



PROGRAMA DE LA ASIGNATURA

1. INFORMACIÓN GENERAL DE UBICACIÓN DE LA ASIGNATURA

Área: Ciencias Tecnológicas y Agrarias

Docente: Bergman Villarroel Juan Carlos

Facultad: CIENCIA Y TECNOLOGÍA

Carrera: ING. EN CIENCIAS DE LA COMPUTACIÓN

Sistema: Normal Semestralizado

Asignatura: PRUEBA Y MANTENIMIENTO DE SOFTWARE

Gestión: 1/2020

Sigla: COM470

Fecha: 25-06-2020

Curso: 7

2. DESCRIPCIÓN Y JUSTIFICACIÓN DE LA ASIGNATURA

El curso está dirigido a los alumnos del 7º semestre de las carreras de Ingeniería de Sistemas e Ingeniería en Ciencias de la Computación. En esta etapa de la Carrera, el estudiante debe desarrollar y adquirir habilidades para realizar pruebas y mantenimiento de software en la práctica de la ingeniería de software que consiste en procesos y técnicas de aseguramiento de calidad, uso de herramientas y frameworks de testeo, verificación y validación, mantenimiento y estrategias de mantenimiento. El propósito de esta asignatura es capacitar en esta disciplina al estudiante la cual promueve y garantiza el desarrollo de software de alto rendimiento, calidad y productividad del programador.

3. RELACIONES DE LA ASIGNATURA

Tiene como pre-requisito a las asignaturas cursadas en sexto semestre como ser: COM350 - Arquitectura de Software y SIS104 – Desarrollo de aplicaciones móviles.

4. OBJETIVO GENERAL

Desarrollar habilidades para el proceso de pruebas y mantenimiento de software que permita verificar la calidad del producto de software, identificar posibles fallos en la implementación, calidad o usabilidad de un programa que incluye muchas actividades desde la planificación y preparación de las pruebas.

5. OBJETIVOS ESPECÍFICOS

- Introducción, conceptos y definiciones de proceso fundamental de testing.
- Diseño y estrategias de mantenimiento y pruebas
- Uso y aplicación de frameworks para encontrar defectos y fallas en el software y plantear soluciones
- Revisión y verificación en la integración adecuada de componentes de software
- Técnicas para la revisión de requisitos en la implementación
- Comprender el uso y la implementación correcta de los test, con el proceso de desarrollo de software.
- Automatizar pruebas de software para garantizar cambios coherentes en casos de prueba.
- Aplicar conceptos y herramientas de testing y desarrollo guiado por pruebas para asegurar la calidad del software.

6. CONTENIDOS MÍNIMOS

Tema 1: Conceptos Basicos de Testing		
Objetivo particular: Que es testing, importancia y objetivos de las pruebas		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
? ¿Qué es testing? (importancia) ? ¿Conceptos de Testing? (pruebas de software) ? Fundamentos del Testing (Conceptos fundamentales y básicos relacionados al testing) ? ¿Porque son necesarias las pruebas? ? Objetivos de las pruebas ? Cuando finalizamos las pruebas	Conocer los objetivos de realizar los test y la importancia que conlleva esta practica.	Comprender la importancia de realizar los test, cual el objetivo,conceptos fundamentales para realizar pruebas
Tema 2: Proceso Fundamental de Testing		
Objetivo particular: Describir los pasos previos y posteriores a la ejecucion de las pruebas		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
- Planificación y Control - Análisis y Diseño - Implementación y Ejecución - Evaluación de criterio de salida y Reportes - Actividades de Cierre	La ejecución de pruebas es solo una parte del proceso de pruebas, dependiendo del enfoque que seleccionemos el proceso de pruebas se va a realizar en diferentes puntos del proceso de desarrollo.	Conocimiento del proceso de testeo en las diferentes etapas del desarrollo de software



