Ministerio de Cultura y Educación
Universidad Nacional de San Luis
Facultad de Ciencias Físico Matemáticas y Naturales
Departamento: Informatica
Area: Area IV: Pr. y Met. de Des. del Soft.

(Programa del año 2019)
(Programa en trámite de aprobación)
(Presentado el 13/05/2019 11:48:47)

I - Oferta Académica

Materia	Carrera	Plan	Año	Período
(OPTATIVAS) INGENIERIA REVERSA	LIC.CS.COMP.	006/05	2019	1° cuatrimestre

II - Equipo Docente

Docente	Función	Cargo	Dedicación
BERON, MARIO MARCELO	Prof. Responsable	P.Adj Exc	40 Hs

III - Características del Curso

Credito Horario Semanal				
Teórico/Práctico	**Teóricas**	**Prácticas de Aula**	**Práct. de lab/ camp/ Resid/ PIP, etc.**	**Total**
Hs	2 Hs	2 Hs	3 Hs	7 Hs

Tipificación	Periodo
B - Teoria con prácticas de aula y laboratorio	1º Cuatrimestre

Duración			
Desde	**Hasta**	**Cantidad de Semanas**	**Cantidad de Horas**
13/03/2019	21/06/2019	15	100

IV - Fundamentación

El ingeniero de software es un profesional que desempeñará sus tareas en el ámbito profesional dirigiendo proyectos y participando activamente en tareas relacionadas con el desarrollo de software. En muchas ocasiones lo hará de forma independiente y en otras en instituciones públicas o privadas. En esos contextos, en muchas situaciones se encontrará con el problema de entender un sistema de software que ya está en funcionamiento. Esta tarea puede ser necesaria por diferentes razones, como por ejemplo: construir la documentación, mantener, evolucionar o migrar un sistema de software, etc. Para realizar las actividades antes mencionadas, el ingeniero se debe compenetrar con el funcionamiento del sistema en su totalidad. Él debe inspeccionar las componentes de software que conforman al sistema de estudio de forma tal de construir vistas que faciliten la comprensión. La tarea mencionada anteriormente no puede ser llevada a cabo de forma manual porque los sistemas que utilizan las organizaciones son de gran envergadura y por consiguiente los costos serán muy altos. La Ingeniería Reversa proporciona soluciones a los problemas descritos previamente. La Ingeniería Reversa es una disciplina central de la Ingeniería de Software porque provee métodos, técnicas y herramientas que posibilitan la extracción y análisis de la información de un producto de software con el fin de determinar cómo funciona y como está construido. El uso de la Ingeniería Reversa posibilita que el ingeniero de software pueda, entre otras tantas posibilidades, deducir la complejidad del sistema por medio de la comprensión de sus partes y de su funcionamiento. Para llevar a cabo exitosamente esta tarea, los métodos y técnicas que la Ingeniería Reversa facilita permiten la elaboración de diferentes vistas de software. Estas vistas actúan como puente cognitivo entre los conocimientos que el programador posee y los conceptos utilizados en el producto de software analizado. La Ingeniería Reversa también es muy útil para detectar y exteriorizar información perdida de los productos de software. La particularidad antes mencionada generalmente aparece cuando el software evoluciona rápidamente o debe migrar a nuevas plataformas. En estas situaciones muchos de los cambios requeridos no son registrados

