



Control de Calidad de Software

I. DATOS DEL CURSO

Periodo: Otoño 2023

Clave: ITIS 261

NRC: 12603

Horario: Lu 7:00-8:00 a.m. y Mi, Vi 7:00-9:00 a.m.

Prerrequisito(s): Ingeniería de Software II

Nombre del Docente: Margarita Carmina García López

Correo electrónico: margarita.garcialop@alumno.buap.mx

Inicio de curso: 07/08/2023

Fin de curso: 12/12/2023

II. DESCRIPCIÓN DEL CURSO:

2.1 Objetivo general de la asignatura:

Desarrollar software, a partir del uso de estrategias y técnicas que sirven para transmitir un cierto nivel de certeza sobre la calidad de un sistema de información

2.2 Objetivos específicos de la asignatura:

- Diseña y desarrolla sistemas innovadores empleando la vanguardia de la tecnología y metodologías de las TI para apoyar el desarrollo productivo en los ámbitos de la administración pública y privada, las redes sociales y de conocimiento.
- Aplica la capacidad crítica, de análisis y síntesis para integrar el pensamiento creativo, crítico y procesos cognitivos en el desarrollo de proyectos de software innovadores en TI; asumiendo una actitud proactiva, congruente con sus conocimientos, habilidades y valores al interior de grupos multidisciplinarios de trabajo.

III.- REQUISITOS DE ACREDITACIÓN Y POLÍTICAS DE CLASE

Con la finalidad de generar un ambiente de aprendizaje óptimo, se establecen las siguientes políticas específicas para la clase:

- El docente se compromete a:
 - Revisar y programar los trabajos y tareas en tiempo y forma.
 - Establecer las condiciones de un sano aprendizaje en el salón virtual o presencial.
 - Respetar las horas clase
 - Mantener un lenguaje digno en el aula y laboratorio.
 - Preparar sus clases con anticipación, dedicación y esmero.
 - Brindar la retroalimentación pertinente de manera que se promueva el aprendizaje del estudiante.
 - Brindar un trato respetuoso a los estudiantes en el grupo.
 - Realizar una evaluación justa y honesta.
 - Notificar al grupo previamente en caso de retardo o ausencia.
- El estudiante se compromete a:
 - Mantener una sana compostura corporal y de lenguaje durante la sesión de clase.
 - Mantener un promedio mínimo de 80% de asistencia a las sesiones.
 - Entregar las tareas con redacción propia, en caso de incurrir a alguna falta se le dará al alumno una advertencia de palabra, en caso de recurrencia los dos alumnos descubiertos reprobarán el parcial automáticamente con nota de cero.
 - Entregar los trabajos en tiempo y forma.
 - Participar activamente en cada una de las sesiones de clase.
 - Justificar sus inasistencias a más tardar dos clases después.
 - Brindar un trato respetuoso a sus compañeros y docente.

- Estar inscrito como alumno en la Unidad Académica en la BUAP.

NOTA: La calificación mínima para considerar el curso acreditado será de 6.

PRESENTACION CONTENIDO TEMÁTICO

| Unidad de Aprendizaje | Contenido Temático | Bibliografía |
|---|--|--|
| 1. Control de calidad del software | 1.1 Concepto de calidad de software 1.2 Aseguramiento de Calidad 1.3 Metodologías y modelos de madurez en el desarrollo de software de calidad 1.4 Procesos de desarrollo de software y mejora de procesos 1.5 Proceso de revisión y Programa de inspección 1.6 Métricas del producto 1.7 Análisis de componentes de software. 1.8 Desarrollo ágil y su relación con la calidad global 1.9 Refactorización | Walkinshaw, N. (2017). Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer. Sacha, Krzysztof (Editor) (2006). Software engineering techniques: Design for quality, Springer-Verlag, Germany. Braude, Eric J., Bernstein, Michael E. (2016), Software engineering: Modern approaches, Second Edition, Waveland Press, USA. |
| 2. Fundamentos de las pruebas de software | 2.1 ¿Por qué es necesario probar? 2.2 ¿Qué es la prueba (testing)? 2.3 Principios generales de las pruebas 2.4 La psicología de las pruebas 2.5 Los modelos de desarrollo de software 2.6 Los niveles de prueba o fases 2.7 Los tipos u objetivos de prueba 2.8 Pruebas de mantenimiento | Walkinshaw, N. (2017). Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer. Ghazali, Umer W. (2014) Software testing: Essential skills for first time testers, Amazon Digital Services LLC, USA. Watkins, John (2009). Agile testing: How to succeed in a extreme testing environment, |