PROGRAMA DE LA ASIGNATURA

Tema 3: Diseño de las Pruebas		
Objetivo particular: Elaboracion de casos de prueba, niveles y tipos de pruebas		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
? Concepto de Caso de Prueba ? Niveles de Prueba - Test unitario - Test de integración - Test de sistema - Test de aceptación ? Tipos de Prueba - Pruebas Funcionales - Pruebas no Funcionales - Prueba estructural - Prueba relacionada a cambios	Conocimiento de las las etapas y tipos de pruebas a realizar con el software	Comprender los distintos niveles del casos de prueba para la deteccion de errores o fallas.
Tema 4: Tipos de Tecnicas de testeo		
Objetivo particular: Conceptos y fundamentos de tecnicas para evaluar el test		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
? Técnicas estáticas Revisiones - Análisis de flujo de control - Análisis de flujo de datos - Análisis del compilador ? Técnicas dinámicas - Caja Blanca o Cobertura de sentencia o Cobertura de rama o Cobertura de condición o Cobertura de camino - Técnicas basadas en experiencia - Caja Negra o Partición de valores de equivalencia o análisis de valores límite o Pruebas de transición de estados o Tablas de decisión o Pruebas de caso de uso	Conocimiento de las tecnicas a emplear par a evaluar los tipos de test	Comprender las diferentes tecnicas para evaluar los tipos de test
Tema 5: Verificación y Validación		
Objetivo particular: Definicion, identificacion y correccion de defectos		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
- Errores, defecto y falla - Identificacion y corrección de defectos - Defincion de V	Actitudes para identificacion de errores, defecto o falla, correccion de defectos.	Conocimiento de actitudes respecto a la verificacion
Tema 6: Introducción a JUnit		
Objetivo particular: Configuracion, instalacion y uso de la libreria JUnit		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
- Instalación del entorno JDK y librerías de JUnit - Configuración IDE (Netbeans	Conocimiento del manejo de la libreria y su configuracion para el entorno	Configurar adecuadamente el entorno para el uso de la libreria
Tema 7: Test con métodos libreria JUnit		
Objetivo particular: Uso de métodos para las asersiones		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
- Primer programa de test - Diferencia entre error y fallo en un test - Anotaciones de los test - Aserciones - Uso de Asserts: (AssertEquals, AssertTrue, AssertFalse, AssertNull, AssertNotNull, AssertSame, AssertNotSame) - Debugear un test	Uso de metodos para test unitarios	Comprender el uso de metodos para pruebas unitarias
Tema 8: Uso de libreria Hamcrest		
Objetivo particular: Manejo de excepciones y matchers		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
? Uso de librería Hamcrest	Manejo de excepciones y asserciones con matchers	Conocimiento del manejo de excepciones y asserciones con matchers
Tema 9: Cobertura y Test Parametrizados		
Objetivo particular: Uso de frameworks para el test de integracion y pruebas unitarias		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores



PROGRAMA DE LA ASIGNATURA

? Cobertura - Cobertura de los test EclEmma ó Jacoco ? Test Parametrizados - Introduccion a test parametrizados - Test parametrizados - Test parametrizados uso de librería junitParams	Uso de librería y frameworks para test parametrizados y test de integración	Conocimiento en el uso de librería para test parametrizados y frameworks de test de integración
---	---	---

Tema 10: Mockito		
Objetivo particular: Uso de la librería en entorno de simulación para test unitarios y de integración		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
Instalación y configuración - Uso de las anotaciones mockito - Verificación de mocks	Uso de la librería en un marco de simulación que tiene como objetivo proporcionar la capacidad de escribir con claridad una prueba de unidad legible utilizando su API simple	Conocimiento del uso de la librería en un marco de simulación

Tema 11: Selenium WebDriver		
Objetivo particular: Uso de la herramienta para automatizar pruebas de interfaz de usuario		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
- Instalación - Configuración - Comprensión del método y forma de uso y aplicación	Conocimiento de la herramienta para automatizar pruebas de interfaz de usuario	Aplicar el conocimiento de la herramienta para automatizar pruebas de interfaz de usuario

