



## I. INFORMACIÓN GENERAL

<b>CURSO</b>	:	Diseño De Experimentos En Ingeniería De Software
<b>CÓDIGO</b>	:	CC55
<b>CICLO</b>	:	201701
<b>CUERPO ACADÉMICO</b>	:	<b>Cardenas Rengifo, Luis Enrique</b>
<b>CRÉDITOS</b>	:	4
<b>SEMANAS</b>	:	16
<b>HORAS</b>	:	2 H (Práctica) Semanal /3 H (Teoría) Semanal
<b>ÁREA O CARRERA</b>	:	Ciencias de la Computacion

## II. MISIÓN Y VISIÓN DE LA UPC

Misión: Formar líderes íntegros e innovadores con visión global para que transformen el Perú.

Visión: Ser líder en la educación superior por su excelencia académica y su capacidad de innovación.

## III. INTRODUCCIÓN

El curso de especialidad de Diseño de Experimentos en Ingeniería de Software, de la carrera de Ciencias de la Computación, es de carácter teórico-práctico y está dirigido a los estudiantes del séptimo ciclo. El curso busca desarrollar la competencia general de ciudadanía y las competencias específicas de la aplicación de conocimientos de ciencias y de la aplicación de fundamentos matemáticos acorde con los ABET - Student Outcomes (A) y (J) respectivamente.

La importancia del curso se debe a que se enfoca en analizar, diseñar y elegir las herramientas y técnicas apropiadas para ejecutar aquellas pruebas con las cuales sea más probable descubrir no conformidades en el producto de software de entretenimiento. Se emplean herramientas libres y propietarias para la administración de proyectos de pruebas, planes de pruebas, casos pruebas, visualización de resultados y reportes, administración de defectos y automatización de las pruebas, y se configuran los procesos de pruebas necesarias para garantizar y controlar la calidad del software. Esto es necesario en el entorno profesional para ofrecer un software de calidad

## IV. LOGRO (S) DEL CURSO

Al finalizar el curso el estudiante desarrolla pruebas de software de entretenimiento teniendo en cuenta las herramientas CASE.

## V. UNIDADES DE APRENDIZAJE

### UNIDAD N°: 1 Fundamentos de las Pruebas de Software

#### LOGRO

El estudiante comprende la importancia del testeo de software, diferencia entre validación y verificación y el modelo de

fallas de software

**TEMARIO**

Introducción a las Pruebas de Software. Importancia de Pruebas de Software. Modelo de Fallas de Software. Definición de Verificación y Validación. Pruebas de Software y QA / Reportes de Errores de Software. / Niveles de testeo (unitarias, integration: big bang, top-down, bottom up, sistemas). Ataques de entradas de software. Ataques de salidas de software. Ataques de Data y Cálculo. Ataques de sistema de archivos y entorno del software.

**HORA(S) / SEMANA(S)**

10 hrs / Semana 1 y 2

**UNIDAD N°: 2 Técnicas de Pruebas**

**LOGRO**

El estudiante desarrolla pruebas de caja blanca, gris y difusas y su implementación usando herramientas CASE

**TEMARIO**

Particiones equivalentes. Valores Límites. Prueba de pares. Testeo en base a modelos. Testeo de caja blanca (cobertura de código fuente). Testeo de caja gris. Pruebas difusas. Pruebas funcionales. Pruebas en base a requerimientos. Diseño de casos de prueba con casos de uso. Pruebas basadas en escenarios. Rastreo de requerimiento en pruebas. Herramientas para pruebas funcionales, no funcionales.

**HORA(S) / SEMANA(S)**

15 hrs / Semana 3, 4 y 5

**UNIDAD N°: 3 Introducción al Diseño de Experimentos**

**LOGRO**

El estudiante integra tópicos de diseño de experimentos estadísticos a la implementación de pruebas funcionales y no funcionales.

