



Ministerio de Educación y Deportes
Universidad Tecnológica Nacional
Rectorado



**APRUEBA CURSO DE ACTUALIZACIÓN DE POSGRADO DEL DOCTORADO EN
INGENIERÍA, MENCIÓN SISTEMAS DE INFORMACIÓN**

Buenos Aires, 15 de septiembre de 2016

VISTO la presentación de la Facultad Regional Santa Fe mediante la cual solicita la aprobación y autorización de implementación del Curso de Actualización de Posgrado "Prueba de Software" para el Doctorado en Ingeniería, mención Sistemas de Información y

CONSIDERANDO:

Que el Curso propuestos responde a la necesidad de brindar a docentes, investigadores y graduados de la Universidad conocimientos científicos actualizados dirigidos a doctorandos en Ingeniería.

Que la Facultad Regional Santa Fe cuenta con un plantel de profesores de elevado nivel académico y profesional, además de una prolongada y amplia experiencia en el dictado de cursos y seminarios vinculados a los propuestos.

Que la Comisión de Posgrado de la Universidad ha analizado los antecedentes que acompañan la solicitud y avala la presentación, y la Comisión de Ciencia, Tecnología y Posgrado recomienda su aprobación.

Que el dictado de la medida se efectúa en uso de las atribuciones otorgadas por el Estatuto Universitario.

Por ello,



Ministerio de Educación y Deportes
Universidad Tecnológica Nacional
Rectorado



EL CONSEJO SUPERIOR DE LA UNIVERSIDAD TECNOLÓGICA NACIONAL

ORDENA:

ARTÍCULO 1°.- Aprobar el currículum del Curso de Actualización de Posgrado "Prueba de Software" para el Doctorado en Ingeniería, mención Sistemas de Información, que figura en el Anexo I y es parte integrante de la presente Ordenanza.

ARTÍCULO 2°.- Autorizar el dictado del mencionado Curso en la Facultad Regional Santa Fe, con el Cuerpo Docente que figura en el Anexo II y es parte integrante de la presente Ordenanza, en el marco de lo establecido por la Ordenanza N° 1313.

ARTÍCULO 3°.- Regístrese. Comuníquese y archívese.



ORDENANZA N° 1556

| |
|--------|
| UTN |
| SCTYP |
| f.c.r. |
| l.p. |

ING. HÉCTOR CARLOS BROTTTO
RECTOR

A.U.S. RICARDO F. O. SALLER
Secretario del Consejo Superior



Ministerio de Educación y Deportes
Universidad Tecnológica Nacional
Rectorado



ORDENANZA N° 1556

ANEXO I

CURSO DE ACTUALIZACIÓN DE POSGRADO

PRUEBA DE SOFTWARE

DOCTORADO EN INGENIERÍA, MENCIÓN SISTEMAS DE INFORMACIÓN

1. FUNDAMENTACIÓN

Entre los aspectos que hacen a la calidad de los productos de software, la calidad funcional, o más precisamente el grado con que el software cumple con los requisitos funcionales establecidos para el mismo, es de fundamental importancia. Uno de los mayores desafíos de la ingeniería de software es, precisamente, brindar herramientas metodológicas y de análisis efectivas, que contribuyan a brindar garantías de calidad funcional. Existen numerosas técnicas, manuales, semi-automáticas y automáticas, que apuntan a garantizar la calidad funcional del software mediante variedades de análisis dinámicos y estáticos. Algunas de éstas son sumamente potentes en el nivel de garantía que proveen, tales como los análisis estáticos vinculados a técnicas basadas en "model checking", o demostración automática de teoremas. Sin embargo, y principalmente debido a razones de escalabilidad y complejidad de las técnicas anteriores, el mecanismo más empleado en la práctica para dar garantías de correcto funcionamiento de software es la prueba de software o "testing". La prueba de software, que esencialmente consiste en ejecutar el software bajo análisis en un número de escenarios de ejecución particulares, es una actividad sumamente costosa en tiempo, y que demanda ingenio y dedicación por parte del o los ingenieros a cargo de la misma. A nivel de prueba de unidad, que prueba unidades individuales de código fuente, la





Ministerio de Educación y Deportes
 Universidad Tecnológica Nacional
 Rectorado



complejidad está especialmente asociada a la producción de entradas ("inputs") para los programas a testear cuando se plantea alcanzar buenos niveles de cobertura (necesario para incrementar las chances de descubrir defectos). Esto demanda, con frecuencia, una gran labor por parte de los ingenieros, para encontrar entradas que cumplan con complejas combinaciones de condiciones que hacen que se ejercite algún camino o porción particular del código bajo prueba. Por esta razón, es de sumo interés práctico el poder generar entradas de programas, y en general casos de prueba, de manera automática.

