JAVA EE PROFILES Y JAVA EE SERVICES Y API’S

API:

Un conjunto de definiciones y protocolos que se utilizan para desarrollar e integrar el software de las aplicaciones. API significa Interfaz de Programación de Aplicaciones.

* Permite la comunicación de productos y servicios entre sí sin necesidad de saber cómo están implementados.
* Simplifica el desarrollo de las aplicaciones y permite ahorrar tiempo.
* Otorga flexibilidad.
* Simplifica el diseño, la administración y el uso de las aplicaciones.
* Proporciona oportunidades de innovación.

Enfoques:

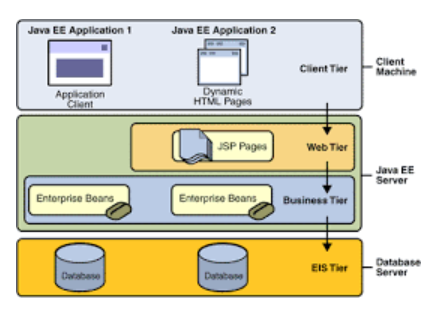
* Privado. Solo pueden ser usadas internamente por lo que las empresas tienen un mayor control sobre ellas.
* De partners. Las APIs se comparten con partners empresariales específicos lo que puede ofrecer flujos de ingresos adicionales sin comprometer la calidad.
* Público. Todo el mundo tiene acceso a las APIs, esto permite que terceros desarrollen aplicaciones que interactúen con una API pública.

JAVA EE

Java Enterprise Edition (Java EE) se basa en Java SE, representa una colaboración entre varios proveedores y líderes del sector proporcionando el soporte de la infraestructura para las aplicaciones.

Algunas de sus características:

* Es portable y escalable.
* Soporta la integración existente y los componentes basados en la arquitectura EJB.
* Simplifica las aplicaciones empresariales definiendo y especificando un conjunto de servicios estándar comunes: denominación, gestión de transacciones, simultaneidad, seguridad y acceso a base de datos.
* Define un modelo de contenedor que aloja y gestiona instancias de componentes de aplicaciones Java EE. Estos contenedores se alojan en servidores Java EE.



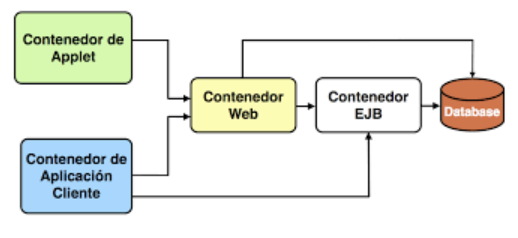
Capas:

* Client Tier. Es donde se muestra la interfaz gráfica de usuario y se ejecuta en el ordenador del cliente.
* Web Tier, Business Tier. Estas capas se ejecutan en el servidor de aplicaciones.
* EIS Tier o Data Tier. Esta capa puede separarse en una o más capas.

Un servidor Java EE es una aplicación de servidor que implementa las APIs de la plataforma Java EE y proporciona los servicios estándar de Java EE. A menudo se les llama servidores de aplicaciones porque permiten entregar datos de aplicaciones a los clientes, así como los servidores web sirven páginas web a los navegadores web.

Un contenedor Java EE es la interfaz entre el componente y la funcionalidad de nivel inferior proporcionada por la plataforma para dar soporte a ese componente. La funcionalidad del contenedor está definida por la plataforma y es diferente para cada tipo de componente. Sin embargo, el servidor permite que los diferentes tipos de componentes trabajen juntos para proporcionar funcionalidad en una aplicación empresarial.

* Contenedor Web. Es la interfaz entre los componentes web y el servidor web. Un componente web puede ser un servlet, una página de Facelets de JavaServer Faces o una página JSP. El contenedor administra el ciclo de vida del componente, envía solicitudes a los componentes de la aplicación y proporciona interfaces a los datos de contexto, como información sobre la solicitud actual.
* Contenedor cliente de la aplicación. Es la interfaz entre los clientes de la aplicación Java EE. Este contenedor se ejecuta en la máquina del cliente y es la puerta de enlace entre la aplicación del cliente y de los componentes del servidor Java EE que utiliza el cliente.
* Contenedor EJBs. Es la interfaz entre los beans empresariales que proporcionan la lógica empresarial en una aplicación Java EE y el servidor Java EE.