adecuadamente por los ingenieros encargados de llevar a cabo la evolución/migración y por consiguiente el uso de técnicas de Ingeniería Reversa permitirá subsanar este inconveniente. Por otra parte, los procesos de mantenimiento, evolución y migración requieren cambios en el producto de software, los cuales pueden producir efectos no deseados y peor aún desconocidos tanto para el ingeniero como para el usuario de software. En este contexto, también es útil la Ingeniería Reversa porque dicha disciplina ayudará a la detección de dichos efectos. Lo antes mencionado redundará en la prevención de fallas que pueden hacer que el software provoque daños irreparables en una organización o más problemático aún pueda causar la pérdida de vidas. El software generalmente se construye en base a otros componentes de software esta característica incrementa significativamente la productividad. En muchas ocasiones las componentes antes mencionadas se encuentran inmersas en dentro de un producto de software y las mismas no han sido detectadas por el ingeniero. Generalmente, esto ocurre en el desarrollo de sistemas grandes en donde se ven involucrados en el proceso de desarrollo equipos grandes de programación. La Ingeniería Reversa es útil para detectar componentes de software reusables de forma tal de poder mejorar la calidad del software e incrementar la productividad. La Ingeniería Reversa puede ser utilizada para detectar vulnerabilidades en los sistemas informáticos. Lamentablemente, en la actualidad, se han comenzado a producir muchos ataques cibernéticos que atentan con la seguridad de las organizaciones y de los estados naciones. Muchas veces estos ataques se producen porque el software, en sus partes críticas, ha sido débilmente diseñado y protegido. La Ingeniería Reversa a través de la utilización de técnicas de inspección ayuda a detectar las partes vulnerables de los sistemas de software. Como se puede observar, por las numerosas aplicaciones (se pueden mencionar más) descritas en los párrafos precedentes, la Ingeniería Reversa es una disciplina fundamental para el desempeño profesional del Ingeniero en Informática. Por consiguiente, el aprendizaje de los conceptos, métodos y herramientas de Ingeniería Reversa, le proporcionarán al ingeniero un nivel de distinción en su desempeño profesional.

V - Objetivos / Resultados de Aprendizaje

•Conocer los principales campos de aplicación del a Ingeniería Reversa.
•Construir y manejar herramientas de inspección de componentes de software.
•Elaborar vistas de sistemas de software.
•Tener criterios para la elección de una herramienta de Ingeniería Reversa para un problema particular.
•Poder aplicar técnicas de Ingeniería Reversa para analizar vulnerabilidades de software.

VI - Contenidos

Unidad 1: Ingeniería Reversa
Conceptualización. Ingeniería Reversa del Software. El proceso de Ingeniería Reversa. Herramientas para la Ingeniería Reversa.

Unidad 2: Extracción de la Información
Extracción Ad-hoc de la Información. Extracción de Información Estática: formal e informal. Extracción de Información Dinámica: Forma e Informal. Técnicas y Herramientas para la extracción de la información: Comandos del Sistema Operativo, Analizadores de Código, Depuradores, Profilers, Instrumentadores.

Unidad 3: Representaciones de Programas
Representaciones de Programas: Modelos UML, Comunicaciones de Módulos, Árbol Dinámico de Invocaciones de Métodos/Funciones. Grafo de Invocaciones de Métodos/Funciones, etc.

Unidad 4: Visualización de Software
Conceptualización. Visualización de la Estructura del Sistema. Visualización del Comportamiento del Sistema. Visualización del Código del Sistema. Construcción de visualizaciones de software. Librerías gráficas como herramientas para la construcción de visualizaciones de software.

Unidad 5: Aplicaciones de la Ingeniería Inversa
Industria del Software. Seguridad Informática: Ingeniería Reversa de Malwares. Cracking. Servicios Web: Análisis de la Comprensibilidad de Descripciones de Servicios Web.

VII - Plan de Trabajos Prácticos

Práctico 1: Ingeniería Reversa

Los prácticos de esta unidad consistirán en el análisis de los diferentes campos de aplicación de la Ingeniería Reversa. Además se analizarán las ventajas que provee la aplicación de la Ingeniería Reversa para la solución de problemas típicos en cada campo de aplicación.

Práctico 2: Extracción de la Información
Los prácticos de esta unidad consistirán en el uso de herramientas de extracción de información desde los sistemas de software.

Práctico 3: Representaciones de Programas
En los prácticos de esta unidad los alumnos construirán diferentes representaciones de programas pequeños seleccionados por la cátedra.

Práctico 4: Visualización de Software
Los prácticos de esta unidad consistirán en el análisis de las ventajas y desventajas de visualizaciones de software tradicionales. También se requerirá a los alumnos que propongan visualizaciones de programas que permitan resaltar algún aspecto específico del software.

