a los alumnos enfocarse rápidamente en los problemas y algoritmos a desarrollar, dejando de lado cuestiones triviales de sintaxis, compilación, etc. Esto es apoyado por la existencia de herramientas especialmente diseñadas con Python para el ámbito educativo (pythoncard/gui2py y web2py), que serán utilizados en el presente proyecto.

A su vez, Python es uno de los 10 lenguajes más utilizados a nivel mundial y forma parte de una fórmula ventajosa para la productividad, calidad y mantenibilidad del software en muchas compañías e instituciones alrededor del mundo, incluyendo Google y YouTube.com como algunos de los referentes más importantes.

Bibliografía y Materiales abiertos, libres y gratuitos

La mayoría de los materiales de consulta utilizados como base son libre y abiertos, pudiendo citar principalmente:

- [Máster Universitario en Software Libre](#)³² (Universitat Oberta de Catalunya): Si bien es una carrera de postgrado, los contenidos seleccionados son adecuados al nivel terciario y muy bien organizados. La modalidad "virtual" a distancia de la UOC ayuda a una estructura de los documentos simple y gradual, facilitando la lectura y práctica auto-dirigida por parte del alumno, para complementar y profundizar los temas presentados en clase. Además de los contenidos básicos, incluyen lecturas adicionales muy relevantes. Los materiales docentes y programas de estudio pueden consultarse en el sitio [UOC OpenCourseWare](#)³³. Licencia del sitio: CC BY-NC-SA. Licencia de los documentos: GNU FDL.
- [Proyecto Alqua.org \(editorial virtual libre\)](#)³⁴: "Alqua es una comunidad dedicada a la recopilación de libros y material libre. En concreto, han publicado varios libros y manuales de física, matemáticas e informática, que se pueden descargar, modificar y copiar libremente". Los libros son más que nada apuntes de algunas materias, pero lamentablemente algunos han sido retirados por no estar terminados o corregidos. Licencia de los documentos: CC BY-NC-SA
- [Luis Miguel Armendáriz](#)³⁵ (Güimi): "Elaboración propia utilizando principalmente apuntes de trabajo, de distintas asignaturas universitarias, trabajos del profesor Montañana publicados en RedIRIS y artículos de la wikipedia (<http://www.wikipedia.org>)". Recopilaciones muy interesantes y completas para utilizar como material de consulta. Licencia del sitio y los documentos: CC BY-SA. Ver página de documentación para el listado completo.

³² <http://estudios.uoc.edu/es/masters-universitarios/software-libre/>

³³ <http://ocw.uoc.edu/informatica-tecnologia-y-multimedia/>

³⁴ <http://alqua.org/>

³⁵ http://www.guimi.net/index.php?pag_id=cmsxp90_personal.html

- [Latinux Press](http://www.latinuxpress.com/)³⁶: "Consortio de Cooperación Internacional que impulsa y desarrolla soluciones basadas en Software Libre y Tecnologías Abiertas". Traducción del libro de web2py al español.
- [Materia Algoritmos y Programación 1, FIUBA cátedra Wachenchauer](https://sites.google.com/site/fiuba7540rw/)³⁷ ("Python"), de la Facultad de Ingeniería, Universidad de Buenos Aires.
- [Grupo de Usuarios de PostgreSQL Argentina](http://www.postgresql.org.ar/)³⁸: El sitio incluye una recopilación de artículos y traducciones del manual oficial, que fui realizando en estos años para la materia Bases de Datos. Ver sección Documentación.
- [Grupo de Usuarios de Python Argentina](http://www.python.org.ar/)³⁹: especialmente la sección "Aprendiendo Python" que incluye una extensa bibliografía

Desafíos y oportunidades

Traducción de la Documentación oficial al Español y Adaptación de contenidos

Si bien el inglés es la "lingua franca" en estos temas de tecnología, sería importante tener los manuales oficiales traducidos (Python y PostgreSQL principalmente, las páginas manual de GNU/Linux ya están traducidas -la mayoría-), sobre todo porque materias como "Inglés Técnico" están fuera del alcance y posibilidades de esta propuesta, y por consiguiente puede llegar a ser una complicación, más que nada si los alumnos usan servicios de traducción automática que en la mayoría de las veces es muy imprecisa.

Al ser software libre, generalmente la documentación está licenciada bajo los mismos términos, lo que permite traducirla y modificarla al contexto de esta propuesta (algo que sería más complicado o imposible con software propietario).

