

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



**CS3P2. Cloud Computing (Obligatorio)**

**1. Información general**

1.1 Escuela	:	Ciencia de la Computación
1.2 Curso	:	CS3P2. Cloud Computing
1.3 Semestre	:	10 <sup>mo</sup> Semestre.
1.4 Prerrequisitos	:	CS370. Big Data. (9 <sup>no</sup> Sem)
1.5 Condición	:	Obligatorio
1.6 Modalidad de aprendizaje	:	Presencial
1.7 horas	:	1 HT; 4 HP;
1.8 Créditos	:	3
1.9 Plan	:	Plan Curricular 2016

**2. Profesores**

**Titular**

- Alvaro Henry Mamani-Aliaga <ahmamani@ucsp.edu.pe>
  - Doctor en Ciencia de la Computación, UNSA, Perú, 2019.
  - Master en Ciencia de la Computación, IME-USP, Brasil, 2011.

**3. Fundamentación del curso**

Para entender las técnicas computacionales avanzadas, los estudiantes deberán tener un fuerte conocimiento de las diversas estructuras discretas, estructuras que serán implementadas y usadas en laboratorio en el lenguaje de programación.

**4. Resumen**

1. Bases teóricas de la Computación en la Nube 2. Procesamiento de datos 3. Virtualización, Contenerización 4. Tendencias en Computación en la Nube 5. Sistemas distribuidos

**5. Objetivos Generales**

- Que el alumno sea capaz de modelar problemas de ciencia de la computación usando grafos y árboles relacionados con estructuras de datos.
- Que el alumno aplicar eficientemente estrategias de recorrido para poder buscar datos de una manera óptima.

**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. (**Usar**)
- 6) S.O. Aplicar la teoría de la computación y los fundamentos del desarrollo de software para producir soluciones basadas en computación. (**Usar**)

**7. Contenido**

<b>UNIDAD 1: Bases teóricas de la Computación en la Nube (12)</b>	
<b>Resultados del estudiante: 1,6</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• Introducción a la Computación en la Nube</li> <li>• Modelos de Servicio en Computación en la Nube</li> <li>• Tipos de despliegue en Computación en la Nube</li> <li>• Infraestructura y Centros de Datos</li> <li>• Tendencias en Investigación en Computación en la Nube</li> </ul>	<ul style="list-style-type: none"> <li>• Comprender los conceptos relacionados a la Computación en la Nube.</li> <li>• Comprender la infraestructura y componentes de un Centro de Datos.</li> <li>• Entender los modelos de servicio y tipos de despliegue en Computación en la Nube.</li> <li>• Conocer las tendencias en investigación en el área de Computación en la Nube.</li> </ul>
<b>Lecturas: aboveTheCloud, surveySecurity, mobileCloud</b>	

<b>UNIDAD 2: Procesamiento de datos (15)</b>	
<b>Resultados del estudiante: 1,6</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• Introducción al framework Hadoop.</li> <li>• Sistema de Archivo Distribuido de Hadoop.</li> <li>• Introducción al modelo de programación MapReduce.</li> <li>• Introducción al framework Spark.</li> </ul>	<ul style="list-style-type: none"> <li>• Entender los conceptos relacionados al framework Hadoop.</li> <li>• Entender los conceptos relacionados al Sistema de Archivo Distribuido de Hadoop.</li> <li>• Entender y aplicar el modelo de programación MapReduce.</li> <li>• Entender los conceptos relacionados al framework Spark.</li> </ul>
<b>Lecturas: mapreduce, spark, yarn</b>	

<b>UNIDAD 3: Virtualización, Contenerización (15)</b>	
<b>Resultados del estudiante: 1,6</b>	
<b>Contenido</b>	<b>Objetivos Generales</b>
<ul style="list-style-type: none"> <li>• Introducción a la Contenerización.</li> <li>• Evolución de la Contenerización.</li> <li>• Diferencias entre Contenerización y Virtualización.</li> </ul>	<ul style="list-style-type: none"> <li>• Entender el concepto de Contenerización.</li> <li>• Crear y utilizar contenedores.</li> <li>• Entender las diferencias entre Contenerización y Virtualización</li> </ul>
<b>Lecturas: CborgOmegaKubernetes, borg, ContainerizationPaaSCloud, VirtualizationContainerization</b>	

