CIENCIA
DE LA COMPUTACIÓN

UTEC
UNIVERSIDAD DE INGENIERÍA
Y TECNOLOGÍA

CE2A

ÍNDICE

UNIVERSIDAD DE INGENIERÍA Y TECNOLOGÍA
SILABO 2020-2

1. ASIGNATURA

CS2901 - Ingeniería de Software I

2. DATOS GENERALES

2.1 Créditos: Cuatro (4) créditos
2.2 Horas de teoría: Dos (2) semanales
2.3 Horas de práctica: Cuatro (4) quincenales
2.4 Duración del período: Dieciséis (16) semanas
2.5 Condición: Obligatorio
2.6 Modalidad: Virtual
2.7 Requisitos:
- CS1103 Programación Orientada a Objetos II
- CS2701 Base de Datos I

3. PROFESORES

3.1 Profesor coordinador del curso
Jesus Bellido Angulo (jbellido@utec.edu.pe)
Horario de atención: Martes de 4 a 5PM

3.2 Profesor(es) instructor(es) del curso
Jesus Bellido Angulo (jbellido@utec.edu.pe)
Horario de atención: : Martes de 4 a 5PM

4. INTRODUCCIÓN AL CURSO

La tarea de desarrollar software, excepto para aplicaciones sumamente simples, exige la ejecución de un proceso de desarrollo bien definido. Los profesionales de esta área requieren un alto grado de conocimiento de los diferentes modelos de proceso de desarrollo, para que sean capaces de elegir el más idóneo para cada proyecto de desarrollo. Por otro lado, el desarrollo de sistemas de mediana y gran escala requiere del uso de biblioteca de patrones y componentes del dominio de técnicas relacionadas al diseño basado en componentes.

5. OBJETIVOS

Sesión 1: Describir cómo el proceso de ingeniería de requisitos apoya la obtención y validación de los requisitos de comportamiento.

Sesión 2: Interpretar un modelo de requisitos dada por un sistema de software simple.

Sesión 3: Describir los retos fundamentales y técnicas comunes que se utilizan para la obtención de requisitos.

Sesión 4: Identificar los requisitos funcionales y no funcionales en una especificación de requisitos dada por un sistema de software.

Sesión 5: Realizar una revisión de un conjunto de requisitos de software para determinar la calidad de los requisitos con respecto a las características de los buenos requisitos

Sesión 6: Aplicar elementos clave y métodos comunes para la obtención y el análisis para producir un conjunto de requisitos de software para un sistema de software de tamaño medio.

Sesión 7: Crear un prototipo de un sistema de software para reducir el riesgo en los requisitos.

Sesión 8: Formular los principios de diseño, incluyendo la separación de responsabilidades, encapsulamiento, acoplamiento y cohesión, y la encapsulación.

Sesión 9: Usar un paradigma de diseño para diseñar un sistema de software básico y explicar cómo los principios de diseño del sistema se han aplicado en este diseño.

Sesión 10: Construir modelos del diseño de un sistema de software simple los cuales son apropiado para el paradigma utilizado para diseñarlo.

Sesión 11: Seleccionar componentes adecuados para el uso en un diseño de un producto de software

Sesión 13: Describir uno o más patrones de diseño que podrían ser aplicables al diseño de un sistema de software simple.

Sesión 14: Construir modelos de verificación y validación de software por medio de pruebas de unidad, funcionales, de integración y de stress.

Sesión 15: Utilizar herramientas y metodologías de Integración Continua.

6. COMPETENCIAS

Las competencias que ha trabajar en este curso son:

- **a1**: Capacidad de aplicar conocimientos computación (*nivel 2*)
  El estudiante resuelve los problemas propuestos donde describe, selecciona y aplica lo aprendido de los fundamentos y principios de la Ingeniería de Software.

- **b** : Capacidad de resolver problemas complejos de computación y otras disciplinas relevantes en el dominio (*nivel 3*).
  El estudiante da solución a un problema real construyendo software donde aplica los conocimientos de computación adquiridos.