En el mismo sentido, sería interesante una adaptación de algunos materiales mencionados para el lenguaje Python, unificando los contenidos ya que actualmente están basados en Java, C o Pascal (especialmente para nuestro país). También sería útil completar algunos temas (como normalización de bases de datos), de los cuales todavía no se encuentra suficiente documentación libre y abierta en castellano.

Investigación y Adaptación de herramientas

Python y PostgreSQL tienen herramientas y entornos de desarrollo simples como [IDLE](#) y [PgAdmin3](#). Si bien son suficientes para presentar los contenidos, posiblemente sería conveniente mejorarlos aún más para que sean más didácticos, flexibles y útiles para otros tipos de público.

En este sentido, ver el proyecto [rad2py](https://code.google.com/p/rad2py/)⁴⁰ en el que se busca integrar en una herramienta ligera y de fácil uso, los temas de programación (y depuración y diseño de interfaces), bases de datos e ingeniería de software (repositorios, calidad, etc.), aplicable tanto en educación como al desarrollo profesional. También existen otras herramientas como [ninja-ide](http://ninja-ide.org/)⁴¹ y [pgmodeller](http://www.pgmodeler.com.br/)⁴² (cuyo desarrollo ha sido iniciado en Argentina y Brasil respectivamente), que si bien tienen otros objetivos y por ende una estructura interna más compleja, pueden llegar a ser útiles para esta propuesta.

³⁶ <http://www.latinuxpress.com/>

³⁷ <https://sites.google.com/site/fiuba7540rw/>

³⁸ <http://www.postgresql.org.ar/>

³⁹ <http://www.python.org.ar/>

⁴⁰ <https://code.google.com/p/rad2py/>

⁴¹ <http://ninja-ide.org/>

⁴² <http://www.pgmodeler.com.br/>

En paralelo, también se está trabajando en proyectos como [gui2py](#)⁴³ (derivación renovada de [PythonCard](#) para desarrollo rápido y simple de interfaces visuales), y en seguir mejorando [web2py](#)⁴⁴, ambos diseñados no sólo para ser utilizados con fines educativos, sino también para brindar herramientas completas más amigables para aquellos con experiencia previa en otros lenguajes (PHP, Visual Basic, Delphi, etc.) y quieren dar sus primeros pasos con Python, permitiéndoles luego desarrollar aplicaciones de mayor envergadura (y quizás, migrar programas legados propietarios).

Cátedra abierta introductoria: “Introducción al Software Libre”

A su vez, se pretende comenzar a explorar el interés por parte de la UDE y la demanda en general, en esta unidad académica sobre la temática, con una cátedra libre y totalmente virtual, que se base en una [Introducción al Software Libre](#).

Equipo Docente:

Dado el carácter integral de la formación abordada, y en consonancia con los requisitos exigidos a los cuerpos académicos según las reglamentaciones vigentes, los profesores: Provenirán de un conjunto mixto universidad - empresa, con una adecuada formación teórica, práctica, experiencia profesional y participación en comunidades de software libre; Salvo casos excepcionales, deberán tener una formación de nivel universitario o terciario, como mínimo equivalente a la certificación que se imparte en cada curso.

Inicialmente se podrá comenzar con un plantel mínimo de al menos dos docentes, con la posibilidad de hacer una convocatoria pública una vez confirmada la presente propuesta.

Infraestructura y Equipamiento

Dada la característica semi-presencial, los cursos se desarrollarán principalmente mediante un campus virtual ([moodle](#)). El seguimiento se realizará principalmente con foros pro-activos coordinados por el profesor de cada curso, con una evaluación continua mediante una serie de trabajos prácticos. Se estima conveniente un encuentro presencial por mes para cada materia (posiblemente un día sábado contemplando los alumnos que deban viajar desde el interior), para presentar los temas teóricos, brindar tutorías, clases de consulta y exámenes de acreditación. Para la formación práctica, se necesitaría disponer de laboratorio para llevar a cabo los talleres. Excepto las evaluaciones de acreditación, el resto de los encuentros sería opcional, ya que buscan brindar clases de apoyo para los alumnos que lo necesiten.

