

Asignatura: OPC13 – Cloud Computing

Ensayo de resultados de aprendizaje de la **semana 11**

Temas: Getting started with storage, Getting started with serverless

Integrantes:

Adrian Caleb Jaramillo Flores
Matrícula: 367857
a367857@uach.mx

Ana Rebeca Moreno Reza
Matrícula: 367783
a367783@uach.mx

Abel González Mireles
Matrícula: 361031
a361031@uach.mx

Miguel David Rodríguez González.
Matrícula: 343786
a343786@uach.mx

Docente:

David Maloof Flores
Matrícula: 197589
a197589@uach.mx

1. Resumen Tema “Getting started with storage”

Se puede almacenar información de manera tradicional en servidores manejados en las mismas instalaciones o se puede guardar en la nube, usando los servicios de alguna compañía como AWS; lo cual proporciona diferentes ventajas: **rentable** (pagar por lo que se usa), **seguro** (se guarda en múltiples servidores), **accesible** (permite acceso a múltiples usuarios), **escalable** (cambia según la carga de trabajo), **administrado** (no tienes que preocuparte de servidores), **respaldado** (se guarda información en diferentes localizaciones por si alguno falla).

Así mismo existen tres tipos de almacenamiento de la información en la nube, se habla de **almacenamiento de bloque** (son de tamaño fijo, se almacenan en un dispositivo, las apps y sistemas fijan el cómo se accede, combinan o modifican los bloques, cada bloque puede almacenarse en diferentes sistemas y configurarse para distintos SO), de **archivos** (metodología para acceder a datos en un sistema de archivos compartidos, su estructura es jerárquica) y de **objetos** (se almacenan como objetos, cada objeto se compone de datos, metadatos [tamaño del objeto, propósito, ...] y una clave [identificador único del objeto], cuando se actualiza la información se actualiza todo el objeto).

2. Resumen Tema “Getting started with serverless”

Los **microservicios** tienen un enfoque arquitectónico y organizativo para el desarrollo de software. El software se compone de pequeños servicios independientes los cuales se comunican mediante API 's bien definidas. Son propiedad de pequeños equipos autónomos.

Acoplamiento ajustado: En una aplicación monolítica, si fallara una pieza, la aplicación completa colapsaría. Si hay un pico de demanda, toda la arquitectura debe escalar. Agregar funciones a una aplicación monolítica se vuelve más complejo a medida que pasa el tiempo. Las partes de la base de código deben trabajar entre sí para sincronizarse correctamente.

Desventajas:

- Todos los procesos tienen un acoplamiento ajustado y se ejecutan como un solo servicio.
- Mientras más crece la base de código se vuelve más complejo agregar o mejorar funciones.
- Limita el crecimiento y aumento de un único error del proceso

Acoplamiento débil: En una arquitectura de microservicios, una aplicación se crea con componentes independientes que ejecutan cada proceso de la aplicación como un servicio. Estos servicios se comunican a través de una interfaz bien definida mediante API ligeras. Debido a que se ejecutan de forma independiente, cada servicio se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación. Si un componente falla, los otros componentes permanecen activos.

Beneficios de los microservicios:

- **Usan API públicas** - Hacerlo es la única forma de obtener los beneficios de los microservicios.
- **Usan la herramienta adecuada para el trabajo** - Cada servicio se crea para una funcionalidad explícita.
- **Aseguran sus servicios** - Están diseñados para brindar seguridad en cada paso.
- **Ser un buen ciudadano** - Como propietario de un servicio, debe comprender como le llaman otros servicios.
- **Transformación de la empresa** - El cambio requiere una transformación en la organización y los mecanismos de comunicación entre los equipos.
- **Automatizar todo** - Se recomienda automatizar la mayor parte posible de la canalización de entrega.

El cómputo sin servidor puede crear y ejecutar aplicaciones y servicios sin pensar en los servidores, por lo que no tiene que aprovisionar, escalar o administrar servidor alguno. Además, administra todo lo necesario para ejecutar y escalar la aplicación con alta disponibilidad.

Ventajas:

- Los desarrolladores pueden centrarse en el producto principal.
- Se encarga de administrar y revisar el funcionamiento de los servidores o los tiempos de ejecución.

La arquitectura dirigidas por eventos usa eventos para iniciar acciones y comunicación entre servicios desacoplados, los eventos son el mecanismo principal para compartir la información entre los servicios y publica información para que otros servicios la consuman.

Ventajas:

- **Escalado y fallas independientes** - Sus servicios son interoperables, pero sin un servicio presenta un error, el resto seguirá ejecutándose.
- **Facilidad de auditoría** - Actúa como una ubicación central para auditar su aplicación y definir las políticas.
- **Desarrollo con agilidad** - Elimina la necesidad de una fuente de coordinación entre los servicios del productor y el consumidor, lo que agiliza sus procesos de desarrollo.
- **Reducción de costos** - Suceden bajo demanda a medida que el evento se presenta en el enrutador.

Las tecnologías sin servidor son un servicio que se utiliza para construir sus arquitecturas, también son las herramientas que utiliza para construir esas secciones de su arquitectura.

Ventajas:

- **Sin administración de servidores.**
- **Servicios de pago por valor** - La utilización de recursos se optimiza automáticamente y nunca paga por aprovisionamiento excesivo.
- **Escalado continuo** - Se pueden configurar para escalar automáticamente de cero a demandas máximas para que pueda adaptarse a las necesidades de los clientes más rápido que nunca.
- **Alta disponibilidad** - Se tiene integrada de forma predeterminada.

Sabes

@pataconposting

MrPato

**Sabes que vas a comer lo mismo
por 4 días cuando sacan esta olla**

Nadie:

Tutoriales de cocina:

