



**Nombre completo:**

**Correo corporativo: No. de empleado:**

**Empresa: Área:**

**Objetivo**

Consta de 4 secciones en donde se preguntarán temas relacionados con la Arquitectura REST y estándares HTTP, estrategia de diseño y conocimientos de la plataforma Apigee.

*Recomendaciones:*



*Tómate el tiempo necesario*



*Ten confianza en tus posibilidades y en el trabajo/estudio bien realizado. ¡Puedes hacerlo!*



*Comienza por aquello que estés más seguro*



*Revísalo antes de entregar*

**Sección I - Arquitectura REST**

1. Defina API.- Una API es un acrónimo para interfaz de programación de aplicaciones. Se entiende como un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar aplicaciones,permitiendo la comunicación entre ellas. Pueden ser de dos tipos: APIs de dominio y APIs de Producto
2. Defina REST.- REST es es un tipo de arquitectura de desarrollo web basado en el protocolo HTTP para mejorar las comunicaciones cliente-servidor, define y proporciona elementos para obtener y generar datos y operaciones, devolviendo esos datos en formatos específicos, como XML y JSON. Se asocian al concepto de CRUD, a través de métodos http correspondientes: post, get, put, delete
3. Explique el concepto servicios web RESTful.- Los servicios web RESTFUL son programas basados en REST.
4. Explique que es un recurso en REST.- Un recurso REST se entiende como una entidad que representa un concepto del negocio y que puede ser accedido públicamente y son expuestos a través de los servicios RESTful.
5. ¿Cuál es el protocolo utilizado en REST?
6. HTTP
7. JMX
8. XML
9. SOAP
10. ¿Cuál es la forma más popular de representar un recurso en REST?
11. XML
12. JSON
13. Ambos
14. ¿Qué significa JSON?
15. JavaScript ObjectNumber
16. JavaScript ObjectNotation
17. JavaScript ObjectNotification
18. JavaScript ObjectNaming
19. ¿Cuáles son los lenguajes de marcado disponibles en los servicios REST?
20. HTML & VBSCRIPT
21. XML & Java
22. JSON & XML
23. HTML & XML
24. ¿A qué categoría pertenecen los códigos HTTP 1XX?
25. Redirection
26. Client Error
27. Informational
28. Success
29. ¿A qué categoría pertenecen los códigos HTTP 2XX?
30. Client Error
31. Redirection
32. Success
33. Server Error
34. ¿A qué categoría pertenecen los códigos HTTP 3XX?
35. Redirection
36. Client Error
37. Success
38. Server Error
39. ¿A qué categoría pertenecen los códigos HTTP 4XX?
40. Client Error
41. Informational
42. Server Error
43. Success
44. ¿A qué categoría pertenecen los códigos HTTP 5XX?
45. Server Error
46. Redirection
47. Informational
48. Client Error
49. ¿Qué indica el código HTTP 200?
50. Success
51. Completed
52. Warning
53. Error
54. ¿Cuál es el código HTTP relacionado a página no encontrada?
55. 402
56. 404
57. 403
58. 401
59. ¿Cuál es el código HTTP relacionado con el error interno del servidor?
60. 502
61. 501
62. 500
63. 504
64. ¿Cuáles son los diferentes métodos HTTP disponibles?
65. GET,SET,DELETE,MARK
66. GET,HEAD,CONNECT,OPTIONS
67. GET,PUT,REMOVE,POST
68. GET,SAVE,PUT,POST
69. ¿Cuál de los siguientes componentes de la solicitud HTTP se utiliza para identificar el recurso en el servidor?
70. VERB
71. URI
72. HTTP Version
73. RequestHeader
74. Nombra los métodos HTTP más comúnmente usados ​​soportados por REST.- GET,POST,PUT,DELETE,HEAD, OPTIONS
75. ¿Hay alguna diferencia entre las operaciones PUT y POST? Explíquelo. Si, la diferencia radica en el uso de cada uno ya que el método PUT es utilizado para la actualización de un registro existente, mientras que POST se utiliza para la creación de un registro. Los códigos de respuesta son 200 (OK) y 201 (Created) correspondientemente.
76. ¿Qué es la URI? .- Es un componente de la solicitud HTTP y se utiliza para identificar de forma única el recurso en el servidor.
77. ¿Qué entiendes por Payload en los servicios web RESTFul? .- Es el conjunto de datos enviados o recibidos entre el cliente y el servidor a través de los servicios web RESTFul, usualmente en formato JSON.
78. ¿Cuál es el límite para un Payload en el método POST?.- No hay un límite establecido en el estándar para las peticiones POST, sin embargo el límite está definido por la tecnología utilizada. Para la herramienta apigee utilizada en el grupo el valor por default es de 10MB.

