

Asignatura: OPC13 – Cloud Computing

Ensayo de resultados de aprendizaje de la **semana 5**

Temas: Cloud Storage, Algorithms, Program Design, Getting started with compute

Integrante:

José Angel Quispe Meza
Matrícula: 390007
a390007@uach.m

1. Resumen Tema “Cloud Storage”

El curso de Cloud Storage aborda cómo almacenar y gestionar grandes volúmenes de datos en la nube, enfatizando conceptos clave como disponibilidad, redundancia y regiones. La disponibilidad se refiere al tiempo en que los datos están accesibles en línea; una conexión estable y proveedores confiables aseguran un acceso continuo. La redundancia implica almacenar copias de datos en múltiples ubicaciones, garantizando que, ante cualquier falla, los datos estén siempre seguros y disponibles. Al distribuir los datos en varias regiones, los usuarios pueden acceder a ellos desde la ubicación más cercana, optimizando la velocidad y reduciendo latencias.

AWS ofrece diversas opciones de almacenamiento adaptadas a diferentes necesidades empresariales: Amazon S3 para almacenamiento de objetos, ideal para datos no estructurados; Amazon EBS para almacenamiento en bloque; y Amazon EFS para sistemas de archivos compartidos. Amazon S3 es especialmente robusto en términos de durabilidad y disponibilidad, ya que replica los datos en múltiples ubicaciones. Además, AWS incorpora encriptación de datos y políticas de acceso para garantizar la seguridad y el cumplimiento de normativas. El almacenamiento en la nube es fundamental para empresas, científicos, y usuarios en general, facilitando desde el almacenamiento de registros financieros hasta la colaboración en proyectos científicos y el uso diario de redes sociales y streaming. Organizaciones como la NASA, que emplea AWS para centralizar sus archivos y recursos multimedia, son un claro ejemplo de cómo la nube optimiza el acceso y manejo de información crucial en diversos sectores.

2. Resumen Tema “Algorithms”

El módulo de Algoritmos se enfocó en los principios fundamentales que permiten resolver problemas de manera eficiente mediante el diseño de algoritmos efectivos. Un algoritmo es esencialmente una serie de pasos que permiten llevar a cabo una tarea específica, y su diseño es crucial para la velocidad y la eficiencia de los programas. En este curso, aprendí sobre la importancia de seleccionar los algoritmos correctos según el tipo de problema que se quiere resolver. Por ejemplo, existen algoritmos especializados para la búsqueda y clasificación de datos, dos operaciones comunes en

sistemas que manejan grandes volúmenes de información. El curso también destacó la complejidad computacional y cómo esta afecta el rendimiento de los programas. Comprender la diferencia entre complejidad temporal (el tiempo que toma ejecutar el algoritmo) y complejidad espacial (la cantidad de memoria que requiere) me ayudó a ver cómo los recursos en la nube se pueden utilizar de forma más eficiente. Este conocimiento es esencial en la programación moderna, especialmente en un entorno de nube donde los recursos son limitados y cada operación tiene un costo. Los algoritmos son la base de todas las aplicaciones que interactúan con datos, y aprender sobre ellos me ha dado las herramientas para diseñar soluciones más rápidas y eficaces.

3. Resumen Tema “Program Design”

El curso de Program Design me permitió comprender los pasos clave para desarrollar aplicaciones bien estructuradas, eficientes y fáciles de mantener. La fase de diseño es fundamental en el ciclo de vida del software, ya que permite planificar cómo se organizará el programa, definir sus funciones y establecer las interacciones entre sus componentes. Aprendí sobre técnicas de planificación, como el uso de diagramas de flujo y pseudocódigo, que facilitan visualizar la lógica antes de la codificación.

Un aspecto central fue la modularidad, que consiste en dividir el programa en módulos, cada uno con una responsabilidad específica. Este enfoque no solo hace que el código sea más fácil de probar y depurar, sino que también mejora su escalabilidad, un beneficio clave para aplicaciones en la nube que manejan grandes volúmenes de datos y múltiples solicitudes. Otros conceptos importantes incluyen variables, que representan factores con valores diversos dentro del programa, y las estructuras de control, que orientan al programa sobre cómo gestionar la información. Las estructuras de control, como los bucles e instrucciones condicionales ("si... entonces..."), permiten al programa tomar decisiones. Además, los algoritmos son procesos que transforman los datos proporcionados por el usuario y pueden incluir varias funciones y bucles para verificar condiciones específicas. Estos elementos son esenciales para crear aplicaciones organizadas y escalables, especialmente en entornos de computación en la nube.

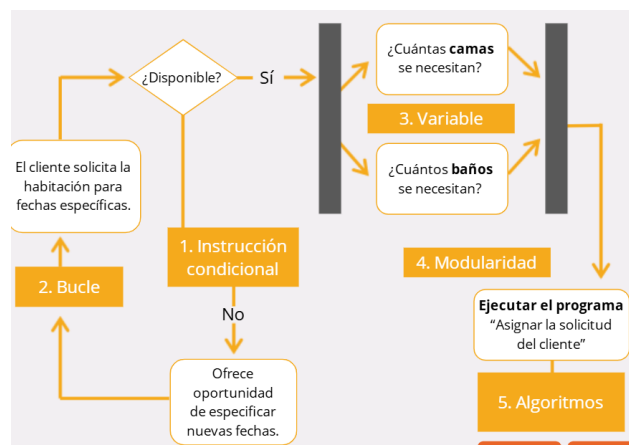


Figura del diagrama del servicio de reservas en un hospedaje

4. Resumen Tema “Getting started with compute”

Finalmente, el curso Getting Started with Compute se enfocó en el uso práctico de la computación en la nube. En este laboratorio, tuvimos la oportunidad de explorar diferentes tipos de instancias de cómputo en AWS y cómo pueden configurarse para satisfacer diversas necesidades empresariales. Las instancias de cómputo son máquinas virtuales que proporcionan el poder computacional necesario para ejecutar aplicaciones y procesar datos en la nube.

AWS ofrece una variedad de instancias, desde las de propósito general hasta aquellas optimizadas para cargas de trabajo específicas, como procesamiento intensivo de CPU o GPU. En el laboratorio, aprendí cómo lanzar una instancia, conectarme a ella y ejecutar aplicaciones básicas. Este ejercicio práctico me permitió experimentar de primera mano cómo la computación en la nube facilita el acceso a recursos de alta capacidad sin la necesidad de invertir en infraestructura física. La nube permite que cualquier usuario tenga acceso a un servidor en cuestión de minutos, lo que acelera enormemente el desarrollo de aplicaciones y la experimentación con nuevos proyectos.

Además, el curso resaltó la importancia de la elasticidad de la nube, que permite ajustar los recursos según la demanda. Por ejemplo, en una aplicación de comercio electrónico, las instancias pueden escalar hacia arriba en momentos de alta demanda, como durante una promoción, y reducirse durante las horas de menos tráfico. Este enfoque no solo optimiza los recursos, sino que también ayuda a reducir costos, ya que las empresas solo pagan por lo que realmente utilizan.