Práctico 5: Aplicaciones de la Ingeniería Reversa
Los prácticos de esta unidad consistirán en el uso de herramientas de Ingeniería Reversa para detectar software malicioso, hacer cracking, analizar el nivel de seguridad de servicios web.

VIII - Regimen de Aprobación

Condiciones de Regularización:
i) Asistir al 60% de las clases teóricas, prácticas y de laboratorio.
ii) Aprobar los prácticos de laboratorio. Se otorgarán dos recuperaciones por cada práctico de laboratorio a todos los alumnos.
iii) Aprobar un parcial teórico-práctico con una nota mayor o igual a 6 (seis). Se otorgarán dos recuperaciones para cada evaluación parcial a todos los alumnos.

Condiciones de Aprobación:
i) Por promoción, el alumno debe: a) Contar con las condiciones de regularización ii) y iii), debiendo aprobar el examen parcial teórico-práctico con una nota mayor o igual a 7 (siete), b) Asistir al 80% de las clases teóricas, prácticas y de laboratorio, y c) Aprobar una evaluación final integradora con una nota mayor o igual 7.

ii) Por examen final.

Alumnos Libres: No se aceptan alumnos con esta condición.

IX - Bibliografía Básica

[1] Eilam, E. Reversing. Secrets of Reverse Engineering. Wiley. ISBN-10: 0764574817, ISBN-13: 978-0764574818. 2005.
[2] Dang, B; Gazet, A; Bachaalny, E. Practical Reverse Engineering. Wiley. ISBN: 978-1-118-78731-1. 2014.
[3] Ball, T; Eick, S. Software Visualization in the Large. IEEE Computer, Vol. 29, No.4, April 1996. pp. 33-43. 1996.
[4] Pressman, R; Maxim, B. Software Engineering: A Practitioner's Approach. McGraw-Hill. 8va Edición. ISBN-10:0078022126. ISBN-13: 978-0078022128. 2014.
[5] Bernardis, E. Extracción de Información Dinámica desde Programas Escritos en Lenguaje Java. Tesis de Licenciatura en Ciencias de la Computación. Facultad de Ciencias Físico Matemáticas y Naturales. Universidad Nacional de San Luis. 2013.
[6] Miranda, E. Evaluación de Funcionalidades de Visualización de Software Provistas por Librerías Gráficas. Tesis de Licenciatura en Ciencias de la Computación. Facultad de Ciencias Físico Matemáticas y Naturales. Universidad Nacional de San Luis. 2013.
[7] Azcurra, J. Análisis de Identificadores para Abstraer Conceptos del Dominio del Problema. Tesis de Licenciatura en Ciencias de la Computación. Facultad de Ciencias Físico Matemáticas y Naturales. Universidad Nacional de San Luis. 2015.
[8] Stasko, J; Brown, M; Domingue, J; Price, B. Software Visualization. MIT Press. ISBN-10: 0262193957.ISBN-13: 978-0262193955. 1998.

X - Bibliografia Complementaria

[1]

XI - Resumen de Objetivos

[1] Conocer los campos de aplicación del a Ingeniería Reversa.
[2] Construir y manejar herramientas de inspección.
[3] Elaborar vistas de sistemas de software.
[4] Criterios para la elección de herramientas de Ingeniería Reversa.
[5] Aplicar técnicas de Ingeniería Reversa.

XII - Resumen del Programa

Unidad 1: Ingeniería Reversa
Unidad 2: Extracción de la Información
Unidad 3: Representaciones de Programas
Unidad 4: Visualización de Software
Unidad 5: Aplicaciones de la Ingeniería Inversa

XIII - Imprevistos

Serán resueltos por la cátedra en la medida que los mismos vayan surgiendo.

XIV - Otros

ELEVACIÓN y APROBACIÓN DE ESTE PROGRAMA	
	Profesor Responsable
Firma:	
Aclaración:	
Fecha:	