

Universidad Católica San Pablo (UCSP)
Escuela Profesional de
Ciencia de la Computación
SILABO



CS100. Introducción de Ciencia de la Computación
(Obligatorio)

2023-I

1. Información general

1.1 Escuela	:	Ciencia de la Computación
1.2 Curso	:	CS100. Introducción de Ciencia de la Computación
1.3 Semestre	:	2 ^{do} Semestre.
1.4 Prerrequisitos	:	Ninguno
1.5 Condición	:	Obligatorio
1.6 Modalidad de aprendizaje	:	Virtual
1.7 horas	:	2 HT; 2 HP;
1.8 Créditos	:	3
1.9 Plan	:	Plan Curricular 2016

2. Profesores

Titular

- Ernesto Cuadros-Vargas <ecuadros@ucsp.edu.pe>
 - Doctor en Ciencia de la Computación, ICMC-USP, Brasil, 2004.
 - Master en Ciencia de la Computación, ICMC-USP, Brasil, 1998.

3. Fundamentación del curso

La Ciencia de la Computación es un campo de estudio enorme con muchas especialidades y aplicaciones. Este curso brindará a sus participantes, una visión panorámica de la informática y mostrará sus campos más representativos, como son: Algoritmos, Estructuras de de Datos, Sistemas Operativos, Bases de Datos, etc.

4. Resumen

1. Introducción 2. Almacenamiento de datos 3. Tratamiento de Datos 4. Visión general de Sistemas Operativos 5. Redes e Internet 6. Algoritmos 7. Lenguajes de Programacion 8. Procesos de Software 9. Algoritmos y Estructuras de Datos fundamentales 10. Base de Datos 11. Inteligencia Artificial 12. Computación Gráfica

5. Objetivos Generales

- Brindar un panorama del área del conocimiento que es cubierta en la ciencia de la computación.

6. Contribución a los resultados (Outcomes)

Esta disciplina contribuye al logro de los siguientes resultados de la carrera:

- 1) S.O. Analizar un problema computacional complejo y aplicar los principios computacionales y otras disciplinas relevantes para identificar soluciones. **(Familiarizarse)**
- 4) S.O. Reconocer las responsabilidades profesionales y hacer juicios informados en el campo profesional de computación con principios éticos. **(Familiarizarse)**
- 5) S.O. Funcionar efectivamente como miembro o líder de un equipo involucrado en actividades apropiadas a la disciplina del programa. **(Familiarizarse)**
- 6) S.O. Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. **(Familiarizarse)**
- 7) S.O. Desarrollar tecnología computacional buscando el bien común, aportando con formación humana, capacidades científicas, tecnológicas y profesionales para solucionar problemas sociales de nuestro entorno. **(Familiarizarse)**

7. Contenido**UNIDAD 1: Introducción (2)****Competencias: 1****Contenido**

- Introducción a la computación.
- Historia de la computación.

Objetivos Generales

- Incentivar a los alumnos el estudio de Computación como una ciencia. [Familiarizarse]

Lecturas: Brookshear (2015)

UNIDAD 2: Almacenamiento de datos (2)**Competencias: 1,5****Contenido**

- Lógica proposicional.
- Conectores lógicos.
- Tablas de verdad.
- Forma normal (conjuntiva y disyuntiva)
- Bits, Bytes y Words.
- Representación de datos numérica y bases numéricas.

Objetivos Generales

- Convertir declaraciones lógicas desde el lenguaje informal a expresiones de lógica proposicional y de predicados [Familiarizarse]
- Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de fórmulas y cálculo de formas normales [Familiarizarse]
- Explicar por qué en computación todo es datos, inclusive las instrucciones [Familiarizarse].
- Explicar las razones de usar formatos alternativos para representar datos numéricos [Familiarizarse].

Lecturas: brookshear2014

UNIDAD 3: Tratamiento de Datos (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Revisión e historia de la Arquitectura de Computadores. • Organización Básica de la Máquina de Von Neumann. • Unidad de Control. • <i>Instruction sets</i> y tipos (manipulación de información, control, I/O) • Assembler y Programación en Lenguaje de Máquina. • Organización y Operaciones de la Memoria Principal. 	<ul style="list-style-type: none"> • Convertir declaraciones lógicas desde el lenguaje informal a expresiones de lógica proposicional y de predicados [Familiarizarse] • Aplicar métodos formales de simbolismo proposicional y lógica de predicados, como el cálculo de la validez de formulas y cálculo de formas normales [Familiarizarse] • Explicar la organización de la maquina clásica de von Neumann y sus principales unidades funcionales [Familiarizarse]. • Describir cómo se ejecuta una instrucción en una máquina de von Neumann con extensión para hebras, sincronización multiproceso y ejecución SIMD (máquina vectorial) [Familiarizarse].
Lecturas: brookshear2014	

UNIDAD 4: Visión general de Sistemas Operativos (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Papel y el propósito del sistema operativo. • Funcionalidad de un sistema operativo típico. • Procesos y subprocesos. • <i>Scheduling</i> y políticas. 	<ul style="list-style-type: none"> • Explicar los objetivos y funciones de un sistema operativo moderno [Familiarizarse]
Lecturas: brookshear2014	

UNIDAD 5: Redes e Internet (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Organización de la Internet (proveedores de servicios de Internet, proveedores de contenido, etc) • Piezas físicas de una red, incluidos hosts, routers, switches, ISPs, inalámbrico, LAN, punto de acceso y firewalls. • Esquemas de denominación y dirección (DNS, direcciones IP, identificadores de recursos uniformes, etc) 	<ul style="list-style-type: none"> • Articular la organización de la Internet [Familiarizarse]. • Listar y definir la terminología de red apropiada [Familiarizarse]. • Describir la estructura en capas de una arquitectura típica en red [Familiarizarse].
Lecturas: brookshear2014	

UNIDAD 6: Algoritmos (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Conceptos y propiedades de los algoritmos <ul style="list-style-type: none"> – Comparación informal de la eficiencia de los algoritmos (ej., conteo de operaciones) • Rol de los algoritmos en el proceso de solución de problemas • Estrategias de solución de problemas <ul style="list-style-type: none"> – Funciones matemáticas iterativas y recursivas – Recorrido iterativo y recursivo en estructura de datos – Estrategias Divide y Conquistar • Conceptos y principios fundamentales de diseño <ul style="list-style-type: none"> – Abstracción – Descomposición de Program – Encapsulamiento y camuflaje de información – Separación de comportamiento y aplicación 	<ul style="list-style-type: none"> • Discute la importancia de los algoritmos en el proceso de solución de un problema [Familiarizarse]. • Discute como un problema puede ser resuelto por múltiples algoritmos, cada uno con propiedades diferentes [Familiarizarse]. • Crea algoritmos para resolver problemas simples [Familiarizarse].
Lecturas: brookshear2014	

UNIDAD 7: Lenguajes de Programacion (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Programas que tienen otros programas como entrada tales como interpretes, compiladores, revisores de tipos y generadores de documentación. • Árboles de sintaxis abstracta, para contrastar la sintaxis correcta. • Estructuras de datos que representan código para ejecución, traducción o transmisión. 	<ul style="list-style-type: none"> • Explicar como programas que procesan otros programas tratan a los otros programas como su entrada de datos [Familiarizarse]. • Describir un árbol de sintaxis abstracto para un lenguaje pequeño [Familiarizarse]. • Describir los beneficios de tener representaciones de programas que no sean cadenas de código fuente [Familiarizarse].
Lecturas: brookshear2014	

UNIDAD 8: Procesos de Software (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Introducción a modelos del proceso de software (e.g., cascada, incremental, agil): <ul style="list-style-type: none"> – Actividades con ciclos de vida de software. • Consideraciones a nivel de sistemas, ejem., la interacción del software con su entorno. • Conceptos de calidad de software. 	<ul style="list-style-type: none"> • Describa cómo el software puede interactuar y participar en varios sistemas, incluyendo la gestión de información, integración, control de procesos y sistemas de comunicaciones [Familiarizarse]. • Describir las ventajas y desventajas relativas entre varios modelos importantes de procesos (por ejemplo, la cascada, iterativo y ágil) [Familiarizarse].
Lecturas: brookshear2014	

