

DIPLOMATURA EN SOFTWARE LIBRE

Curso: Ingeniería del Software libre

Planificación inicial y Temario detallado de estudio – Versión Preliminar tentativa

Institución: Universidad del Este - La Plata - Buenos Aires – Argentina

Período: 2º CUATRIMESTRE de 2015

Docente a cargo: Mg. Lic. Mariano Reingart, Ing. Sebastián Dominguez

Horas semanales: 4

PRESENTACIÓN GENERAL

La creciente complejidad de los sistemas tecnológicos enfrenta al trabajador técnico profesional con situaciones cotidianas que requieren la puesta en acción de competencias configuradas como capacidades complejas. Estos conocimientos, habilidades, destrezas y actitudes deben ser construidas en el nivel educativo mediante abordajes pluridisciplinarios que tiendan a estrechar la diferencia entre el saber hacer exigido en el campo profesional y el saber y el hacer, a menudo fragmentados en las prácticas pedagógico-didácticas vigentes en la actualidad.

Asimismo, el conocimiento sobre qué es el Software Libre, cómo se gestiona y especialmente sobre su importancia para ámbitos como el Estado (en tanto estrategia de soberanía tecnológica) y las implicaciones de su existencia a nivel de la sociedad, es un valor adicional para los egresados de la diplomatura. Esto se debe a que en la actualidad muchas de las prácticas humanas que realizamos cotidianamente utilizan software y la posibilidad de su control se presenta como un asunto estratégico y de importancia individual y colectiva.

Por ello, se propone el trabajo por proyecto (integrador y transversal), con el objetivo de facilitar a los alumnos los conocimientos para desarrollar software libre, bajo la metodología, lenguaje de programación, base de datos y demás herramientas vistas durante el año en esta diplomatura. Para ello los alumnos conformarán un equipo de trabajo, y cumplirán los roles necesarios para participar en el desarrollo de un proyecto de software libre relevante. La investigación y la organización de tales actividades a desarrollar, así como la delegación de las tareas estará a cargo en su totalidad por los alumnos, orientados por los docentes.

OBJETIVOS

Facilitar a los alumnos y las alumnas las herramientas necesarias para que puedan aplicar los conocimientos adquiridos a lo largo de la diplomatura en idear, definir, formular, presentar y desarrollar un proyecto de software libre (trabajo final integrador), incluyendo:

- Aplicar técnicas de relevamiento de un proyecto de software libre.
- Especificar requerimientos de un proyecto de software libre.
- Realizar un análisis sobre un proyecto de software libre.
- Realizar el diseño e implementación de un proyecto de software libre.
- Aplicar técnicas de pruebas (testing), aseguramiento de calidad y seguimiento.
- Documentar un proyecto de software libre.
- Presentarlo ante un público determinado utilizando herramientas adecuadas.

El eje de la práctica profesionalizante se centra en la búsqueda de capacidades profesionales para lograr:

- La crítica y el diagnóstico a través de una actitud científica.
- Una actitud positiva ante la innovación y el adelanto tecnológico.
- La participación en equipos de trabajo para la resolución de problemas y la toma de decisiones, a pesar de trabajar normalmente de manera individual.
- La adaptación a nuevos sistemas de organización del trabajo.

Con estas capacidades el alumno podrá participar en el medio productivo de una manera más eficiente. Se pone énfasis en la permanente investigación y búsqueda de herramientas que permitan afianzar los conceptos adquiridos durante la cursada. Los alumnos realizarán un estudio de campo supervisado mediante observaciones y participación en proyectos de software libre existentes y relevantes.

UNIDADES TEMÁTICAS

Eje temático central: Si bien las unidades están segmentadas por temas y saberes, no son conocimientos aislados, ya que por el contrario, se encauzarán desarrollando un proyecto práctico unificado bajo la filosofía y características del “Software Libre”.

Unidad 1. Fundamentos de las Metodologías Ágiles:

Ingeniería de Software. Herramientas. Métodos. Procesos. Calidad. Ciclo de vida tradicional en cascada. Desarrollo evolutivo. WIN-WIN. Concurrencia. Manifiesto Ágil. XP y Scrum. Modelos de producción del Software Libre.

Unidad 2. Herramientas de Gestión Proyectos Libres

Documentación. Formato Wiki. Adjuntos. Línea de tiempo. Edición colaborativa. Usuarios. Temas o Incidentes (issues/tickets). Hitos (milestones). Hoja de ruta (roadmap). Repositorio de código abierto (software libre): SourceForge, GoogleCode, GitHub.

Unidad 3. Gestión de la configuración del software, versiones y cambios

Línea Base. Herramientas: subversion (svn), mercurial (hg), git. Comandos básicos: obtener (init / clone o checkout), agregar (add), borrar (delete), diferencia (diff), comprometer (commit), revertir (revert), actualizar (update). Repositorios distribuidos: pull, push, merge. Integración con proyectos de software libre: Trac, GoogleCode, GitHub o similar.