**Sección II - Estrategia de Diseño**

1. Defina Contract First.- El concepto de Contract First está orientado a la definición y diseño de un contrato para el servicio. Este contrato le indica al consumidor como se espera que sea la petición(request) y respuesta(response). Una vez que este ha sido establecido el proveedor de servicio puede trabajarlo apegándose a lo establecido en dicho contrato.

1. Defina Features.- Se entienden como pruebas automáticas cuyo propósito es el de servir como soporte para el proyecto. Contienen escenarios en un [formato de lenguaje](https://behave.readthedocs.io/en/stable/gherkin.html#chapter-gherkin) natural [que](https://behave.readthedocs.io/en/stable/gherkin.html#chapter-gherkin) describe una feature o parte de ella con ejemplos representativos de los resultados esperados. Son de texto sin formato codificados en UTF-8.
2. Defina Mock.- Podemos definir un mock como un objeto simulados cuyo objetivo es imitar el comportamiento de objetos reales de una forma controlada. En una API se usan para verificar que las llamadas a los servicio se realizan correctamente.
3. Defina API de dominio.- Una Api de dominio hace referencia a aquellas que no están relacionadas con otras APIs de dominio y se enfocan en el aspecto para el que fueron diseñadas
4. Defina API de producto.- Una Api de producto es aquella que utiliza otras API’s de dominio para cumplir con su funcionalidad

**Sección III - Plataforma de APIS Apigee**

1. ¿Qué es un API Proxy?.- Un Api proxy es un sistema intermediario que actúa como enrutador desde un único punto de entrada hacia un grupo de servicios de backend, esto da flexibilidad y protege a los consumidores del servicio, pues en caso de realizarse cambios, el endpoint puede dirigirse a un Mock que permita simular el comportamiento de la respuesta esperada antes de modificar el backend y cambiarse al back una vez finalizado el desarrollo de manera transparente.
2. ¿Qué es un ProxyEndpoint?.- Es una parte de un API Proxy en él se define la forma en que los clientes consumen sus API, se configuran políticas para hacer cumplir la seguridad, verificaciones de cuotas, control de acceso y limitación de velocidad, definición de la URL del API proxy. También determina si las aplicaciones acceden al API Proxy a través de HTTP o HTTPS.
3. ¿Qué es un TargetEndpoint?.- **:** Es la parte del API Proxy que define la forma en que el proxy de API interactúa con sus servicios de backend. Es aqui donde se configura el reenvió de las solicitudes al servicio de backend adecuado, incluida la definición de cualquier configuración de seguridad, protocolo HTTP o HTTPS y otra información de conexión, además se pueden adjuntar *políticas* para asegurarse de que los mensajes de respuesta sean enviado con el formato adecuado para la aplicación que realizó la solicitud.
4. ¿Cuáles son los diferentes flows que existen en Apigee?.- Preflow, Postflow y Conditional Flow
5. ¿En qué flow definiría un esquema común para todos mis recursos APIS de validación de token de acceso?.- En el preflow, pues antes de realizar el enrutamiento primero se debería de validar que cumple con la autorización requerida
6. ¿Qué es una Política?.- Una política es un módulo que implementa una función de administración específica y limitada de fácil uso y sin la necesidad de escribir código, pues dicho de un modo más puntual pueden verse como APIs que ofrecen funcionalidades particulares en temas de seguridad, Administración del tráfico, Mediación y Extensión.