- **c1**: Capacidad para diseñar un sistema, un componente o un proceso para satisfacer las necesidades deseadas dentro de restricciones realistas (*nivel 3*).
  El estudiante aplica y analiza los criterios de diseño de un componente de software que cumplan con requerimientos previamente definidos.

- **d1**: Capacidad de trabajo en equipo (*nivel 2*)
  El estudiante forma parte de un equipo para alcanzar un objetivo común siguiendo una metodología.

- **h1**: Comprende el impacto de las soluciones computacionales en un contexto global, económico, ambiental y de la sociedad (*nivel 1*)
  A partir de un problema propuesto el estudiante analiza, diseña, implementa y prueba una solución con impacto en la sociedad.

7. RESULTADOS DE APRENDIZAJE

Al finalizar el curso de Ingeniería de software se espera:

RA1. Que el estudiante sea capaz de aplicar los estándares de calidad en el desarrollo de software.

RA 2. Que el estudiante sea capaz de modelar y construir software usando herramientas y metodologías adecuadas.

RA 3. Que el estudiante sea capaz de diseñar, implementar y verificar un sistema, proceso, programa o componente computacional apropiado para alcanzar las necesidades deseadas.

RA 4. Que el estudiante sea capaz de diseñar soluciones que aseguren la calidad, costo y time-to-market en los procesos de desarrollo.

8. TEMAS

1. **Introducción**
   1. Ingeniería de Software
   2. Metodologías Tradicionales
   3. Metodologías Ágiles
   4. Planificación
   5. Estimación
2. **Análisis**
   1. Requerimientos
      1. Obtención
      2. Requerimientos Funcionales

        3. Requerimientos No Funicionales
    2. User stories
    3. Use Cases
        1. Diagrama de Casos de Uso
3. **Diseño**
    1. Principios de diseño
        1. Cohesión
        2. Acoplamiento
    2. Patrones de Diseño de Diseño de Software
    3. Arquitectura de Software
        1. Diagrama de Arquitectura de Software
        2. Estilo de arquitectura de Software
    4. UML
        1. Diagrama de Clases
        2. Diagrama de Secuencia
4. **Implementación**
    1. Code Review
    2. Code Quality
5. **Pruebas**
    1. Unit Testing
    2. Functional Testing
    3. Automated Function Testing
    4. Integration Testing
    5. Stress Testing
6. **Deployment**
    1. Development Environment
    2. Production Environment
    3. Integración Continua

9. PLAN DE TRABAJO

9.1 Metodología

Este curso presenta por metodología activa el aprendizaje clásico y el aprendizaje basado en problemas; ambos son fundamentales para introducir al estudiante a los conceptos básicos y afianzar la base necesaria para los siguientes cursos de carrera. Ambos aumentan el interés del estudiante y promueven su compromiso en el aprendizaje.

9.2 Sesiones de teoría

Las sesiones de teoría se llevan a cabo en clases magistrales donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos. Se fomenta la participación individual y en equipo para exponer sus ideas, motivándolos con puntos adicionales en las diferentes etapas de la evaluación del curso.

9.3 Sesiones de práctica (laboratorio o taller)

En las sesiones de laboratorio se propondrán problemas para verificar que los alumnos hayan alcanzado el logro planteado para cada una de las unidades de aprendizaje, estas actividades les permitirá aplicar los conocimientos adquiridos durante las sesiones de teoría y además estos retos permitirá la evaluación el desempeño de los alumnos.

10. SISTEMA DE EVALUACIÓN

Parte de la evaluación continua serán presentaciones grupales, y prácticas individuales en clase y laboratorio. Las prácticas en clase serán presentadas el mismo día salvo algunas excepciones. Mientras que las prácticas de laboratorio tendrán una fecha de entrega.

