

**San Pablo Catholic University (UCSP)**  
**Undergraduate Program in**  
**Computer Science**  
**SILABO**



**CS3P2. Cloud Computing (Mandatory)**

**1. General information**

1.1 School	:	Ciencia de la Computación
1.2 Course	:	CS3P2. Cloud Computing
1.3 Semester	:	10 <sup>mo</sup> Semestre.
1.4 Prerequisites	:	CS370. Big Data. (9 <sup>th</sup> Sem)
1.5 Type of course	:	Mandatory
1.6 Learning modality	:	Virtual
1.7 Horas	:	1 HT; 2 HP; 2 HL;
1.8 Credits	:	3

**2. Professors**

**3. Course foundation**

In order to understand the advanced computational techniques, the students must have a strong knowledge of the various discrete structures, structures that will be implemented and used in the laboratory in the programming language.

**4. Summary**

1. Distributed Systems 2. Cloud Computing 3. Centros de Procesamiento de Datos 4. Cloud Computing 5. Cloud Computing 6. Modelos de Programación

**5. Generales Goals**

- That the student is able to model computer science problems using graphs and trees related to data structures.
- That the student apply efficient travel strategies to be able to search data in an optimal way.

**6. Contribution to Outcomes**

This discipline contributes to the achievement of the following outcomes:

- a) An ability to apply knowledge of mathematics, science. (**Usage**)
- b) An ability to design and conduct experiments, as well as to analyze and interpret data. (**Usage**)
- i) An ability to use the techniques, skills, and modern computing tools necessary for computing practice. (**Usage**)
- j) Apply the mathematical basis, principles of algorithms and the theory of Computer Science in the modeling and design of computational systems in such a way as to demonstrate understanding of the equilibrium points involved in the chosen option. (**Usage**)

**7. Content**

UNIT 1: Distributed Systems (15)	
Competences: a,b	
Content	Generales Goals
<ul style="list-style-type: none"> <li>• Faults (cross-reference OS/Fault Tolerance) <ul style="list-style-type: none"> <li>– Network-based (including partitions) and node-based failures</li> <li>– Impact on system-wide guarantees (e.g., availability)</li> </ul> </li> <li>• Distributed message sending <ul style="list-style-type: none"> <li>– Data conversion and transmission</li> <li>– Sockets</li> <li>– Message sequencing</li> <li>– Buffering, retrying, and dropping messages</li> </ul> </li> <li>• Distributed system design tradeoffs <ul style="list-style-type: none"> <li>– Latency versus throughput</li> <li>– Consistency, availability, partition tolerance</li> </ul> </li> <li>• Distributed service design <ul style="list-style-type: none"> <li>– Stateful versus stateless protocols and services</li> <li>– Session (connection-based) designs</li> <li>– Reactive (IO-triggered) and multithreaded designs</li> </ul> </li> <li>• Core distributed algorithms <ul style="list-style-type: none"> <li>– Election, discovery</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Distinguish network faults from other kinds of failures [Familiarity]</li> <li>• Explain why synchronization constructs such as simple locks are not useful in the presence of distributed faults [Familiarity]</li> <li>• Write a program that performs any required marshalling and conversion into message units, such as packets, to communicate interesting data between two hosts [Usage]</li> <li>• Measure the observed throughput and response latency across hosts in a given network [Usage]</li> <li>• Explain why no distributed system can be simultaneously consistent, available, and partition tolerant [Familiarity]</li> <li>• Implement a simple server – for example, a spell checking service [Usage]</li> <li>• Explain the tradeoffs among overhead, scalability, and fault tolerance when choosing a stateful v stateless design for a given service [Familiarity]</li> <li>• Describe the scalability challenges associated with a service growing to accommodate many clients, as well as those associated with a service only transiently having many clients [Familiarity]</li> <li>• Give examples of problems for which consensus algorithms such as leader election are required [Usage]</li> </ul>
<b>Readings:</b> Coulouris et al. (2011)	

