**A.1. ¿Qué es AWS Lambda y cuál es su propósito principal en la arquitectura sin servidor?**

Aws Lambda es un servicio serveless creado por Amazon web Services. Su función es ejecutar el backend de una aplicación sin tener que crear el servidor ya que lambda se encarga de hacer todo el trabajo. Esto es beneficioso para el desarrollador ya que no se encarga de nada más que mantener el código.

**A.2. Describe la arquitectura de una función Lambda. ¿Qué eventos pueden desencadenar una función Lambda?**

Lambda es un servicio que se ejecuta en respuesta de eventos y ésta se encarga de crear la infraestructura necesaria para ejecutarse de acuerdo a la demanda del evento.

Lambda está compuesta por:

-Código de la función: El código proporcionado por el programador debe contener la lógica que debe hacer la función al momento de ejecutarse. Este puede ser Python, JavaScript, c#, go.

-Paquete de Implementación: El código debe estar comprimido en un archivo zip para poder subirlo a Amazon.

-Configuración de la Función: Se debe configurar el nombre de la función, la cantidad de memoria asignada, el tiempo de espera y las variables de entorno.

-Eventos: Los eventos son desencadenadores que activan la función Lambda. Éstos pueden ser fuentes externas a Amazon web services como recursos de éste. Algunos recursos son Dynamo, Api Gateway, S3, Cognit, Step Functions, entre otros.

-Gestión de Invocación: Aws administra los recursos en función en respuesta a eventos. Puede escalar para manejar múltiples peticiones y así asegurar que se entregue respuesta a cada uno de los eventos.

- Ejecución de la Función: Cuando se activa una función Lambda, se ejecuta el código contenido en el paquete de implementación.

- Registro y Supervisión: AWS Lambda se integra con Amazon CloudWatch para la supervisión y el registro de métricas y registros de las funciones Lambda.

- Resultados y Respuesta: La función Lambda puede generar resultados que pueden enviarse a otros servicios de AWS o devolverse al origen del evento

- Escalabilidad: AWS Lambda proporciona escalabilidad automática. Puede ajustar la cantidad de memoria y la configuración de tiempo de espera para adaptarse a las necesidades de rendimiento de su aplicación.

-Control de Acceso y Seguridad: Puede definir políticas de permisos de IAM para controlar quién puede invocar su función y qué recursos pueden acceder. También puede utilizar variables de entorno para almacenar información sensible, como credenciales de API.

**A.3. ¿Qué es GraphQL y cómo se diferencia de REST? ¿Cuándo podría ser beneficioso utilizar GraphQL en lugar de REST?**

GraphQL es un lenguaje de consulta para APIs y un entorno de ejecución que permite a los clientes solicitar exactamente los datos que necesitan, Una de las diferencias más grandes es que GraphQL es más flexible que Rest ya que puede agrupar consultas en una sola y así evitar que colapse.

Es más beneficioso que Rest ya que al ser más flexible puede adaptarse a la cantidad de datos que necesite el cliente y esto lo hace más eficiente que Rest.

**A.4. ¿Cuáles son las diferencias clave entre una base de datos relacional y una base de datos NoSQL como DynamoDB?**

Las bases de datos relacionales como su nombre lo dice tienen relación entre las tablas, tienen un lenguaje estructurado para manipular la información llamado SQL. Por otro lado las no relacionales no tienen una estructura definida y esto la hace más flexible a la hora de manipular datos. Uno de los usos más comunes de las bases de datos no relacionales es el uso en microservicios y big data.

**A.6. ¿Cuál es el propósito principal de Amazon API Gateway en AWS y cómo se relaciona con Lambda?**

Api Gateway es un gestor de tráfico entre el backend y el cliente lo que permite el control de acceso a éste aplicando políticas y autenticación. También es el encargado de enrutar la solicitud al servicio adecuado y entregarle respuesta a este.

Api Gateway se encarga de gestionar la información del backend y brindarle mayor seguridad a lambda.

**A.7. Explique qué es AWS AppSync y en qué tipos de aplicaciones es más útil su uso.**

Es un servicio que permite la creación de Apis de manera rápida y eficiente. Esta se encarga de manipular la información a través de GraphQL. Es recomendable utilizar AppSync cuando tiene más de un origen de datos ya que manipula la información para entregar sólo la información que se necesita.

Puede ser utilizado en una app tipo red social y aplicaciones que utilicen muchas fuentes de datos como big data, bases de datos y otras apis.

**A.10. Explique el concepto de escalado automático en DynamoDB y por qué es importante en aplicaciones con alta demanda**

Al ser una aplicación que no tiene servidor se adapta a las necesidades al momento de ser requerido. Esto es importante para las aplicaciones de alta demanda porque se va adaptando de acuerdo a la cantidad de peticiones para que cada petición pueda ser respondida de forma eficaz y eficiente.