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

CS3P2. Cloud Computing (Mandatory)

2023-II

1. General information		
1.1 School	:	Ciencia de la Computación
1.2 Course	:	CS3P2. Cloud Computing
1.3 Semester	:	10^{mo} Semestre.
1.4 Prerrequisites	:	CS370. Big Data. (9^{th} Sem)
1.5 Type of course	:	Mandatory
1.6 Learning modality	:	Face to face
1.7 Horas	:	1 HT; 4 HP;
1.8 Credits	:	3
1.9 Plan	:	Plan Curricular 2016

2. Professors

Lecturer

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

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:

1) Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions. (**Usage**)

6) Apply computer science theory and software development fundamentals to produce computing-based solutions. (**Usage**)

7. Content

UNIT 1: Distributed Systems (15)

Competences:

Content	**Generales Goals**
• Faults (cross-reference OS/Fault Tolerance) – Network-based (including partitions) and node-based failures – Impact on system-wide guarantees (e.g., availability) • Distributed message sending – Data conversion and transmission – Sockets – Message sequencing – Buffering, retrying, and dropping messages • Distributed system design tradeoffs – Latency versus throughput – Consistency, availability, partition tolerance • Distributed service design – Stateful versus stateless protocols and services – Session (connection-based) designs – Reactive (IO-triggered) and multithreaded designs • Core distributed algorithms – Election, discovery	• Distinguish network faults from other kinds of failures [Familiarity] • Explain why synchronization constructs such as simple locks are not useful in the presence of distributed faults [Familiarity] • Write a program that performs any required marshalling and conversion into message units, such as packets, to communicate interesting data between two hosts [Usage] • Measure the observed throughput and response latency across hosts in a given network [Usage] • Explain why no distributed system can be simultaneously consistent, available, and partition tolerant [Familiarity] • Implement a simple server – for example, a spell checking service [Usage] • Explain the tradeoffs among overhead, scalability, and fault tolerance when choosing a stateful v stateless design for a given service [Familiarity] • 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] • Give examples of problems for which consensus algorithms such as leader election are required [Usage]

Readings: Coulouris et al. (2011)

UNIT 2: Cloud Computing (15)

Competences:

Content	**Generales Goals**

• Visión global de *Cloud Computing*.
• Historia.
• Visión global de las tecnologias que envuelve.
• Beneficios, riesgos y aspectos económicos.
• Cloud services
  – Infrastructure as a service
  ∗ Elasticity of resources
  ∗ Platform APIs
  – Software as a service
  – Security
  – Cost management
• Internet-Scale computing
  – Task partitioning
  – Data access
  – Clusters, grids, and meshes
	• Explicar el concepto de Cloud Computing. [Familiarity]
	• Listar algunas tecnologias relacionadas con Cloud Computing. [Familiarity]
	• Explain strategies to synchronize a common view of shared data across a collection of devices [Familiarity]
	• Discutir las ventajas y desventajas del paradigma de Cloud Computing. [Familiarity]
	• Expresar los beneficios económicos así como las carácteristicas y riesgos del paradigma de Cloud para negocios y proveedores de cloud. [Familiarity]
	• Diferenciar entre los modelos de servicio. [Usage]

Readings: Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013)

UNIT 3: Centros de Procesamiento de Datos (10)

Competences:

Content	**Generales Goals**
• Visión global de un centro de procesamiento de datos. • Consideraciones en el diseño. • Comparación de actuales grandes centros de procesamiento de datos.	• Describir la evolución de los Data Centers. [Familiarity] • Esbozar la arquitectura de un data center en detalle. [Familiarity] • Indicar consideraciones de diseño y discutir su impacto. [Familiarity]

Readings: Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013)

UNIT 4: Cloud Computing (20)

Competences:

Content	**Generales Goals**

• Virtualization
  – Shared resource management
  – Migration of processes
• Seguridad, recursos y isolamiento de fallas.
• Almacenamiento como servicio.
• Elasticidad.
• Xen y WMware.
• Amazon EC2.
	• Virtualization
	  – Shared resource management
	  – Migration of processes
	. [Familiarity]
	• Explain the advantages and disadvantages of using virtualized infrastructure. [Familiarity]
	• Identificar las razones por qué la virtualización está llegando a ser enormente útil, especialmente en la cloud. [Familiarity]
	• Explicar diferentes tipos de isolamiento como falla, recursos y seguridad proporcionados por la virtualización y utilizado por la cloud. [Familiarity]
	• 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]
	• Definir virtualización y identificar diferentes tipos de máquinas virtuales. [Familiarity]
	• Identificar condiciones de virtualización de CPU, reconocer la diferencia entre *full virtualization* y *paravirtualization*, explicar emulación como mayor técnica para virtualización del CPU y examinar planificación virtual del CPU en Xen. [Familiarity]
	• Esbozar la diferencia entre la clásica memoria virtual del SO y la virtualización de memoria. Explicar los múltiplos niveles de mapeamiento de páginas en oposición a la virtualización de la memoria. Definir memoria *over-commitment* e ilustrar sobre WMware *memory ballooning* como técnica de reclamo para sistemas virtualizados con memoria *over-committed*. [Familiarity]

Readings: Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013)

UNIT 5: Cloud Computing (12)	
Competences:	
Content	**Generales Goals**
• Cloud-based data storage – Shared access to weakly consistent data stores – Data synchronization – Data partitioning – Distributed file systems – Replication • Visión global sobre tecnologías de almacenamiento. • Conceptos fundamentales sobre almacenamiento en la cloud. • Amazon S3 y EBS. • Sistema de archivos distribuidos. • Sistema de bases de datos NoSQL.	• Describir la organización general de datos y almacenamiento. [Familiarity] • Identificar los problemas de escalabilidad y administración de la big data. Discutir varias abstracciones en almacenamiento. [Familiarity] • 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] • Comparar y contrastar diferentes tipos de bases de datos. Discutir las ventajas y desventajas sobre las bases de datos NoSQL. [Usage] • Discutir los conceptos de almacenamiento en la cloud. [Familiarity]
Readings: Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013)	

UNIT 6: Modelos de Programación (12)	
Competences:	
Content	**Generales Goals**
• Visión global de los modelso de programación basados en cloud computing. • Modelo de Programación MapReduce. • Modelo de programación para aplicaciones basadas en Grafos.	• Explicar los aspectos fundamentales de los modelos de programación paralela y distribuida. [Familiarity] • Diferencias entre los modelos de programación: MapReduce, Pregel, GraphLab y Giraph. [Usage] • Explicar los principales conceptos en el modelo de programación MapReduce. [Usage]
Readings: Hwang, Dongarra, and Fox (2011), Buyya, Vecchiola, and Selvi (2013), Low et al. (2012), Malewicz et al. (2010), Baluja et al. (2008)	

8. Methodology

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. Assessment Theory Sessions:
The theory sessions are held in master classes with activities including active learning and roleplay to allow students to internalize the concepts.

Practical Sessions:
The practical sessions are held in class where a series of exercises and/or practical concepts are developed through problem solving, problem solving, specific exercises and/or in application contexts.

Evaluation System:
The final grade is obtained through of:

CONTINUOUS ASSESMENT	EVALUATIONS
Continuous assessment 1 : 24 % **Continuous assessment 2** : 36 %	**Midterm Exam** : 20 % **Final Exam** : 20 %
60%	40%

Where:

Continuous Assessment: It includes group work, active participation in class, exercise test.

- Continuos assessment 1 (weeks 1 - 9)
- Continuos assesment 2 (weeks 10 - 17)

To pass the course you must obtain 11.5 or more in the final grade .

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.