1. ¿Cuáles son las diferentes categorías de políticas que ofrece la plataforma?.- De Seguridad, Administración del tráfico, Mediación y Extensión.
2. Explique qué se puede hacer con la política Spikearrest.- Este tipo de política permite proteger los recursos expuestos de los ataques de denegación de servicio (DOS o DDOS) ya que esta política regula la cantidad de solicitudes procesadas por el API Proxy así como también las enviadas al backend.
3. Defina el concepto de Mediación.- Las políticas de mediación ofrecen la posibilidad de realizar transformaciones del formato del payload en una petición o respuesta entre el cliente y el recurso, también permiten aplicar ciertos tipos de validación de mensajes, dando posibilidad de generar mensajes de error.
4. Defina que es un Sharedflow.- .-Es un flujo de datos compartido, el cual permite reutilizar gran parte de los procesos contenidos en el api esto debido a que permite secuenciar el orden en que se ejecutarán las políticas.
5. ¿Cuál es la política más usada en apigee?.- La política SpikeArrest
6. ¿Qué recursos ofrece apigee para realizar la mediación y transformación?.- Las políticas de JSON TO XML, XML to JSON,Raisefault,XSL Transform,Open API Spec Validation,SOAP Message validation,AssignMessage, ExtractVariables, Access Entity, Key Value Map Operations
7. Defina qué es una Regla de Ruteo o RouteRule.- Este es un tipo de reglas que permite definir la ruta que seguirá una solicitud dada una o más condiciones, esta será enviada a un TargetEndpoint en el cual se define el tipo de servicio que se utilizará oara responder, puede ser una respuesta simulada (mock) o un servicio de backend..
8. Defina qué es un Target Server.- Un target server es un elemento que sirve como un balanceador de cargas y protección de los recursos pues se utilizan para enmascarar las url específicas de los servicios de backend, de manera que la petición al recurso siempre puede ser respondida sin saber cuál fue es el origen del servidor que respondió a la petición.
9. ¿Qué es un API Product?.- Es un conjunto de APIs, las cuales pueden estar distribuidos en diferentes grupos de desarrollo pero que se pueden acceder sin importar el dominio al que pertenezcan.
10. ¿Qué es un API Key?.- Una api key es un valor único que identifica a una App, este valor viaja encriptado y codificado en Base6. Para su creación es necesario definir una App la cual está definida como la relación entre el Developer y el API Product. En un esquema de autenticación OAuth 2.0 esta API key es la que se proporciona al realizar una solicitud de token enviándola como Basic Authentication

**Sección IV - Esquemas de Seguridad**

1. Defina OAuth 2.0.- Es un marco de autorización que permite a un usuario otorgar acceso a un sitio web o aplicación de terceros a los recursos protegidos de este, sin necesariamente revelar sus credenciales o incluso su identidad.
2. ¿Cuáles son los diferentes métodos de concesión que existen para OAuth 2.0?.- OAuth 2.0 define cuatro flujos para obtener un token de acceso:

* Authorization Code
* implicit
* Resource Owner Password Credentials
* Client Credentials

1. Explique cada método de concesión OAuth 2.0.-

**Client Credentials**.- El flujo funciona de la siguiente forma:

1. La aplicación cliente se autentica con el servidor de autorización Auth0 utilizando su API Key y su secret
2. El servidor de autorización Auth0 valida el EL API Key y el secret del cliente.
3. Si las credenciales son válidas entonces el servidor de autorización Auth0 responde con un token de acceso.
4. La aplicación cliente entonces puede usar el token de acceso obtenido para llamar a una API en su nombre.
5. La API responde con los datos solicitados.