<b>UNIT 2: Cloud Computing (15)</b>	
<b>Competences: a,b</b>	
<b>Content</b>	<b>Generales Goals</b>
<ul style="list-style-type: none"> <li>• Visión global de <i>Cloud Computing</i>.</li> <li>• Historia.</li> <li>• Visión global de las tecnologías que envuelve.</li> <li>• Beneficios, riesgos y aspectos económicos.</li> <li>• Cloud services <ul style="list-style-type: none"> <li>– Infrastructure as a service <ul style="list-style-type: none"> <li>* Elasticity of resources</li> <li>* Platform APIs</li> </ul> </li> <li>– Software as a service</li> <li>– Security</li> <li>– Cost management</li> </ul> </li> <li>• Internet-Scale computing <ul style="list-style-type: none"> <li>– Task partitioning</li> <li>– Data access</li> <li>– Clusters, grids, and meshes</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• Explicar el concepto de Cloud Computing. [Familiarity]</li> <li>• Listar algunas tecnologías relacionadas con Cloud Computing. [Familiarity]</li> <li>• Explain strategies to synchronize a common view of shared data across a collection of devices [Familiarity]</li> <li>• Discutir las ventajas y desventajas del paradigma de Cloud Computing. [Familiarity]</li> <li>• Expresar los beneficios económicos así como las características y riesgos del paradigma de Cloud para negocios y proveedores de cloud. [Familiarity]</li> <li>• Diferenciar entre los modelos de servicio. [Usage]</li> </ul>
<b>Readings:</b> Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013)	

<b>UNIT 3: Centros de Procesamiento de Datos (10)</b>	
<b>Competences: i</b>	
<b>Content</b>	<b>Generales Goals</b>
<ul style="list-style-type: none"> <li>• Visión global de un centro de procesamiento de datos.</li> <li>• Consideraciones en el diseño.</li> <li>• Comparación de actuales grandes centros de procesamiento de datos.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la evolución de los Data Centers. [Familiarity]</li> <li>• Esbozar la arquitectura de un data center en detalle. [Familiarity]</li> <li>• Indicar consideraciones de diseño y discutir su impacto. [Familiarity]</li> </ul>
<b>Readings:</b> Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013)	

UNIT 4: Cloud Computing (20)	
Competences: i,j	
Content	Generales Goals
<ul style="list-style-type: none"> <li>• Virtualization <ul style="list-style-type: none"> <li>– Shared resource management</li> <li>– Migration of processes</li> </ul> </li> <li>• Seguridad, recursos y aislamiento de fallas.</li> <li>• Almacenamiento como servicio.</li> <li>• Elasticidad.</li> <li>• Xen y Wmware.</li> <li>• Amazon EC2.</li> </ul>	<ul style="list-style-type: none"> <li>• Virtualization <ul style="list-style-type: none"> <li>– Shared resource management</li> <li>– Migration of processes</li> </ul> </li> <li>. [Familiarity]</li> <li>• Explain the advantages and disadvantages of using virtualized infrastructure. [Familiarity]</li> <li>• Identificar las razones por qué la virtualización está llegando a ser enormemente útil, especialmente en la cloud. [Familiarity]</li> <li>• Explicar diferentes tipos de aislamiento como falla, recursos y seguridad proporcionados por la virtualización y utilizado por la cloud. [Familiarity]</li> <li>• Explicar la complejidad que puede tener el administrar en términos de niveles de abstracción y interfaces bien definidas y su aplicabilidad para la virtualización en la cloud. [Familiarity]</li> <li>• Definir virtualización y identificar diferentes tipos de máquinas virtuales. [Familiarity]</li> <li>• Identificar condiciones de virtualización de CPU, reconocer la diferencia entre <i>full virtualization</i> y <i>paravirtualization</i>, explicar emulación como mayor técnica para virtualización del CPU y examinar planificación virtual del CPU en Xen. [Familiarity]</li> <li>• Esbozar la diferencia entre la clásica memoria virtual del SO y la virtualización de memoria. Explicar los múltiples niveles de mapeamiento de páginas en oposición a la virtualización de la memoria. Definir memoria <i>over-commitment</i> e ilustrar sobre Wmware <i>memory ballooning</i> como técnica de reclamo para sistemas virtualizados con memoria <i>over-committed</i>. [Familiarity]</li> </ul>
<b>Readings:</b> Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013)	

