



# Universidad Nacional de Ingeniería (UNI)

Escuela Profesional de

Ciberseguridad

Sílabo 2024-II

## 1. CURSO

CS3P2. Cloud Computing (Obligatorio)

## 2. INFORMACIÓN GENERAL

- 2.1 Curso : CS3P2. Cloud Computing
- 2.2 Semestre : 10<sup>mo</sup> Semestre.
- 2.3 Créditos : 3
- 2.4 horas : 1 HT; 4 HP;
- 2.5 Duración del periodo : 16 semanas
- 2.6 Condición : Obligatorio
- 2.7 Modalidad de aprendizaje : Presencial
- 2.8 Prerrequisitos : CS370. Big Data. (9<sup>no</sup> Sem)

## 3. PROFESORES

Atención previa coordinación con el profesor

## 4. INTRODUCCIÓN AL 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.

## 5. OBJETIVOS

- 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. RESULTADOS DEL ESTUDIANTE

1) ()

6) Aplicar principios y prácticas de seguridad para mantener las operaciones en presencia de riesgos y amenazas.()

## 7. TEMAS

Unidad 1: Bases teóricas de la Computación en la Nube (12 horas)	
Resultados esperados: 1,6	
Temas	Objetivos de Aprendizaje
<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>
Lecturas : [aboveTheCloud], [surveySecurity], [mobileCloud]	

Unidad 2: Procesamiento de datos (15 horas)	
Resultados esperados: 1,6	
Temas	Objetivos de Aprendizaje
<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>
Lecturas : [mapreduce], [spark], [yarn]	

Unidad 3: Virtualización, Contenerización (15 horas)	
Resultados esperados: 1,6	
Temas	Objetivos de Aprendizaje
<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>
Lecturas : [CborgOmegaKubernetes], [borg], [ContainerizationPaaSCloud], [VirtualizationContainerization]	

Unidad 4: Tendencias en Computación en la Nube (12 horas)	
Resultados esperados: 1,6	
Temas	Objetivos de Aprendizaje
<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>
Lecturas : [Cormen2009], [Preparata], [Berg]	

Unidad 5: Sistemas distribuidos (15 horas)	
Resultados esperados:	
Temas	Objetivos de Aprendizaje
<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>
Lecturas : [coulouris]	

## 8. PLAN DE TRABAJO

### 8.1 Metodología

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.

### 8.2 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.

### 8.3 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.

## 9. SISTEMA DE EVALUACIÓN

\*\*\*\*\* EVALUATION MISSING \*\*\*\*\*

## 10. BIBLIOGRAFÍA BÁSICA