**Implicit**.- Este flujo se recomienda solo utilizar para casos de uso de inicio de sesión, lo encontramos en los portales que nos solicitan autenticación de google, facebook, twitter, etc. Su uso es de la siguiente forma

1. El usuario hace clic en Iniciar sesión en la aplicación.
2. Se redirige al usuario al servidor de autorización de Auth0 pasando un parámetro **response\_type: id\_token** que indica el tipo de credencial solicitada, también transmite un parámetro **response\_mode: form\_post** con lo que se garantiza la seguridad.
3. El servidor de autorización Auth0 redirige al usuario a la solicitud de inicio de sesión y autorización.
4. El usuario se autentica utilizando una de las opciones de inicio de sesión configuradas y se le permite visualizar una página de consentimiento que enumera los permisos que Auth0 otorgará a la aplicación.
5. Su servidor de autorización Auth0 redirige al usuario a la aplicación con un token de identificación.

**Authorization Code.**- Es considerado como el tipo más seguro y su funcionamiento es el siguiente:

1. El usuario hace clic en Iniciar sesióndentro de la aplicación.
2. Se redirige al usuario al servidor de autorización de Auth0
3. El servidor de autorización Auth0 redirige al usuario a la solicitud de inicio de sesión y autorización.
4. El usuario se autentica utilizando una de las opciones de inicio de sesión configuradas y se le permite visualizar una página de consentimiento que enumera los permisos que Auth0 otorgará a la aplicación.
5. El servidor de autorización de Auth0 redirige al usuario a la aplicación con una autorización code, que es válida para un uso.
6. El SDK de Auth0 envía este code al servidor de autorización de Auth0 junto con el ID de cliente y el secreto del cliente de la aplicación.
7. El servidor de autorización Auth0 verifica el código, asi como tambien las credenciales del cliente (Id y Secret).
8. El servidor de autorización Auth0 responde con un token de identificación y un token de acceso y, opcionalmente, un token de actualización.
9. La aplicación cliente puede usar el token de acceso para llamar a una API para acceder a información sobre el usuario.
10. La API responde con los datos solicitados.

**Resource Owner Password Credentials**.-Este tipo de autenticación no es recomendado para aplicaciones de terceros y solo se recomienda utilizarlo en aplicaciones de confianza debido que solicita que los usuarios proporcionen credenciales (nombre de usuario y contraseña), normalmente mediante un formulario interactivo. Su funcionamiento se describe a continuación:

1. El usuario hace clic en **Iniciar sesión** dentro de la aplicación e ingresa sus credenciales.
2. La aplicación envía las credenciales del usuario al servidor de autorización Auth0
3. El servidor de autorización Auth0 valida las credenciales.
4. El servidor de autorización Auth0 responde con un token de acceso y, opcionalmente, un token de actualización.
5. La aplicación cliente puede usar el token de acceso para llamar a una API para acceder a información sobre el usuario.
6. La API responde con los datos solicitados.
7. ¿Cuál es el esquema de seguridad que deben cumplir mis servicios backend para que se pueda establecer una comunicación segura hacia ellos?.- Deben cumplir con el marco OAuth 2.0
8. Explique qué es TLS.- El TLS es un protocolo criptográfico que garantiza una transmisión segura en Internet, se refiere a la siguiente generación del Certificado SSL. Permite y garantiza el intercambio de datos en un entorno securizado y privado entre el usuario y el servidor, mediante aplicaciones como HTTP, POP3, IMAP, SSH, SMTP o NNTP. Realiza la encriptación mediante dos protocolos en capas diferentes: el protocolo de autenticación llamado TLS Record Protocol y el de mútuo acuerdo también conocido como TLS Handshake Protocol.
9. Explique cómo funciona el esquema de seguridad JWT.- Es un estándar abierto basado en JSON para crear un token que contiene la identidad de un usuario. El JWT viaja firmado digitalmente y consta de 3 partes:. headers, payload y la firma.

Su funcionamiento es el siguiente:

1. El usuario envía sus datos de inicio de sesión al servidor,
2. El servidor valida las credenciales y genera el JWT
3. El servidor envía el JWT como respuesta a la aplicación cliente,
4. El cliente envía este token en cada petición
5. El servidor verifica que el JWT sea válido y responde la petición.
6. Explique qué entiende por proveedor de identidad.- Un IDM, es un servicio particular que se encarga generalmente de los servicios de directorio, gestión de identidades de usuarios, administración de accesos y políticas organizacionales.