

PROGRAMA DE ASIGNATURA

I.- DATOS GENERALES

Nombre de la Carrera o Programa: **Ingeniería Informática**

Nombre de la Asignatura: **Electiva: Aseguramiento de la Calidad del Software**

Departamento y/o cátedra: **Ingeniería del Software**

Régimen: **Semestral**

Número de Unidades Crédito: **4**

Ubicación en el plan de estudios: Variable, a partir de 197 UC aprobadas

Tipo de asignatura: Obligatoria		Electiva	X	N° horas semanales: Teóricas	3	Prácticas / Seminarios	0	Laboratorio	0
---------------------------------	--	----------	----------	---------------------------------	----------	---------------------------	----------	-------------	----------

Prelaciones/Requisitos:
Ingeniería del Software

Asignaturas a las que aporta:

Fecha de aprobación del Programa en el Consejo de Facultad:

II.- JUSTIFICACIÓN

Parnas (2002) señalaba que el software es bien conocido por su baja confiabilidad. En parte, esto se debe a la dificultad de lidiar con la complejidad de los sistemas de software; pero el conocimiento, las habilidades y el profesionalismo inadecuados de muchos de los profesionales también contribuyen con este problema. Esta situación no ha mejorado significativamente, al contrario, se agrava con los sistemas actuales. A su vez Spillner (2019) indica que con el avance vertiginoso que han tenido áreas tales como la Inteligencia Artificial, está llevando a que cada vez más, decisores de áreas críticas, tales como la medicina, se apoyen en este tipo de desarrollos. Pero en otras áreas, como la conducción autónoma, algunas decisiones deben tomarse con tanta rapidez que solo el propio sistema puede tomarlas. Estos ejemplos ya muestran cuán crucial es la calidad de los sistemas y, por lo tanto, la educación de las personas responsables del desarrollo y prueba de estos sistemas.

El primer paso entonces, como señala Georick (2020), es cuando las personas tienen un sentido de lo que realmente significa el término calidad y saben lo que tienen que hacer para lograrlo. Por otro lado, cada vez más las actividades de aseguramiento de la calidad (SQA, por sus siglas en inglés, Software Quality Assurance) y de pruebas (CQ, por sus siglas en inglés, Control Quality) tienen más presencia en los planes de los desarrollos del software, pues cada vez más los productos de software demandan altos niveles de calidad. A su vez, la situación se complica con la tendencia actual del lanzamiento de nuevas versiones con más frecuencia.

En principio, cualquier profesional del área puede ser un “probador” de software; sin embargo, en la actualidad hay instituciones internacionales que promueven estándares y habilidades con el fin de conformar un cuerpo de conocimiento alrededor de esta área.

Las normas y sus documentos técnicos asociados, como fuentes de conocimiento codificado, pueden considerarse una forma de transferencia de tecnología. La correcta selección y aplicación de los estándares apropiados debería incrementar la productividad de una organización y tener un impacto económico positivo en ella (Laporte, et al, 2018).

Una gran mayoría de organizaciones que desarrollan software son entidades muy pequeñas (PYME), que tienen hasta 25 personas. Las PYME a menudo tienen dificultades para asociar los estándares ISO / IEC con sus necesidades comerciales y justificar su aplicación en sus prácticas profesionales. (Laporte, et al, 2018)

Venezuela no es una excepción, “(...) el 86,7% de las empresas desarrolladoras de software incluidas en el estudio poseen 20 o menos trabajadores; es decir abarcan la micro, pequeña y mediana empresa (...)” (GIDYC (ULA); CEISOFT (CPETM); LISI (USB), 2011). Por ello, instituciones como ISO e IEEE proponen estándares para asegurar la calidad para toda una variedad de organizaciones, desde las que se encuentran en etapas muy tempranas (*startups*) hasta las más maduras.

Todo lo anterior, plantea entonces, la necesidad de reforzar a los Ingenieros en Informática con una buena base en SQA y CQ. Esto incluye conocer y aplicar los principios de SQA, así como los estándares internacionales, en una situación práctica.