También se puede explorar las posibilidades de [cursos abiertos masivos en línea](#) (MOOC) como por ej. “[Python para Zumbis](#)” (uno de los primeros cursos de ese estilo en Brasil, con decenas de miles de alumnos inscriptos, +200 mil reproducciones en YouTube, y un alto grado de aprobación, sin presupuesto ni personal dedicado). Otro ejemplo, en inglés, es el curso “[Programación para todos](#)” (Universidad de Michigan), que incluso tiene todos los materiales bajo una licencia abierta que podría tomarse como base, traducir y adaptar. El [MIT](#) tiene también un curso introductorio similar. En este sentido puede analizarse la iniciativa [Program.ar](#) del estado nacional, que busca incentivar el aprendizaje de ciencias de la computación (en especial el motor [Pilas Engine](#) para introducción a los videojuegos desarrollado en Python).

⁴³ <http://gui2py.googlecode.com/>

⁴⁴ <http://www.web2py.com> - <http://www.web2py.com.ar/> y <http://web2py.googlecode.com>

Certificación:

A quienes cumplan con la aprobación de los ocho cursos y del trabajo final de integración, la institución les otorgará el correspondiente certificado de Diplomado en Software Libre. Aquellas personas que solamente cumplan con la aprobación de uno o más cursos pero no con la totalidad de los requisitos, se les certificará los cursos aprobados. *“La Diplomatura en Software Libre constituye una capacitación que no otorga Título, sino certificación, no homologable con las carreras de grado.”*

Puntaje docente y Capacitación de empleados públicos

Sería conveniente analizar la posibilidad de que los cursos otorguen puntaje para docentes. Una búsqueda sobre cursos relacionados en el [nomenclador provincial de cursos bonificantes](#) (Prov. Bs.As.) arroja los siguientes resultados:

Código	Título (D: a distancia, P: presencial)	Institución	Hs	Ptos
19702	Op. PC c/ GNU/Linus y Software Libre (P)	UNQ	60	0.14
18721	Form. Doc. s/ Hardware y Software Libres	Dir.Cap.	60	0.44
18408	GNU/Linux en la Form. Técnico Profesional	Inst.Municip.	180	0.36
20384	Administrador Linux. Curso Avanzado (D)	Dirección Provincial de Educación Técnico Profesional	106	0.30
20382	Linux Básico Para Usuarios Finales (D)		24	0.22
20003	Lab. de Prog. Desarrollo Ap. Móviles (P)		48	0.40
20321	Act. en Programación en Leng. Java (P)		40	0.48
19782	NGN - Redes de Datos (D)	INET	80	0.14
17784	Especialización en Bases De Datos	ENAIIP	48	0.28
11884	Programacion Estructurada y Leng. Pascal	E.E.M.Nº3		0.10
16801	Creación de Sitios Web	Ort Argentina	60	0.26

Los diversos cursos también podrían aplicarse en capacitaciones de sectores gubernamentales, pudiendo citar cursos similares: [“Administración GNU/LINUX I \(IN-ME-10944\)”](#)⁴⁵ y [“Instalación y administración de Apache \(IN-ME-11004\)”](#)⁴⁶, brindados por el Instituto Nacional de la Administración Pública (INAP⁴⁷).

Programa de Crédito Fiscal SEPyme y Becas

El Ministerio de Industria tiene un programa de [crédito fiscal para capacitación](#) de recursos humanos⁴⁸, para el cual han sido aptos en otras diplomaturas⁴⁹ (posiblemente suspendido, ver el instructivo⁵⁰). Existen varios programas de becas para alumnos y docentes (FONCyT, Bicentenario, PROG.R.ES.AR) que podrían llegar a aplicar a esta propuesta, ver sitio [“Estudiar Computación”](#)⁵¹ para mayor información.

⁴⁵ <https://capacitacion.inap.gov.ar/?cursos=administracion-gnulinux-i-in-me-10944>

⁴⁶ <https://capacitacion.inap.gov.ar/?cursos=instalacion-y-administracion-de-apache-in-me-11004>

⁴⁷ http://www.jgm.gov.ar/sgp/paginas_dhtml?pagina=30

⁴⁸ <http://www.industria.gov.ar/credito-fiscal-capacitacion>

⁴⁹ <http://www.sceu.frba.utn.edu.ar/e-learning/component/content/article/331.html>

⁵⁰ <http://www.industria.gov.ar/wp-content/uploads/2012/08/Instructivo-Carga-de-Unidades-Capacitadoras-UCAPs.pdf>

⁵¹ <http://www.estudiarcomputacion.gov.ar/becas/>