Unidad 4. El Proceso de Software Personal

Introducción. Fases: Planificación, Diseño, Codificación, Revisión, Compilación, Pruebas, Postmortem. Estándares de codificación, conteo y clasificación de errores. Métodos estadísticos de estimación de recursos. Reportes e índices de rendimiento y calidad. Implicancias para el desarrollo de software libre.

Unidad 5. Aspectos legales y económicos del Software Libre

Derechos de autor. Licencias de Software. Cesiones y Obligaciones. Garantías y Responsabilidades. Patentes. Marcas. Secretos industriales y Confidencialidad. Modelos de negocio. Regulaciones de Seguridad. Consideraciones sobre Datos Personales y Privacidad. Ética Profesional.

Unidad 6. Trabajo Integrador Final (TIF)

Redacción de textos científico-técnicos. Presentación de documentos y elaboración de presentaciones. Exposición de contenidos en vídeo.

METODOLOGÍA

La metodología consistirá en la organización de pequeños grupos de trabajo, dentro de cada grupo cada alumno trabajará desde su computadora individual comunicados vía Internet por el campus virtual.

El profesor asumirá un método de enseñanza directa ya que conducirá y dictará las pautas a seguir, realizando una exposición del tema a desarrollar (con demostraciones de las actividades prácticas o presentación de temas teóricos), para que luego los alumnos puedan reproducir y analizar los nuevos contenidos a aprender.

Se busca un aprendizaje significativo, favoreciendo que los alumnos relacionen con mayor facilidad los conocimientos aprendidos a lo largo de la diplomatura. Este tipo de aprendizaje es el que hará efectiva la comprensión de los temas, donde habrá un proceso de elaboración de conocimiento de una manera integral.

EVALUACIÓN

Las evaluaciones serán formativas con base constructivista (evitando la mera memorización mecánica o repetitiva), de carácter procesual (teniendo en cuenta los procedimientos realizados y no solo el resultado final), con contenidos teóricos y prácticos, llevadas a cabo principalmente en el campus virtual.

El alumno deberá poder contestar preguntas teóricas sobre los contenidos vistos en el curso y desarrollar ejercicios prácticos usando las herramientas utilizadas durante la cursada. También se valorará favorablemente la participación en grupos de usuarios, foros y sitios de proyectos de software libre existentes y relevantes.

Se evaluará de manera continuada, bajo modelos de evaluación tradicional (con varios exámenes parciales domiciliarios por cuatrimestre: actividades -trabajos prácticos- a entregar por el campus virtual) y enfoques alternativos actualizados medida por tecnología (observaciones de la participación y colaboración activa con criterios concretos). Se apunta a una evaluación sumativa de todos los conceptos, que refiera rendimiento del aprendizaje de los alumnos, debiendo poner de manifiesto la internalización de los conceptos abordados. Cada evaluación parcial será escrita por computadora (con elementos multimedia), en la que el alumno desarrollará los contenidos teórico-prácticos vistos en el curso en base a una guía de examen facilitada por el profesor. El alumno deberá elaborar y entregar en tiempo y forma todas las actividades obligatorias que proponga el equipo docente.

La evaluación integradora consistirá en la presentación del proyecto desarrollado, y un coloquio en el cual el alumno expondrá y explicará la funcionalidad de la aplicación fundamentando las características en base al código del programa y como lo ha desarrollado.

Actividades:

Se planifican las siguientes actividades de evaluación:

- **Actividad N°1:** Propuesta de proyecto para el trabajo final (breve presentación). Acta de Constitución Proyecto, Pila de funcionalidades del Producto (Scrum), Mapa de Navegación (maqueta), Tabla de tareas / Pizarra con Pila de funcionalidades del Sprint (Scrum), Presentación audiovisual tentativa. Documentación preliminar. Presentación audiovisual preliminar
- **Actividad N°2:** Relevamiento técnico para el envío de contribuciones (repositorio, licencia principal, estilos de codificación, prácticas de aseguramiento de calidad, etc.) incluyendo entrevistas a desarrolladores principales
- **Actividad N°3:** Análisis de compatibilidad de licencias de software libre / código abierto del proyectos (contemplando dependencias de terceros)
- **Actividad N°4:** Modelo de negocio básico (si aplica), análisis de evolución del código fuente, encuesta a los desarrolladores (motivaciones, prioridades, etc.), o similar.
- **Actividad N° 5:** Resultados preliminares (cambios aportados, incidencias resueltas, hitos cumplidos, documentación agregada, etc.). Artículo breve (tesina) y Vídeo demostrativo; Documentación definitiva; Presentación audiovisual definitiva. Además de hacer uso del campus virtual, se deberá utilizar las herramientas que utilice la comunidad (ej. GitHub o similar) para un correcto seguimiento y constatación de las tareas desarrolladas por el alumno.

Calificación:

La calificación será porcentual (0 a 100), siendo 70 el valor mínimo para aprobar los exámenes, actividades y ejercicios. No se contemplan recuperatorios, por lo que el alumno deberá aprobar al menos 2/3 de las actividades propuestas por el docente.

Para acreditar los conocimientos, el alumno deberá aprobar los exámenes parciales para poder rendir el examen final integrador.

