

# **PLANIFICACION ANUAL 2020**

**INSTITUCION:** *Escuela Superior de Comercio N° 43*

**CARRERA:** *Técnico Superior en DESARROLLADOR DE SOFTWARE*

**ESPACIO CURRICULAR:** *Ingeniería de software I*

**CAMPO de la FORMACION:** *ESPECIFICA*

**CURSO:** *Primer año*

**CICLO LECTIVO:** *2020*

**PROFESOR:** *Mauro I. Aguilar*

**ASIGNACION HORARIA:** *4 hs. catedra semanales*

**FORMATO:** *Materia*

**REGIMEN de CURSADO:** *Anual*

**PLAN DECRETO:** *Resolución 2120/2016*

## **FUNDAMENTACION**

El mercado demanda en forma constante software más complejo y la idea de programación comienza a tomar un enfoque global cuyo alcance va mas allá de las computadoras personales o servidores, es este marco, hoy en día, podríamos alcanzar la telefonía celular, centrales telefónicas, electrodomésticos (IOT), entre otros.

**El software es considerada una actividad industrial y como tal, debe estar sujeta a normas, procesos y cumplir con estándares.**

En este contexto es que debemos centrar nuestros esfuerzos en alcanzar dichas metas recurriendo a la estructuración de métodos de trabajo y a procesos estandarizados que nos permitan abarcar en términos grupales, que son los equipos de desarrollo las actividades de Análisis, Diseño y Programación de Software.

Para el enfoque de Análisis y Diseño encontramos UML (Unified Modeling Language) que es un lenguaje de modelado de Software.

UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y compuestos reciclados.

## **OBJETIVOS- EXPECTATIVAS DE LOGRO**

- Esta unidad curricular permite introducir al estudiante en el trabajo de Ingeniería, llevando a cabo proyectos con la utilización de métodos y la ayuda de herramientas propias de su ámbito de acción. Reconocer, diferenciar, documentar y validar las necesidades que justifican sus proyectos. Documentar escenarios y propuestas que respondan a dichas necesidades. La tecnología informática es una disciplina sujeta a cambios constantes, nuevos productos, nuevas arquitecturas, diversas plataformas de software y hardware; bajo este contexto, desarrollar software a escala y ser competitivo en el mercado, se ha convertido en los últimos años en un verdadero desafío. En este ámbito resulta necesario cambiar algunos hábitos de aprendizaje, es necesario centrar los esfuerzos en la arquitectura, independientemente del entorno en el que se desarrolle
- Analizar los aspectos vinculados a la Ingeniería del software, sus fundamentos y su importancia en el contexto actual
- Aplicar diferentes normas y métodos que permiten obtener mejores resultados, en cuanto al desarrollo y uso del software, mediante la aplicación correcta de estos procedimientos se puede llegar a cumplir de manera satisfactoria con los objetivos fundamentales de la ingeniería de software.

- Utilizar diagramas UML como parte de los procesos de ingeniería orientados, modelado a partir de casos de estudio.
- Valorar el pensamiento lógico individual y grupal.
- Ampliar los conocimientos, afianzar los contenidos a través de la consulta en sitios web sugeridos por la cátedra.
- Adquirir destrezas en el trabajo grupal y exposición.

## **CONTENIDOS CONCEPTUALES**

### **UNIDAD 1 : SOFTWARE e INGENIERIA DEL SOFTWARE**

- Software. ¿Qué es el software? ¿Quién lo hace?
- Función del Ingeniero de Software.
- Importancia del software
- ¿Cuáles son los pasos?
- El papel evolutivo del software.
- La naturaleza cambiante del software.
- Categorías del software: - de sistemas, de aplicación, científico y de ingeniería, empotrado, de líneas de producto, aplicaciones basadas en web, de ingeniería artificial.
- Software heredado
- Que es la ingeniería del software
- Diferencia entre ingeniería de software e ingeniería de sistemas.
- Que es un proceso de software
- Que es un modelo de procesos de software
- Cuáles son los costos de la ingeniería del software.
- Qué son los métodos de la ingeniería del software. Estructurado, orientado a objetos, UML ( lenguaje unificado modelado)
- Que es el CASE
- Atributos de un buen software.
- Retos que afronta la ingeniería del software
- Responsabilidad social ética en la ingeniería del software

### **UNIDAD 2 : EL PROCESO DEL SOFTWARE . UNA VISION GENERAL**

- Que es un proceso del software. Quien lo hace y porque es importante. Cuáles son los pasos.
- Marco de trabajo para el proceso. Actividades del marco de trabajo. Actividades sombrillas.
- Patrones del proceso
- Evaluación del proceso.
- Modelos del proceso del software. Modelo en cascada, desarrollo evolutivo, ingeniería del software basada en componentes.
- El proceso unificado (PU) . UML .Fases del PU y flujo de trabajo.