III.- CONTRIBUCIÓN DE LA ASIGNATURA AL DESARROLLO DE LAS COMPETENCIAS

Competencia General 1 (CG1): Aprender a aprender con calidad

Unidad de Competencia 3 (CG1 – U3):

Identifica, plantea y resuelve problemas

Criterios de desempeño de la U3:

1. Reconoce diferencias entre una situación actual y la deseada.
2. Analiza el problema y obtiene información requerida para solucionarlo.
3. Formula opciones de solución que responden a su conocimiento, reflexión y experiencia previa.

Competencia General 2 (CG2): Aprender a trabajar con el otro

Unidad de Competencia 1 (CG2 – U1):

Participa y trabaja en equipo.

Criterios de desempeño de la U1:

1. Identifica roles y funciones de todos los miembros del equipo.
2. Realiza las tareas establecidas por el equipo.

Competencia Profesional Básica 1 (CPB1): Modela para la toma de decisiones

Unidad de Competencia 2 (CPB1 – U2):

Criterios de desempeño de la U2:

III.- CONTRIBUCIÓN DE LA ASIGNATURA AL DESARROLLO DE LAS COMPETENCIAS	
Simula computacionalmente situaciones de la vida real	<ol style="list-style-type: none"> 1. Identifica el modelo que represente la situación real para Recolecta datos de la vida real. 2. Utiliza herramientas de software para la simulación de los datos recolectados.
Competencia Profesional Específica 1 (CPE1): Gestiona proyectos informáticos	
Unidad de Competencia 2 (CPE1 – U2): Planifica y desarrolla un proceso de evaluación integral de un proceso de desarrollo informático	Criterios de desempeño de la U2: <ol style="list-style-type: none"> 1. Planifica el proceso de evaluación del proceso y del producto del software 2. Ejecuta actividades de evaluación del proceso de desarrollo del producto informático y del producto mismo 3. Emite informes de diagnóstico del proceso y del producto con base a mediciones de las variables definidas. 4. Recomienda acciones para una mejora efectiva del proceso de desarrollo del producto informático

IV.- UNIDADES TEMÁTICAS	
UNIDADES	TEMAS
1. Fundamentos de SQA	<ol style="list-style-type: none"> 1.1. Calidad del Software 1.2. Características de Calidad del Producto de software y del Proceso del Software 1.3. Aseguramiento de la Calidad del Software (SQA) 1.4. Errores, defectos, fallas
2. Cultura de Calidad	<ol style="list-style-type: none"> 2.1. Mejoramiento continuo del proceso 2.2. Especificaciones libres de defectos 2.3. Construir con calidad 2.4. Costo de la calidad 2.5. Prevención 2.6. Métricas 2.7. Control Estadístico 2.8. Presión sobre la Calidad
3. Estándares de Calidad el proceso y de SQA	<ol style="list-style-type: none"> 3.1. CMM 3.2. ISO/IEC 29110 3.3. ISO/IEC 12207 3.4. ISO/IEC 15289 3.5. IEEE 730
4. Verificación y Validación	<ol style="list-style-type: none"> 4.1. Definición 4.2. Como la ven los estándares (CMM, ISO/IEC&IEEE 12207) 4.3. Evaluación de la madurez y capacidad del proceso de desarrollo de software
5. Técnicas de Aseguramiento de la Calidad	<ol style="list-style-type: none"> 5.1. Inspecciones 5.2. Auditorias 5.3. Listas de chequeo
6. Control de Calidad	<ol style="list-style-type: none"> 6.1. Pruebas conceptos

IV.- UNIDADES TEMÁTICAS	
UNIDADES	TEMAS
	6.2. Estrategia de Pruebas 6.3. Enfoque de Pruebas 6.4. Especificación de las Pruebas 6.5. Gestión de Riesgos en las Pruebas 6.6. Ejecución y reporte
7. Tendencias del Control de Calidad	7.1. Pruebas en DevOps 7.2. Pruebas de Sistemas Autónomos 7.3. Pruebas de Inteligencia Artificial 7.4. Pruebas en un contexto Ágil 7.5. Pruebas con Equipos Distribuidos
8. Aseguramiento de la Calidad del Proceso para proyectos basados en UML	8.1. UML y Calidad 8.2. Técnicas de aseguramientos de la calidad: estéticas, sintácticas y semánticas 8.3. Métricas y Estimaciones para proyectos basados en UML
9. Proceso de Aseguramiento de la Calidad	9.1. Metodologías para la formulación de métricas 9.2. Método GQM 9.3. IEEE 1061 9.4. Enfoque PSM
10. Aseguramiento de la Calidad	10.1. Organizaciones 10.2. Plan de Aseguramiento de la Calidad 10.3. Estándares Organizacionales 10.4. Programas de Medición

