
Universidad de Ingeniería y Tecnología

Sílabo del curso – Periodo 2017-1

- 1. Código del curso y nombre:** CS1100 – Introducción a la Ciencia de la Computación
- 2. Créditos:** 4 créditos
- 3. Horas por sesión (teoría y laboratorio):** 2 – teoría; 4 – laboratorio
Número total de sesiones por tipo: 14 – teoría; 14 – laboratorio
- 4. Nombre, e-mail y horas de atención del instructor o coordinador del curso:**

Coordinador:

Ernesto Cuadros ecuaedros@utec.edu.pe

Atención previa coordinación con el profesor.

Instructor(es):

- Katia CánepaVega kcanepa@utec.edu.pe
- Jorge Alvarado
- María Hilda Bermejo mbermejo@utec.edu.pe
- Juan Carlos Bueno jbueno@utec.edu.pe
- Ernesto Bringas ibringas@utec.edu.pe
- Iván Calle icalle@utec.edu.pe
- Teófilo Chambilla tchambilla@utec.edu.pe
- José Díaz jdiazl@utec.edu.pe
- Jaime Farfán jfarfan@utec.edu.pe
- José Fiestas
- Mariano Melgar
- Randiel Melgarejo rmelgarejo@utec.edu.pe
- Patricio Morriberón
- Rubén Rivas rrivas@utec.edu.pe

Atención previa coordinación con el profesor.

- 5. Bibliografía: libro, título, autor y años de publicación:**

a. Básica:

- Brookshear J.G.; Brylow D.; Computer Science and overview; 12th Edition; Global Edition; Pearson, 2015

b. Complementaria:

- Gries, P.; Campbell, J.; Montoyo, J.; Practical Programming, 2nd Edition An Introduction to Computer Science Using Python 3; The Pragmatic Bookshelf; Dallas, Texas, Raleigh, North Carolina, 2013. ISBN-13: 978-1-93778-545-1

6. Información del curso

a. Breve descripción del contenido del curso

Durante el desarrollo del curso, se busca que el alumno comprenda y luego aplique el pensamiento computacional. El alumno comprenderá la relación entre el binomio Hombre-Computador, valorando su interrelación y simbiosis al resolver problemas computacionales relacionados a su carrera. Así mismo reconocerá la “escalabilidad” como una de las características primordiales al plantear un algoritmo susceptible de ser codificado a través de un lenguaje de programación.

El curso se imparte recorriendo dos hilos claramente definidos:

a) Sesiones teóricas, en donde a través de clases magistrales se muestra a los alumnos, las áreas mas importantes del cuerpo de conocimientos de la Ciencia de la Computación.

b) Sesiones en laboratorio, en donde los alumnos serán capaces de crear algoritmos considerando la escalabilidad, como uno de los factores primordiales en su concepción.

Se podría utilizar uno o varios lenguajes de programación, para codificar los programas, ya que el curso va focalizado mas al desarrollo de las habilidades para crear buenos algoritmos, que al aprendizaje de la sintaxis de un determinado lenguaje de programación.

La relación entre los temas vistos durante las clases magistrales y las sesiones de laboratorio, se establece a través de aplicaciones y/o instrumentos que permitan a los alumnos interiorizar los conceptos. Entre ellos: Scratch, Instabase, Kahoot, entre otros.

b. Prerrequisitos o correquisitos: Ninguno.

c. Indicar si es un curso obligatorio o electivo: Obligatorio.

7. Objetivos del curso

a. Competencias

Al finalizar el curso el alumno estará en la capacidad de:

- a : Aplicar conocimientos de computación y de matemáticas (nivel 2)
- b : Analizar problemas, identificar y definir los requerimientos computacionales (nivel 2)
- c : Diseñar, implementar y evaluar un sistema, proceso, componente o programa computacional (nivel 2)
- i : Utilizar técnicas y herramientas actuales (nivel 2)
- k : Aplicar los principios de desarrollo y diseño en software de complejidad variable (nivel 1)

El curso aborda los siguientes resultados del estudiante ABET: a,b,c,i,k

b. Resultados de aprendizaje

Logro del curso: Al finalizar el curso, el alumno entiende la importancia de las diferentes áreas que conforman el cuerpo de conocimientos de la Ciencia de la Computación y su influencia en toda área del saber; asimismo el estudiante diseña algoritmos que den solución a problemas puntuales y susceptibles de ser codificados en un lenguaje de programación, valorando la importancia de aplicar el pensamiento computacional al proponer sus algoritmos.

De tal modo que:

- Conoce los tópicos del cuerpo de conocimiento de la Ciencia de la Computación y comprende cómo inciden en las diferentes áreas del saber.
- Comprende la importancia del uso del Computador en todo ámbito del saber.
- Comprende la importancia de aplicar el pensamiento computacional al proponer un algoritmo susceptible de ser codificado en un lenguaje de programación.
- Escribe programas en un lenguaje de programación, utilizando: datos simples y complejos, estructuras de control, funciones, lisas, Diccionarios y Archivos.