Tema 12: Jmeter		
Objetivo particular: Uso de la herramienta para prueba de sobrecarga		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
Instalación, configuración y uso de la herramienta para pruebas de sobrecarga de aplicaciones.	Conocimiento de la herramienta para pruebas de sobrecarga	Aplicar los conocimientos de la herramienta para pruebas de sobrecarga

Tema 13: Test de Cobertura		
Objetivo particular: Uso de frameworks para realizar test de cobertura		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
- Instalación, configuración y uso de frameworks para pruebas de cobertura	Conocimiento y uso de frameworks para pruebas de cobertura	Aplicar los conocimientos del uso de frameworks para pruebas de cobertura

Tema 14: Trabajo de Investigación		
Objetivo particular: Frameworks de testing		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
- Instalación - Configuración - Uso y Aplicación	Conocimiento del framework elegido y su aplicación	Aplicación del framework investigado, relación y uso con el lenguaje de programación utilizado

Tema 15: Desarrollo Guiado por Pruebas de Software - TDD		
Objetivo particular: Usar la técnica de :Escribir las pruebas primero (Test First Development) y Refactorización (Refactoring).		
Sistema de conocimientos	Sistema de habilidades	Sistema de valores
- Introducción a Desarrollo Guiado por Pruebas. - Diseño de Pruebas en Desarrollo Guiado por Pruebas. - Aplicar las herramientas de pruebas unitarias para el desarrollo de software guiado por pruebas.	- La metodología de programación TDD es agnóstica a otros lenguajes de programación ? java, Python, java script, etc.. - Los test van a diseñar nuestro software. - Programación diseñada por test.	Comprender la metodología TDD y aplicar todos los conocimientos de pruebas unitarias en el desarrollo de software guiado por pruebas con el lenguaje de programación que desee el alumno.

Tema 16: Modelos de Desarrollo de Software		
Objetivo particular: Describir los distintos modelos utilizados para el proceso de desarrollo de software, y las actividades que incluyen dentro del proceso de las pruebas		



PROGRAMA DE LA ASIGNATURA

Sistema de conocimientos	Sistema de habilidades	Sistema de valores
El modelo V El modelo Scrum Relacion de las pruebas con los modelos de desarrollo de software	Conocimiento de las métodos de desarrollo y su relacion con las actividades de pruebas de software.	las pruebas no existen de manera aislada, siempre están relacionadas con las demás actividades de desarrollo. Dependiendo el modelo que elijamos en nuestro proyecto va a tener un distinto enfoque hacia las pruebas.

7. DISTRIBUCIÓN DEL FONDO DEL TIEMPO

Nro	Tema	Horas Teo. (P/V)	Horas Prácticas (P/V)			Horas Lab.	Horas Extrac.	Horas Eval.	Total Horas
			Taller	Invest.	Exten.				
1	Conceptos Basicos de Testing	1	0	0	0	0	0	0	1
2	Proceso Fundamental de Testing	1	0	0	0	0	0	0	1
3	Diseño de las Pruebas	1	0	0	0	0	0	0	1
4	Modelos de Desarrollo de Software	1	0	0	0	0	0	0	1
5	Tipos de Tecnicas de testeo	1	0	0	0	0	0	0	1
6	Verificación y Validación	1	0	0	0	0	0	0	1
7	Introducción a JUnit	1	0	0	0	0	0	0	1
8	Test con métodos libreria JUnit	0	0	0	0	2	0	0	2
9	Uso de libreria Hamcrest	1	0	0	0	1	0	0	2
10	Cobertura y Test Parametrizados	1	1	0	0	0	0	0	2
11	Mockito	1	0	0	0	3	0	0	4
12	Selenium WebDriver	1	0	0	0	2	0	0	3
13	Trabajo de Investigación	0	0	2	0	0	0	0	2
14	Desarrollo Guiado por Pruebas de Software - TDD	1	0	1	0	1	0	0	3
15	Test de Cobertura	1	0	0	0	2	0	0	3
16	Jmeter	1	0	0	0	1	0	0	2
Total horas semestre									30
Total horas extracurriculares semestre									0