UNIDAD 4: Tendencias en Computación en la Nube (12)	
Resultados del estudiante: 1,6	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Autoescalamiento.</li> <li>• Infraestructura como código.</li> <li>• Computación sin servidor.</li> </ul>	<ul style="list-style-type: none"> <li>• Entender las diferentes formas de autoescalamiento.</li> <li>• Utilizar las diferentes herramientas para la administración como código en la nube.</li> <li>• Entender el paradigma de Computación sin servidor.</li> </ul>
<b>Lecturas: Cormen2009, Preparata, Berg</b>	

UNIDAD 5: Sistemas distribuídos (15)	
Resultados del estudiante:	
Contenido	Objetivos Generales
<ul style="list-style-type: none"> <li>• Fallos: <ul style="list-style-type: none"> <li>– Fallos basados en red (incluyendo particiones) y fallos basados en nodos</li> <li>– Impacto en garantías a nivel de sistema (p.e., disponibilidad)</li> </ul> </li> <li>• Envío de mensajes distribuido: <ul style="list-style-type: none"> <li>– Conversión y transmisión de datos</li> <li>– Sockets</li> <li>– Secuenciamiento de mensajes</li> <li>– Almacenando <i>Buffering</i>, reenviando y desechando mensajes</li> </ul> </li> <li>• Compensaciones de diseño para Sistemas Distribuidos: <ul style="list-style-type: none"> <li>– Latencia versus rendimiento</li> <li>– Consistencia, disponibilidad, tolerancia de particiones</li> </ul> </li> <li>• Diseño de Servicio Distribuido: <ul style="list-style-type: none"> <li>– Protocolos y servicios Stateful versus stateless</li> <li>– Diseños de Sesión (basados en la conexión)</li> <li>– Diseños reactivos (provocados por E/S) y diseños de múltiples hilos</li> </ul> </li> <li>• Algoritmos de Distribución de Núcleos: <ul style="list-style-type: none"> <li>– Elección, descubrimiento</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Distinguir las fallas de red de otros tipos de fallas [Familiarizarse]</li> <li>• Explicar por qué estructuras de sincronización como cerraduras simples (<i>locks</i>) no son útiles en la presencia de fallas distribuidas [Familiarizarse]</li> <li>• Escribir un programa que realiza cualquier proceso de <i>marshalling</i> requerido y la conversión en unidades de mensajes, tales como paquetes, para comunicar datos importantes entre dos <i>hosts</i> [Usar]</li> <li>• Medir el rendimiento observado y la latencia de la respuesta a través de los <i>hosts</i> en una red dada [Usar]</li> <li>• Explicar por qué un sistema distribuido no puede ser simultáneamente Consistente (<i>Consistent</i>), Disponible (<i>Available</i>) y Tolerante a fallas (<i>Partition tolerant</i>). [Familiarizarse]</li> <li>• Implementar un servidor sencillo - por ejemplo, un servicio de corrección ortográfica [Usar]</li> <li>• Explicar las ventajas y desventajas entre: <i>overhead</i>, escalabilidad y tolerancia a fallas entre escoger un diseño sin estado (<i>stateless</i>) y un diseño con estado (<i>stateful</i>) para un determinado servicio [Familiarizarse]</li> <li>• Describir los desafíos en la escalabilidad, asociados con un servicio creciente para soportar muchos clientes, así como los asociados con un servicio que tendrá transitoriamente muchos clientes [Familiarizarse]</li> <li>• Dar ejemplos de problemas donde algoritmos de consenso son requeridos, por ejemplo, la elección de líder [Usar]</li> </ul>
<b>Lecturas: Coulouris et al. (2011)</b>	

## 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 Sesiones Teóricas:**

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.

**Sesiones Prácticas:**

Las sesiones prácticas se llevan en clase donde se desarrollan una serie de ejercicios y/o conceptos prácticos mediante planteamiento de problemas, la resolución de problemas, ejercicios puntuales y/o en contextos aplicativos.

**Sistema de Evaluación:**

La nota final se obtiene a través de:

EVALUACIONES PERMANENTES	EVALUACIONES
<b>Evaluación Permanente 1</b> : 24 %	<b>Evaluación Parcial</b> : 20 %
<b>Evaluación Permanente 2</b> : 36 %	<b>Evaluación Final</b> : 20 %
60%	40%

Donde:

Evaluación Permanente: Comprende trabajos grupales, participación activa en clase, test de ejercicios.

- Permanente 1 (Semanas 1 - 9)
- Permanente 2 (Semanas 10 - 17)

Para aprobar el curso, el alumno debe obtener 11.5 o más en la nota final.

**References**

Coulouris, George et al. (2011). *Distributed Systems: Concepts and Design*. 5th. Addison-Wesley Publishing Company: USA. ISBN: 0132143011, 9780132143011.