V.- ESTRATEGIAS DE ENSEÑANZA Y DE APRENDIZAJE
Tareas, preguntas guías, actividades, debates, exámenes rápidos, ejercicios y problemas en los temas prácticos, uso de casos reales, aseguramiento de la calidad de un proceso de software y su producto, desarrollo de un proyecto a lo largo del semestre (varias entregas).

VI.- ESTRATEGIAS DE EVALUACIÓN
Exámenes cortos, logros durante el desarrollo de la evaluación del sistema de software, actividades en clase, actividades individuales y grupales.

VII.- REFERENCIAS BIBLIOGRÁFICAS
Textos: ✓ Al-Jadaa, A. Software Metrics. Birzeit University Faculty of Engineering and Information Technology Master of Computing-Software Engineering Course 2014 ✓ Crosby, P. Compleitud, Calidad para el siglo XXI. McGraw Hill. México, 1994.

VII.- REFERENCIAS BIBLIOGRÁFICAS

- ✓ Deming, W. E. **The new Economics: for Industry, Government, Education.** Massachussetts Institute of Technology. Primera edición. U.S.A., 1994
- ✓ Fenton, N. & Bieman, J. **Software Metrics. A rigorous & Practical Approach.** Third Edition Chapman & Hall/CRC Innovations in Software Engineering and Software Development Series. 2014.
- ✓ GIDYC (ULA), CEISOFT (CPETM), LISI (USB) **Estudio de la Industria Nacional del Software: Informe Técnico 2011.** Proyecto de Investigación FONACIT 2005000165. 2011
- ✓ Goerik, S **The Future of Software Quality Assurance.** Springer Open. 2020
- ✓ Gremba, J. & Myers, C. **The IDEAL Model: practical Guide for Improvement.** <http://www.sei.cmu.edu/ideal.brige.html>
- ✓ IEEE **SWEBOK**, 2004.
- ✓ Ishikawa, K **What is Total Quality Control? The Japanese Way.** Prentice Hall. U.S.A., 1985.
- ✓ Juran, J. **Quality Control Handbook.** McGraw Hill. 1974.
- ✓ Laporte, C. y April, A. **Software Quality Assurance.** Ed. Wiley. 2018
- ✓ Laporte, C.; April, A. & Bencherif, K. **Teaching Software Quality Assurance in an undergraduate software Engineering Program.** SQP. Vol 9, N 3. 2007
- ✓ Laporte, C.; Muñoz, M.; Mejia, J.; O'Connor, R. **Applying Software Engineering Standards in Very Small Entities. Form Startups to Grownups.** IEEE Software. 2018
- ✓ Naik, K. & Tripathy, P. **Software Testing and Quality Assurance.** Theory and Practice. Wiley. 2008
- ✓ Paul, M.; Weber, C.; García, S.; Crisis, M.; Busch, M.
- ✓ Parnas, D. **Key Practices of the Capability Maturity Model.** <http://www.ibp.com/pit/ispi/cmm.html>
https://en.wikipedia.org/wiki/David_Parnas Citation from David Parnas's preface to the German book Spillner/Linz "Basiswissen Softwaretest".
- ✓ Pinkster, I.; Burgt, B.; Janssen, D. & Venendaal, E. **Successful Test Management. An Integral Approach.** Springer. 2004
- ✓ Schulmeyer, G. **Handbook of Software Quality Assurance. Fourth edition.** 2008.
- ✓ Somerville, I. **Ingeniería de Software.** Pearson. 9na Edición. 2011.
- ✓ Spiller, A. **Foreword de The Future of Software Quality Assurance.** Springer Open. 2020

Material de apoyo:

Preparados por el profesor