**TEMARIO**

Diseño de experimentos: Definición, La población y la muestra, Estadísticas descriptivas y de inferencia, Etapas que se presentan en la realización de un experimento, Medidas de tendencia central, Medidas de variabilidad. Conceptos básicos de probabilidad y estadística. Pruebas de significancia estadística. Distribuciones de probabilidad conjunta

**TEMARIO DE ESTUDIO AUTO DIRIGIDO 1**

- Diseño de Casos de Pruebas

**HORA(S) / SEMANA(S)**

10 hrs / Semanas 6 y 7

**UNIDAD N°: 4 Proceso de Pruebas**

**LOGRO**

El estudiante administra las pruebas usando un herramienta CASE y prepara los entregables dentro del proceso de pruebas de software de distintos tipos y en diferentes fases de desarrollo

**TEMARIO**

Plan del Proyecto. Etapas de Pruebas. Estrategia de Pruebas. Plan de Pruebas. Casos de Prueba. Entradas de Pruebas. Diseño de Casos de Prueba. Reportes de Errores. Clasificación de los errores. Análisis y rastreo de errores. Reportes de resultados de pruebas. Pruebas de humo. Pruebas de verificación de construcciones. Pruebas de regresión. Herramientas para pruebas de regresión. Pruebas de aceptación del usuario. Pruebas Alpha y Beta

**TEMARIO DE ESTUDIO AUTO DIRIGIDO 2**

- Tipos de Pruebas de Aceptación del usuario
- Tipos de Pruebas Alpha y Beta

**HORA(S) / SEMANA(S)**

35 hrs / Semanas 9, 10, 11, 12, 13, 14 y 15

**VI. METODOLOGÍA**

El curso se dicta en 2 sesiones semanales de 2 horas de teoría y 3 horas de práctica. Los estudiantes organizados en grupos realizan un trabajo práctico en el que se integran los conocimientos y habilidades adquiridos durante el curso.

En las Unidades 3 y 4 desarrollarás, mediante un estudio auto dirigido y guiado remotamente por el profesor, el primer y tercer hito de tu trabajo final. Para cumplir con ambos trabajos se han definido los temas que debes de revisar para cumplir con lo solicitado.

**VII. EVALUACIÓN**

**FÓRMULA**

$$10\% (LB1) + 10\% (LB2) + 15\% (TP1) + 15\% (EA1) + 25\% (TF1) + 20\% (EB1) + 5\% (PA1)$$

TIPO DE NOTA	PESO %
LB - PRACTICA LABORATORIO	10
LB - PRACTICA LABORATORIO	10
TP - TRABAJO PARCIAL	15
EA - EVALUACIÓN PARCIAL	15
TF - TRABAJO FINAL	25
EB - EVALUACIÓN FINAL	20
PA - PARTICIPACIÓN	5

**VIII. CRONOGRAMA**

TIPO DE PRUEBA	DESCRIPCIÓN NOTA	NÚM. DE PRUEBA	FECHA	OBSERVACIÓN	RECUPERABLE
LB	PRACTICA LABORATORIO	1	Semana 6	Individual	NO
LB	PRACTICA LABORATORIO	2	Semana 14	Individual	NO
TP	TRABAJO PARCIAL	1	Semana 7	Grupal	NO
EA	EVALUACIÓN PARCIAL	1	Semana 8	Individual	NO
TF	TRABAJO FINAL	1	Semana 15	Grupal	NO
EB	EVALUACIÓN FINAL	1	Semana 16	Individual	NO
PA	PARTICIPACIÓN	1	Semana 15	Individual	NO

## **IX. BIBLIOGRAFÍA DEL CURSO**

### **BÁSICA**

SOMMERVILLE, Ian (2011) Software engineering. Boston, Massachusetts : Pearson Education.  
(005.1 SOMM 2011)

### **RECOMENDADA**

**(No necesariamente disponible en el Centro de Información)**

PRESSMAN, Roger S. (2010) Ingeniería del software : un enfoque práctico. México, D.F. : McGraw-Hill.  
(005.1 PRES 2010)