MODULO 09 - EJERCICIO 01-A

ALEXIS YURI M.

Completar un cuadro comparativo entre los 4 modelos de servicio en la nube. Deberán usar ejemplos reales (por ejemplo servicios de AWS, Azure o Google Cloud) y explicar las ventajas o desafíos de cada uno.

1. Definición de cada modelo:

- IaaS (Infraestructura como Servicio): Es un modelo que proporciona recursos de infraestructura TI virtualizados, como servidores, redes y almacenamiento, a los cuales se accede a través de internet.

- PaaS (Plataforma como Servicio): Ofrece un entorno completo para desarrollar, ejecutar y gestionar aplicaciones, abstraído de la infraestructura subyacente.

- SaaS (Software como Servicio): Es un modelo en el que las aplicaciones listas para usar se entregan a través de internet, sin requerir instalación o mantenimiento local.

- FaaS (Función como Servicio): También conocido como serverless, permite ejecutar piezas de código o funciones en respuesta a eventos, sin necesidad de gestionar servidores.

2. Cuadro Comparativo de Modelos Cloud.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Modelo** | **Definición** | **Proveedor y Ejemplos** | **¿Quién gestiona? (Cliente)** | **¿Quién gestiona? (Proveedor)** |  |
|  | IaaS | Proporciona infraestructura de TI virtualizada. | AWS. Ejemplos: Amazon EC2 y Amazon S3 | Cliente: Sistema operativo, aplicaciones, datos | Proveedor: Red, hardware, almacenamiento |  |
|  | PaaS | Ofrece un entorno para desarrollar y ejecutar aplicaciones. | Heroku. Ejemplos: Azure App Service , Google App Engine | Cliente: Aplicaciones y datos | Proveedor: Todo lo demás (infraestructura, SO, entorno de ejecución) |  |
|  | SaaS | Proporciona aplicaciones listas para usar. | Microsoft. Ejemplos: Google Workspace, Microsoft 365, Dropbox | Cliente: Solo usa el software | Proveedor: Todo (infraestructura, plataforma y aplicación) |  |
|  | FaaS | Ejecuta funciones en respuesta a eventos. | AWS. Ejemplos: AWS Lambda, Azure Functions | Cliente: Carga el código y define eventos | Proveedor: Ejecuta la función cuando es invocada |  |
|  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Modelo** | **2 Ventajas Clave** | **1 Desafio Clave** |  |
|  | IaaS | 1. Control total sobre la infraestructura. 2. Flexibilidad y escalabilidad. | El cliente debe gestionar el sistema operativo y las aplicaciones. |  |
|  | PaaS | 1. Permite enfocarse en el código. 2. Despliegue ágil de aplicaciones. | Menor control sobre la infraestructura y el entorno de ejecución. |  |
|  | SaaS | 1. No requiere instalación ni mantenimiento local. 2. Acceso desde cualquier lugar. | Poco o nulo control sobre la aplicación o la infraestructura. |  |
|  | FaaS | 1. Altamente escalable y orientado a tareas puntuales. 2. Pago por uso, solo por el tiempo de ejecución. | Dificultad en la gestión de la complejidad a medida que el número de funciones aumenta. |  |
|  |  |  |  |  |