### **UNIDAD 3: REQUERIMIENTOS DEL SOFTWARE**

- Requerimientos del usuario.
- Requerimientos del sistema.
- Requerimientos funcionales y no funcionales.
- Requerimientos de dominio.

## **UNIDAD 4 : INTRODUCCION A UML**

- Introducción al UML.
- Qué es UML
- La orientación a Objetos, OO. Qué es un Objeto. Atributos. Métodos. Qué es clase. Qué es la herencia. Qué es una interfaz.
- Bloques básicos de construcción de UML: Los elementos, las relaciones y los diagramas.
- Diagramas de casos de Uso.
- Como utilizar UML.

## **UNIDAD 5 : UML CASOS DE USOS.**

- Qué es un diagrama de casos de uso. Concepto. Ejemplos.
- Elementos de un caso de uso: actores, casos de usos, relaciones entre casos de usos. Include. Extend.ejemplos
- Descripción de un caso de uso.
- Presentación y desarrollo de casos de usos. Trabajos Prácticos.

### **CONTENIDOS PROCEDIMENTALES**

- Elaboración de Diagramas de Casos de Uso.
- Conocimiento de la ingeniería del software
- Utilización de herramientas UML

### **CONTENIDOS ACTITUDINALES**

- Valorar la importancia de las herramientas e información que maneja.
- Incorporar los conocimientos de la ingeniería del software.
- Autonomía y desarrollo del pensamiento y de la búsqueda en internet.
- Respeto por el pensamiento y lógica de sus compañeros.
- Solidaridad en el trabajo grupal.
- Responsabilidad en el cumplimiento de las tareas asignadas y la asistencia a clases durante el cursado de la materia.

### **ESTRATEGIAS METODOLOGICAS**

- Exposición.
- Debate
- Trabajo individual y grupal.
- Trabajos de investigación para construir el conocimiento de la ingeniería del software.

### **BIBLIOGRAFIA**

- Apuntes de Cátedra.
- Ingeniería del Software. Un enfoque práctico . Roger S. Pressman.
- Ingeniería del Software. 7 ma. Edición. Kendall
- UML gota a gota . Martín Fowler con Kendall Scott.

### **EVALUACIÓN**

- **Exámenes Parciales:** 2 escritos e individuales con una duración de 2hs cátedra.
- **Trabajos Practicos:** 2 ( 1 por cada Cuatrimestre)
- **Coloquio de promoción directa:** 1 (para los alumnos que accedan por asistencia y puntaje)
- **Exámen Final:** teórico-práctico, escrito, individual. Según Calendario Escolar.

### **MODALIDAD DE CURSADO**

- REGULAR con cursado **PRESENCIAL**.
- REGULAR con cursado **SEMIPRESENCIAL**.
- **LIBRE**.

### **Para REGULARIZAR INGENIERIA DE SOFTWARE :**

- Tener un mínimo de 75% de asistencia a clases durante el tiempo que dure el cursado de la materia (anual ). Para el alumno que presente certificado de Trabajo y/o se encuentre en otras situaciones excepcionales, debidamente comprobadas, el mínimo es de 50%.
- Aprobar el 75% de los Trabajos Prácticos.
- Aprobar los 2(dos) Parciales, cada uno con su recuperatorio con nota mínima de 6(seis).

### **Para REGULARIZAR en forma SEMIPRESENCIAL**

- Tener un mínimo de 40% asistencia a clases.
- Aprobar el 100% de los trabajos Prácticos.
- Aprobar los 2(dos) Parciales, cada uno con su recuperatorio, con nota mínima de 6(seis).

### **Para acceder a PROMOCION DIRECTA**

- Cumplir con el % asistencia establecido por el régimen presencial.
- Aprobar el 100% de los Trabajos Prácticos.
- Aprobar los parciales con un mínimo de 8(ocho).
- Aprobar una instancia final integradora con un mínimo de 8(ocho) ,

### **OBSERVACIONES:**

- Si el alumno regulariza INGENIERIA DE SOFTWARE I pero no la promociona tiene derecho a un examen final escrito, en cualquiera de los turnos correspondientes, debiendo obtener calificación mínima de 6(seis).  
Mantiene la regularidad durante 3 años consecutivos a partir del primer turno correspondiente al año lectivo siguiente al del cursado.

### **ALUMNO LIBRE**

El estudiante libre deberá aprobar un exámen final, con modalidad combinada de escrito y oral, obteniendo una calificación mínima de 6 (seis).

.....  
*Aguilar, Mauro Ivan*  
Analista Programador