2. JUSTIFICACIÓN

La generación automática de casos de prueba o test es un área de investigación muy activa en la actualidad. Varias técnicas provenientes de dominios específicos de la Ingeniería de Software y los Métodos Formales han sido adaptadas, y están siendo aplicadas, a la generación automática de test. Algunos ejemplos de técnicas originadas o exitosas en otros dominios, que se aplican a la generación de test, son: el uso de "model checking" (por ejemplo, a través de herramientas tales como Java PathFinder o Kiasan); el uso de análisis basado en "satisfactibilidad" booleana (SAT solving, por ejemplo a través de herramientas como Symstra y Alloy); el análisis basado en "satisfactibilidad" módulo teorías (SMT solving, por ejemplo a través de PEX); las técnicas de búsqueda (por ejemplo, a través de herramientas como Korat y Pkora); la aplicación de computación evolutiva (por ejemplo, a través de EvoSuite); y la generación aleatoria (por ejemplo, como se realiza en las herramientas Randoop, QuickCheck y AutoTest).

Debido a la importancia que el "testing" o prueba de software tiene como técnica general para el análisis y la previsión de garantías de calidad de software, conocer los principios fundamentales del mismo es crucial para la formación de profesionales de Informática.



Ministerio de Educación y Deportes
Universidad Tecnológica Nacional
Rectorado



Además, debido al alto costo que las actividades de validación y verificación demandan (en las cuales se aplican estas técnicas de pruebas de software), conocer el amplio espectro de herramientas de automatización de pruebas de software es esencial como parte del cuerpo de conocimiento de los mencionados profesionales.


3. OBJETIVOS

El objetivo general del curso es conseguir que los alumnos comprendan los fundamentos de validación y verificación de software en general, de la prueba de software ("testing") como mecanismo específico para estas tareas, y las técnicas y metodologías más importantes vinculadas a las pruebas de software.

Los objetivos específicos de este curso se resumen en los siguientes:

- Que los alumnos consigan una formación sólida sobre los elementos que hacen a la calidad de software, con énfasis en calidad funcional.
- Que los alumnos comprendan los fundamentos de las pruebas de software, incluyendo técnicas y metodologías para llevar adelante actividades de validación y verificación usando este enfoque.
- Que los alumnos conozcan diversas técnicas de análisis automático de software, y aprecien y puedan discernir entre las capacidades de cada una para la automatización de tareas vinculadas a las pruebas del software.
- Que los alumnos se familiaricen con herramientas de pruebas de software automáticas, y puedan utilizarlas en la práctica.
-

4. CONTENIDOS

 **Unidad 1.** El rol de la validación y la verificación en la calidad del software. Confiabilidad de



Ministerio de Educación y Deportes
Universidad Tecnológica Nacional
Rectorado



software. El problema de construir programas correctos. ¿Qué es un programa *correcto*? La prueba de software como actividad de validación y verificación. Conceptos fundamentales de prueba de software, sus objetivos y principios. La prueba de software como especificación.

Unidad 2. Pruebas unitarias, conceptos de prueba y granularidad. Automatización de las pruebas de software usando "frameworks" xUnit. Estructuración y composición de test de unidad. Construcción de suites de pruebas de software.

Unidad 3. Criterios y técnicas de diseño de casos de prueba. Pruebas de caja negra y caja blanca. Pruebas de mutación. Criterios de cobertura. Herramientas de medición de cobertura. Independencia en unidades de test. El concepto de doble y su necesidad. Tipos de dobles: "dummies", "stubs", "mocks", etc. Herramientas de apoyo. Integración continua. Estrategias de integración y herramientas.

Unidad 4. Pruebas de software y sus limitaciones. Necesidad de las especificaciones de programas. Anotaciones en programas. Enfoques tradicionales a la validación y verificación de programas. Programas anotados. Especificaciones operativas vs especificaciones declarativas.

Unidad 5. Generación automática de casos de prueba o test. Generación de entradas ("inputs") para pruebas vs generación de test. El problema de la generación automática de oráculos. Estrategias para la generación automática. Generación automática basada en aleatoriedad. Aleatoriedad ciega vs aleatoriedad guiada. Herramientas basadas en generación aleatoria. Generación sistemática de test. Generación exhaustiva acotada. Ejecución simbólica. Generación guiada por criterios de cobertura estructural. Herramientas para la generación sistemática de test. Generación de test mediante computación evolutiva; problemas vinculados a estas estrategias. Herramientas para la generación de test mediante computación evolutiva.



*Ministerio de Educación y Deportes
Universidad Tecnológica Nacional
Rectorado*



5. DURACIÓN

El curso tendrá una carga horaria de SESENTA (60) horas.

6. METODOLOGÍA

El régimen de cursado previsto es presencial. El curso se desarrollará a través de clases teórico-expositivas y la resolución de problemas.

7. EVALUACIÓN:

Para la aprobación del curso se requerirá, además del 80% de asistencia, que los alumnos aprueben un examen final individual.



Ministerio de Educación y Deportes
Universidad Tecnológica Nacional
Rectorado



ORDENANZA N° 1556

ANEXO II

**CURSO DE ACTUALIZACIÓN DE POSGRADO
PRUEBA DE SOFTWARE
DEL DOCTORADO EN INGENIERÍA, MENCIÓN SISTEMAS DE INFORMACIÓN
FACULTAD REGIONAL SANTA FE**

Cuerpo Docente:

- AGUIRRE, Nazareno

Doctor en Ciencias de la Computación, King's College London, University of London, Reino Unido



Licenciado en Ciencias de la Computación, Universidad Nacional de Río Cuarto