Una vez que el alumno haya cumplido con la aprobación de los exámenes y el trabajo integrador final, se entregará al alumno un certificado emitido por la universidad donde conste que el alumno ha aprobado el curso.

En el caso de haber completado la totalidad de los cursos de la diplomatura y habiendo presentado el trabajo final satisfactoriamente, se otorgará el certificado de «Diplomado en Software Libre» (emitido según la normativa de la universidad).

EXPERIENCIA DOCENTE

Mariano Reingart es Licenciado en Sistemas, Magíster en Software Libre (UOC) y actualmente finalizando el Profesorado en Disciplinas Industriales (UTN-INSPT). Es docente en el Instituto Superior Blaise Pascal desde 2009 en materias de 2º y 3º Año (actualmente «Bases de Datos», «Sistemas Operativos», «Interconectividad» y «Práctica Profesional») de las carreras terciarias «Tecnatura superior en Análisis de Sistemas / Redes Informáticas».

Ha trabajado como Analista-Programador Freelance (en varias empresas del sector y actualmente en un emprendimiento propio).

En el área del software libre, es miembro de varias asociaciones y grupos de usuarios/desarrolladores, con activa participación en varios proyectos.

Sebastián Domínguez es Ingeniero en Sistemas de Información, graduado en 2001 en la Universidad Tecnológica Nacional, Reg.

La Plata, se ha orientado hacia el desarrollo e implementación de sistemas de información web y aplicaciones Open Source.

Como socio fundador del DILP (Distrito Informático del Gran La Plata), busca compartir experiencias para crecer en forma colectiva, poniendo como valores principales la profesionalidad, el respeto, el compromiso por el trabajo y honestidad en las relaciones comerciales y humanas. Socio de SOLAR: Asociación Civil Software Libre Argentina.

REFERENCIAS Y ANTECEDENTES**Materia “Práctica Profesional”**

Tecnatura Superior en Análisis de Sistemas - Resolución N°5817/03

Instituto Superior Tecnológico Blaise Pascal - DIPREGEP N° 6131

<http://reingart.blogspot.com.ar/p/materia-practica-profesional.html>

Materia “Ingeniería de Software de Fuentes Abiertas/Libres” (Electiva)

Universidad Tecnológica Nacional – Facultad Regional Córdoba

Modalidad Académica – Propuesta – artículo publicado en las 43º JAIIO STS 2014

BIBLIOGRAFÍA GENERAL**Ingeniería del software en entornos del software libre.**

Jordi Mas, David Aycart Pérez, Marc Gibert Ginestà, Martín Hernández Matías. Segunda

edición: febrero 2007, Barcelona - España. Fundación per a la Universitat Oberta de Catalunya.

http://cv.uoc.es/cdocent/APK5H8_OMRUMW3K7YJKJ.pdf

Desarrollo de proyectos de software libre.

Juan José Amor Iglesias. Israel Herraiz Taberner. Gregorio Robles Martínez.

Segunda edición: febrero 2007. XP06/M2120/02157

<http://materials.cv.uoc.edu/cdocent/1UQS1YE0QRIB978YCDLL.pdf>

Scrum y XP desde las trincheras.

Henrik Kniberg. Traducción: Ángel Medinilla. ISBN: 978-1-4303-2264-1

<http://www.proyectalis.com/wp-content/uploads/2008/02/scrum-y-xp-desde-lastrincheras.pdf>

Flexibilidad con Scrum.

Principios de diseño e implantación con Scrum. Apuntes, conceptos y principios para diseñar, implantar y gestionar proyectos ágiles y equipos multidisciplinares. Autor: Juan Palacio. Noviembre 2007

http://www.navegapolis.net/files/Flexibilidad_con_Scrum.pdf

El Mítico Hombre Mes. Ensayos sobre ingeniería de Software.

Fred Brooks. Edición 20ª Aniversario. Addison Wesley, 1995. ISBN 0-201-83595-9. No hay bala de plata: lo esencial y accidental en la ingeniería del Software: <http://barrapunto.com/~pacoescriba/journal/29784>

PSP (Proceso de Software Personal): Materiales para estudio autodidacta.

Instituto de Ingeniería del Software. Universidad Carnegie Mellon.

<http://www.sei.cmu.edu/tsp/tools/student/>

Redacción de textos científico-técnicos.

Autores: Nita Sáenz Higuera; Rut Vidal Oltra. Universitat Oberta de Catalunya.

http://cv.uoc.es/cdocent/_D_CBBU62JZTHQ9CQQGJ.pdf

Presentación de documentos y elaboración de presentaciones.

Autor: Roser Beneito Montagut. Universitat Oberta de Catalunya.

<http://cv.uoc.es/cdocent/NR64BN4TZTOGMDE0V2D7.pdf>

Gestión y desarrollo de proyectos.

Autor: Alfons Bataller Díaz. Universitat Oberta de Catalunya.

<http://cv.uoc.es/cdocent/V20T7OHXRRBG6LOUUE48.pdf>

Exposición de contenidos en vídeo.

Autor: Antoni Marín Amatller Universitat Oberta de Catalunya.

http://materials.cv.uoc.edu/continguts/PID_00191589/index.html