UNIDAD 9: Algoritmos y Estructuras de Datos fundamentales (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Tipos Abstractos de datos y sus implementaciones: <ul style="list-style-type: none"> – Pilas – Colas – Colas de prioridad – Conjuntos – Mapas • Listas enlazadas • Algoritmos numéricos simples, tales como el cálculo de la media de una lista de números, encontrar el mínimo y máximo. • Algoritmos de búsqueda secuencial y binaria. • Árboles de búsqueda binaria: <ul style="list-style-type: none"> – Operaciones comunes en árboles de búsqueda binaria como seleccionar el mínimo, máximo, insertar, eliminar, recorrido en árboles. • Árboles balanceados (ej. árboles AVL, Árboles red-black, Árboles biselados (splay trees), Treaps) • Estructuras de Datos Avanzadas (ej. B-Trees, Fibonacci Heaps) 	<ul style="list-style-type: none"> • Implementar algoritmos numéricos básicos [Familiarizarse]. • Implementar algoritmos de búsqueda simple y explicar las diferencias en sus tiempos de complejidad [Familiarizarse]. • Discutir factores otros que no sean eficiencia computacional que influyan en la elección de algoritmos, tales como tiempo de programación, mantenibilidad, y el uso de patrones específicos de la aplicación en los datos de entrada [Familiarizarse].
Lecturas: brookshear2014	

UNIDAD 10: Base de Datos (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Enfoque y Evolución de Sistemas de Bases de Datos. • Componentes del Sistema de Bases de Datos. • Diseño de las funciones principales de un DBMS. • Arquitectura de base de datos e independencia de datos. • Uso de un lenguaje de consulta declarativa. 	<ul style="list-style-type: none"> • Explica las características que distinguen un esquema de base de datos de aquellos basados en la programación de archivos de datos [Familiarizarse]. • Describe los diseños más comunes para los componentes base de sistemas de bases de datos incluyendo el optimizador de consultas, ejecutor de consultas, administrador de almacenamiento, métodos de acceso y procesador de transacciones [Familiarizarse].
Lecturas: brookshear2014	

UNIDAD 11: Inteligencia Artificial (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Descripción general de los problemas de Inteligencia Artificial, ejemplos recientes de aplicaciones de Inteligencia artificial. • ¿Qué es comportamiento inteligente? <ul style="list-style-type: none"> – El Test de Turing – Razonamiento Racional versus No Racional • Definición y ejemplos de una amplia variedad de tareas de aprendizaje de máquina • Aprendizaje supervisado <ul style="list-style-type: none"> – Aprendizaje basado en árboles de decisión – Aprendizaje basado en redes neuronales – Aprendizaje basado en máquinas de soporte vectorial (<i>Support vector machines SVMs</i>) 	<ul style="list-style-type: none"> • Determinando las características de un problema dado que sistemas inteligentes deberían resolver [Familiarizarse].
Lecturas: brookshear2014	

UNIDAD 12: Computación Gráfica (2)	
Competencias: 1,5	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> • Aplicaciones multimedia, incluyendo interfaces de usuario, edición de audio y vídeo, motores de juego, cad, visualización, realidad virtual. • Digitalización de datos analógicos, la resolución y los límites de la percepción humana, por ejemplo, los píxeles de la pantalla visual, puntos para impresoras láser y muestras de audio • Algoritmos de visualización y gráficos. • Técnicas de procesamiento de imágenes. 	<ul style="list-style-type: none"> • Identificar usos comunes de presentaciones digitales de humanos (por ejemplo, computación gráfica, sonido) [Familiarizarse]. • Explicar en términos generales cómo las señales analógicas pueden ser representadas por muestras discretas, por ejemplo, cómo las imágenes pueden ser representadas por píxeles [Familiarizarse].
Lecturas: brookshear2014	

8. Metodología

1. El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.
2. El profesor del curso presentará demostraciones para fundamentar clases teóricas.
3. El profesor y los alumnos realizarán prácticas
4. Los alumnos deberán asistir a clase habiendo leído lo que el profesor va a presentar. De esta manera se facilitará la comprensión y los estudiantes estarán en mejores condiciones de hacer consultas en clase.

9. Evaluar

Evaluación Continua 1 : 20 %

Examen parcial : 30 %

Evaluación Continua 2 : 20 %

Examen final : 30 %

References

Brookshear, J. G. (2015). *Computer Science: An Overview*. 12th. Addison-Wesley.