	TEORÍA (T)	**LABORATORIO (L)**
EVALUACIÓN *La ponderación de la evaluación se hará si ambas partes están aprobadas	Evaluación Continua 1 (**C1**) (5%) Evaluación Continua 2 (**C2**) (5%) Examen Parcial (**E1**) (20%) Examen Final (**E2**) (20%)	Proyecto Parcial 1 (**P1**) (10%) Proyecto Parcial 2 (**P2**) (10%) Proyecto Parcial 3 (**P3**) (10%) Proyecto Parcial 4 (**P4**) (10%) Proyecto Final (**P5**) (10%)
	50%	50%
	100%	

Donde:

- **C1** (Semanas 1 - 7) : Tareas + Lecturas + Participación activa en clase.
- **C2** (Semanas 8 - 15) : Tareas + Lecturas + Participací´on activa en clase.
- **E1** : (Semana 8) Examen Parcial.
- **E2** : Examen Final tomado al finalizar el semestre.
- **P1** : Etapa de Análisis (Semana 5). Esta nota puede ser obtenida en dos intentos:
  - Primer Intento: Será evaluado sobre la nota 20.
  - Segundo Intento: Reemplazará la nota obtenida durante el primer intento (inclusive si es menor). La nota máxima en este intento es 15.
- **P2** : Etapa de Diseño (Semana 7). Esta nota puede ser obtenida en dos intentos:
  - Primer Intento: Será evaluado sobre la nota 20.
  - Segundo Intento: Reemplazará la nota obtenida durante el primer intento (inclusive si es menor). La nota máxima en este intento es 15.
- **P3:** Etapa de Implementación y Testeo (Semanas 8 - 13). La nota Dev es el promedio de tres entregas Dev1.1, Dev1.2, Dev1.3. La nota de cada entrega es otorgada por los profesores del curso y el cliente.
- **P4:** Etapa de Pruebas. Esta nota corresponde a la fase de pruebas de unidad, pruebas funcionales y de stress.

- **P4:** Demo Final (Semana 15). Esta nota es otorgada por los profesores del curso y el cliente.

Las rúbricas que permitirán medir las actividades más significativas del curso y que, además se relacionan con la evaluación de las competencias del estudiante son: enlace

11. SESIONES DE APOYO O TUTORÍAS

Este apartado permite formalizar los espacios de apoyo a los estudiantes y que éstos tengan la atención NECESARIA y el tiempo disponible para presentar sus dudas y consultas acerca del curso:

Semana	Fecha/ Hora	Tema a tratar	Objetivos de la sesión
2	11/09/2020 11:00 AM – 12:00 PM	Práctica	Metodologías de Desarrollo
4	25/09/2020 11:00 AM – 12:00 PM	Proyecto	Análisis: Requerimientos Funcionales y Diagrama de Clases
6	09/10/2020 11:00 AM – 12:00 PM	Proyecto	Diseño: Diagrama de Clases, Secuencia y de Arquitectura de Software
8	23/10/2020 11:00 AM – 12:00 PM	Examen 1	Responder a las dudas sobre las preguntas de examen 1, temas relacionados, contenidos, tipos de ejercicios.
10	06/11/2020 11:00 AM – 12:00 PM	Proyecto	Implementación: Revisar la demo funcional 1.0
12	20/11/2020 11:00 AM – 12:00 PM	Proyecto	Implementación: Revisar la demo funcional 1.2
14	04/12/2020 11:00 AM – 12:00 PM	Proyecto	Implementación: Revisar la demo funcional 2.0
16	18/12/2020 11:00 AM – 12:00 PM	Examen 2	Responder a las dudas sobre las preguntas de examen 2, temas relacionados, contenidos, tipos de ejercicios.

12. REFERENCIAS BIBLIOGRÁFICAS

- Sommerville, I. (2018). Software engineering 11th Edition.
- Martin, R. C. (2018). Clean architecture: a craftsman's guide to software structure and design. Prentice Hall.
- Martin, R. C. (2013). Clean Code-Refactoring, Patterns, Testen und Techniken für sauberen Code: Deutsche Ausgabe. MITP-Verlags GmbH & Co. KG.
- Freeman, E., Robson, E., Bates, B., & Sierra, K. (2008). Head first design patterns. " O'Reilly Media, Inc.".
- Eriksson, H. E., Penker, M., Lyons, B., & Fado, D. (2003). UML 2 Toolkit, CafeScribe (Vol. 26). John Wiley & Sons.