UNIT 5: Cloud Computing (12)	
Competences: i,j	
Content	Generales Goals
<ul style="list-style-type: none"> <li>• Cloud-based data storage               <ul style="list-style-type: none"> <li>– Shared access to weakly consistent data stores</li> <li>– Data synchronization</li> <li>– Data partitioning</li> <li>– Distributed file systems</li> <li>– Replication</li> </ul> </li> <li>• Visión global sobre tecnologías de almacenamiento.</li> <li>• Conceptos fundamentales sobre almacenamiento en la cloud.</li> <li>• Amazon S3 y EBS.</li> <li>• Sistema de archivos distribuidos.</li> <li>• Sistema de bases de datos NoSQL.</li> </ul>	<ul style="list-style-type: none"> <li>• Describir la organización general de datos y almacenamiento. [Familiarity]</li> <li>• Identificar los problemas de escalabilidad y administración de la big data. Discutir varias abstracciones en almacenamiento. [Familiarity]</li> <li>• Comparar y contrastar diferentes tipos de sistema de archivos. Comparar y contrastar el Sistema de Archivos Distribuido de Hadoop (HDFS) y el Sistema de Archivos Paralelo Virtual (PVFS). [Usage]</li> <li>• Comparar y contrastar diferentes tipos de bases de datos. Discutir las ventajas y desventajas sobre las bases de datos NoSQL. [Usage]</li> <li>• Discutir los conceptos de almacenamiento en la cloud. [Familiarity]</li> </ul>
<b>Readings:</b> Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013)	

UNIT 6: Modelos de Programación (12)	
Competences: j	
Content	Generales Goals
<ul style="list-style-type: none"> <li>• Visión global de los modelos de programación basados en cloud computing.</li> <li>• Modelo de Programación MapReduce.</li> <li>• Modelo de programación para aplicaciones basadas en Grafos.</li> </ul>	<ul style="list-style-type: none"> <li>• Explicar los aspectos fundamentales de los modelos de programación paralela y distribuida. [Familiarity]</li> <li>• Diferencias entre los modelos de programación: MapReduce, Pregel, GraphLab y Giraph. [Usage]</li> <li>• Explicar los principales conceptos en el modelo de programación MapReduce. [Usage]</li> </ul>
<b>Readings:</b> Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013), Low et al. (2012), Malewicz et al. (2010), Baluja et al. (2008)	

8. Methodology
<p>El profesor del curso presentará clases teóricas de los temas señalados en el programa propiciando la intervención de los alumnos.</p> <p>El profesor del curso presentará demostraciones para fundamentar clases teóricas.</p> <p>El profesor y los alumnos realizarán prácticas</p> <p>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.</p>

## 9. Assessment

**Continuous Assessment 1** : 20 %

**Partial Exam** : 30 %

**Continuous Assessment 2** : 20 %

**Final exam** : 30 %

## References

- Baluja, Shumeet et al. (2008). “Video Suggestion and Discovery for Youtube: Taking Random Walks Through the View Graph”. In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. ACM: Beijing, China, pp. 895–904. ISBN: 978-1-60558-085-2. DOI: 10.1145/1367497.1367618.
- Buyya, Rajkumar, Christian Vecchiola, and S. Thamarai Selvi (2013). *Mastering Cloud Computing: Foundations and Applications Programming*. 1st. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA. ISBN: 9780124095397, 9780124114548.
- Coulouris, George et al. (2011). *Distributed Systems: Concepts and Design*. 5th. Addison-Wesley Publishing Company: USA. ISBN: 0132143011, 9780132143011.
- Hwang, Kai, Jack Dongarra, and Geoffrey C. Fox (2011). *Distributed and Cloud Computing: From Parallel Processing to the Internet of Things*. 1st. Morgan Kaufmann Publishers Inc.: San Francisco, CA, USA. ISBN: 0123858801, 9780123858801.
- Low, Yucheng et al. (Apr. 2012). “Distributed GraphLab: A Framework for Machine Learning and Data Mining in the Cloud”. In: *Proc. VLDB Endow.* 5(8), pp. 716–727. ISSN: 2150-8097. DOI: 10.14778/2212351.2212354.
- Malewicz, Grzegorz et al. (2010). “Pregel: A System for Large-scale Graph Processing”. In: *Proc. ACM SIGMOD*. SIGMOD '10, pp. 135–146. DOI: 10.1145/1807167.1807184.