8. Lista de temas a estudiar durante el curso

Sesiones de teoría:

1. ¿Qué es Computación?
2. Almacenamiento de datos
3. Manipulación de Datos
4. Networking e internet
5. Algoritmos y eficiencia
6. Lenguajes de Programación
7. Ingeniería de Software
8. Abstracción de datos
9. Sistemas de Bases de datos
10. Computer Graphics
11. Inteligencia Artificial
12. Teoría de la Computación

Sesiones de Laboratorio:

1. Algoritmos y Scratch (Lenguaje de Programación visual)
2. Entorno de programación, instrucciones de entrada y salida de datos. Variable, constante, tipos de datos y operadores, expresiones y precedencia de operadores
3. Estructuras de control selectivas
4. Estructuras de control repetitivas
5. Introducción a Objetos, clases, referencia a objetos y alcance
6. Módulos y uso de funciones definidas en librerías matemáticas y Urllib
7. Listas y String
8. Diccionarios, Tuplas
9. Listas multidimensionales

10. Archivos

9. Metodología y sistema de evaluación

Metodología:

Sesiones de teoría:

Durante las sesiones de teoría, se darán a conocer las diferentes áreas del cuerpo de conocimientos de la Ciencia de la Computación. Estos conocimientos se impartirán en clases magistrales en el auditorio de la universidad, en donde se realizarán actividades que propicien un aprendizaje activo, con dinámicas que permitan a los estudiantes interiorizar los conceptos.

En cada semana se presentará un tópico diferente del campo de la Ciencia de la Computación, los alumnos revisarán previamente la información del capítulo correspondiente e ingresarán a la sesión llevando un resumen escrito en puño letra. Esto se evaluará a lo largo del ciclo y se obtendrá la nota bajo el rubro de **RT Resúmenes de teoría**.

Para verificar que los logros de aprendizaje planteados hayan sido alcanzados, los estudiantes realizarán dos trabajos, una a mitad de ciclo y otro a fin de ciclo. Estos trabajos podrían ser: videos, blogs o la construcción de páginas web, en donde ellos construyan “artefactos” que permitan dar a conocer las diferentes áreas del cuerpo de conocimientos de la Ciencia de la Computación. Estas actividades serán grupales y evaluadas bajo el rubro de **Evaluaciones Teóricas (ET)**.

Sesiones de Laboratorio:

Se realizan en un laboratorio de cómputo, 4 horas por semana. En estas sesiones los alumnos desarrollan sus habilidades para diseñar algoritmos considerando las características del pensamiento computacional. Los algoritmos que diseñen luego serán codificados utilizando el lenguaje de programación que se indique. Para verificar que los alumnos hayan alcanzado el logro de aprendizaje de cada unidad, resolverán un set de problemas que serán testeados de manera automática a través del HackerRank. Estas actividades serán evaluadas a través del rubro **Set de Problemas (SP)**

Asimismo, durante las sesiones de laboratorio, los estudiantes en grupo realizarán actividades que les permitan ir descubriendo los conocimientos y desarrollando sus habilidades para plantear y probar sus algoritmos. Estas actividades serán evaluadas a través del rubro de **Desempeño en clase (DC)**.

Durante el ciclo, los alumnos desarrollarán 3 proyectos, cuyo grado de complejidad irá aumentado de acuerdo al avance de los contenidos del curso. Estos proyectos serán evaluados a través del rubro **Proyecto (PY)**.

Los proyectos serán realizados de manera grupal a través de un trabajo colaborativo, en el cual cada miembro del grupo tendrá asignado un rol que ejercerá por una semana, al cabo de la cual habrá rotación de roles y cada alumno ejercerá un rol distinto en cada semana hasta culminar el proyecto.

Sistema de Evaluación:

Los contenidos teóricos se evaluarán a través de:

ET Evaluaciones Teóricas: 2 evaluaciones (ET1 y ET2)

RT Resúmenes de Teoría (1)

Los contenidos desarrollados en las sesiones de laboratorios, se evaluarán a través de 3 rubros:

DC Desempeño en clase: 2 (DC1 y DC2)

SP Set de Problemas: 6 evaluaciones (SP1, SP2 .. SP6).

PY Proyectos: 3 evaluaciones (PY1, PY2, PY3)

El promedio Final del curso se calcula a través de la siguiente fórmula:

$$\text{Nota Final} = 0.2 \text{ ET} + 0.1 \text{ RT} + 0.1 \text{ DC} + 0.20 \text{ SP} + 0.4 \text{ PY}$$

Donde las siglas significan:

ET = Evaluación Teórica (2)

RT = Resúmenes de Teoría (1)

DC = Desempeño en Clase (2)

SP = Set de Problemas (6)

PY = Proyecto (3)

Donde:

ET = 0.30 (ET1) + 0.70 (ET2)

DC = Todas tienen el mismo peso

SP = Todas tienen el mismo peso

PY = 0.10 (PY1) + 0.30 (PY2) + 0.60 (PY3)