P = Presencial; V = Virtual

8. CRONOGRAMA

8.1 Cronograma de plan temático y actividades

PLAN TEMÁTICO

Nr o	Tema	Semanas																																		
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33		
1	Conceptos Basicos de Testing																																			
2	Proceso Fundamental de Testing																																			
3	Diseño de las Pruebas																																			
4	Modelos de Desarrollo de Software																																			
5	Tipos de Tecnicas de testeo																																			
6	Verificación y Validación																																			
7	Introducción a JUnit																																			
8	Test con métodos libreria JUnit																																			
9	Uso de libreria Hamcrest																																			
10	Cobertura y Test Parametrizados																																			
11	Mockito																																			
12	Selenium WebDriver																																			

[illegible]

No hay actividades

8.2 Cronograma de evaluaciones (parciales, final y segunda instancia)

No se realizó el registro del Cronograma de actividades

La metodología utilizada consiste en abordar la materia con la introducción, definición y conceptos básicos del testing, el proceso que implica y porque es importante realizar pruebas y mantenimiento al software.

Para empezar con el diseño de las pruebas y de acuerdo al nivel de conocimiento de los estudiantes respecto al manejo de algún IDE de desarrollo y lenguaje de programación siendo que pueden usar el de su elección, pero orientado a la web y de preferencia herramientas Open Source, para lo cual se les guía en un laboratorio desde su instalación y configuración.

Se realiza clases teóricas y prácticas de laboratorio en aula, así como también trabajos de investigación que los alumnos exponen con la herramienta elegida por ellos. Para la enseñanza se utiliza proyector data display, pizarra y laboratorio con equipos de computación y la conexión a internet.

Se califica las prácticas de laboratorio realizadas en clase, así como también dos exámenes parciales, uno final y las tareas de investigación.

10. RECURSOS DIDÁCTICOS

Plataforma ecampus

Video Conferencia ZOOM

Plataforma Classroom

Videos de clases

Formularios

Contenido interactivo

11. ACTIVIDADES DE INVESTIGACIÓN Y/O INTERACCIÓN

Elaboracion de practicas de laboratorio de computacion aplicando el uso de frameworks en la resolucion de problemas, envio a la GitHub o la plataforma ecampus

Trabajos de investigacion y exposicion, y envio a la plataforma ecampus

12. SISTEMA DE EVALUACIÓN DE APRENDIZAJE

Sistema de Evaluacion SEA:I de la USFX

25% laboratorio

10% trabajo de investigacion

30% parciales

35% final

13. BIBLIOGRAFÍA

Proyecto de Diseño Curricular de Creación de la Carrera de Ingeniería en Ciencias de la Computación– Carrera de Ingeniería de Sistemas, Ingeniería en Telecomunicaciones e Ingeniería en Diseño y Animación Digital - 2017

WHITTAKER J.: Exploratory Software Testing. Ed. Addison Wesley, 2009.

BLACK R.: Managing the Testing Process, 3rd Edition. Ed. Wiley, 2009.

CRISPIN, L., GREGORY, J.: Agile Testing: A Practical Guide for Testers and Agile Teams. Ed. Addison-Wesley, 2009.



PROGRAMA DE LA ASIGNATURA

Datos Complementarios

Programa elaborado por primera vez: 17/09/2019

Programa modificado por última vez: 22/06/2020

Apartados actualizados:

Bibliografía:

Contenido Mínimo:

Indicaciones metodológicas y de investigación:

Actividades de investigación y/o interacción:

Firma del docente

Firma del(a) Director(a) de Carrera