| | | |
|---------------------------------------|---|--|
| | | Cambridge University Press, USA. |
| 3. Técnicas de diseño de prueba | 3.1 Identificar y diseñar casos de prueba 3.2 Categorías de técnicas de diseño de prueba 3.3 Basada en especificación (caja negra) 3.5 Basada en estructura (caja blanca) Técnicas basadas en la experiencia 3.6 Elección de las técnicas de prueba 3.7 Técnicas estáticas 3.7.1 Proceso de prueba 3.7.2 Uso de herramientas | Walkinshaw, N. (2017). Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer. Myers, Glenford J., Sandler, Corey, Badgett, Tom (2011). The art of software testing, 3rd Edition, Wiley, USA. Watkins, John (2009). Agile testing: How to succeed in a extreme testing environment, Cambridge University Press, USA. |
| 4. Gestión de pruebas | 4.1 Organización de pruebas 4.2 Planificación y estimación de prueba 4.3 El seguimiento y control de los progresos de prueba 4.4 Gestión de la configuración 4.5 Riesgo y pruebas Gestión de incidentes | Walkinshaw, N. (2017). Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer. Lewis, William E. (2008). Software testing and continuous quality improvement, Third Edition, Auerbach Publications, USA. Hunt, Andy, Thomas, Dave (2003) Pragmatic unit testing, The Pragmatic Bookshelf, USA. |
| 5. Herramientas de soporte de pruebas | 5.1 Tipos de herramientas de prueba 5.2 El uso efectivo de herramientas, beneficios y riesgos potenciales 5.3 Uso de una herramienta 5.3.1 Pruebas de Unidad 5.3.2 Integración Continua DevOps | Walkinshaw, N. (2017). Software Quality Assurance: Consistency in the Face of Complexity and Change. Springer. Kaczanowski, Tomek (2013). Practical unit testing with Junit and Mockito, Tomasz Kaczanowski. |










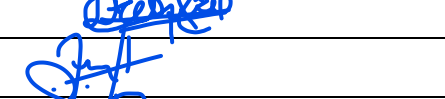





| | | |
|--|--|---|
| | | <p>Pajankar, Ashwin (2017). Python unit testautomation: Practical techniques for Python developers and testers, 1st Edition,Aprees, USA.</p> <p>Gundecha, Unmesh (2012) Selenium testing tools cookbook, Packt Publishing, USA.</p> |
|--|--|---|






VI.- CRITERIOS Y PORCENTAJES DE EVALUACIÓN

- Tareas 20%
- Exámen 30%
- Practicas 20%
- Proyecto Final 30%

El promedio de los 2 parciales que componen el curso y el proyecto final corresponderán al 100% de la nota final.

RELACIÓN DE ENTERADOS

| Matricula | Nombre | Firma |
|-----------|-----------------------------------|---|
| 202062678 | ALVAREZ ENRIQUEZ MARIANA |  |
| 201933809 | ARMAS HERNANDEZ KEVIN |  |
| 202079051 | BANUELOS CAMANO KEVIN JOSAPHAT |  |
| 201934245 | BERNAL MARTINEZ JUAN |  |
| 202032011 | COAGTLE TEMIS ABRAHAM | |
| 202034104 | DIAZ AMARO OSCAR |  |
| 202034202 | DIAZ JIMENEZ YOHARY |  |
| 202081733 | FAUSTINO SANCHEZ DANIEL |  |
| 202035891 | FLORES LECHUGA EVELYN DOLORES |  |
| 202036329 | FRANCISCO PEREZ JESUS ENRIQUE |  |
| 202082331 | GALINDO MENDEZ JONATHAN SEBASTIAN |  |
| 202084500 | HUEPA HERNANDEZ ROSA |  |
| 202045342 | MARTINEZ CARRERA ITZEL DANIELA |  |
| 202049385 | OJEDA JIMENEZ MARTIN ALEJANDRO |  |
| 201926783 | PABLO RUIZ YOSSELIN |  |
| 202089881 | REYES PEREZ BRYAN ARMANDO |  |
| 201909575 | ROSALES ZAYAS VICTOR MANUEL | |

| | | |
|-----------|--------------------------------|---|
| 202075260 | SALINAS CHAVEZ RAYMUNDO | |
| 202057738 | SANTOS VASQUEZ YOALLI ITZAE |  |
| 201866715 | SILVA CRUZ DANIEL ENRIQUE |  |
| 202077419 | VILLA LUNA ISAAC EDUARDO |  |
| 202061734 | XICOTENCATL MENESES LUIS ANGEL |  |
| 202062102 | ZAVALETA SANCHEZ VICENTE |  |