API’S

* API de persistencia de Java(JPA). Es una solución basada en estándares de Java para la persistencia. La persistencia utiliza un enfoque de mapeo relacional/objeto para cerrar la brecha entre un modelo orientado a objetos y una base de datos relacional. Esta API también puede utilizarse en aplicaciones Java SE fuera del entorno Java EE.
* API de transacciones Java(JTA). Brinda una interfaz estándar para demarcar transacciones. La arquitectura Java EE proporciona una confirmación automática predeterminada para manejar las confirmaciones y reversiones de transacciones. Una confirmación automática significa que cualquier otra aplicación que esté viendo datos verá los datos actualizados después de cada operación de lectura o escritura de la base de datos. Sin embargo, si su aplicación realiza dos operaciones de acceso a la base de datos separadas que dependen una de la otra, querrá usar la API de JTA para demarcar dónde comienza, revierte y confirma toda la transacción, incluidas ambas operaciones.
* API de Java para servicios web RESTful(JAX-RS). Define una API para el desarrollo de servicios web construidos de acuerdo con el estilo arquitectónico Representational State Transfer (REST). Una aplicación JAX-RS es una aplicación web que consta de clases empaquetadas como un servlet en un archivo WAR junto con las bibliotecas necesarias.
* API de servicio de mensajes de JAVA(JSM). Es un estándar de mensajería que permite a los componentes de la aplicación Java EE crear, enviar, recibir y leer mensajes. Permite la comunicación distribuida que está débilmente acoplada, es confiable y asíncrona.
* API de JavaMail. Se utiliza para enviar notificaciones por correo electrónico. Esta API tiene dos partes:
* Una interfaz a nivel de aplicación usada por los componentes de la aplicación para enviar correo.
* Una interfaz de proveedor de servicios. Esto permite que los componentes de la aplicación envíen correo de internet.
* API de seguridad Java EE. Define interfaces de complemento portátiles para la autenticación HTTP y los almacenes de identidad, y una interfaz SecurityContext inyectable que proporciona una API para la seguridad programática.
* API de Java para WebSocket. WebSocket es un protocolo de aplicación que proporciona comunicaciones full-duplex entre dos pares a través de TCP. La API de Java para WebSocket permite que las aplicaciones Java EE creen puntos finales utilizando anotaciones que especifican los parámetros de configuración del punto final y designan sus métodos de devolución de llamada del ciclo de vida.
* API de Java para procesamiento JSON(JSON-P). Permite que las aplicaciones Java EE analicen, transformen y consulten datos JSON mediante el modelo de objetos o el modelo de transmisión.
* API de Java para enlaces JSON(JSON-B). Proporciona una capa de enlace para convertir objetos Java hacia y desde mensajes JSON. JSON-B también admite la capacidad de personalizar el proceso de mapeo predeterminado utilizado en esta capa de enlace mediante el uso de anotaciones Java para un campo, propiedad JavaBean, tipo o paquete determinado, o proporcionando una implementación de una estrategia de nomenclatura de propiedad.

PERFILES JAVA EE

Un perfil es un conjunto de tecnologías y API de Java EE que atienden a comunidades de desarrolladores y tipos de aplicación específicos.

* Perfil de plataforma completa. Este ha sido diseñado para desarrolladores que requieren el conjunto completo de la API de Java EE para el desarrollo de aplicaciones empresariales. Este perfil se instala al instalar Sun GlassFish Enterprise Server vv aunque también se instala como parte de la instalación del SDK de Java EE 6.
* Perfil web. Contiene tecnologías web que forman parte de la plataforma completa y está diseñado para desarrolladores que no precisen todo el conjunto de la API de Java EE. Se instala al instalar el perfil web de Sun GlassFish Enterprise Server v3 aunque también se instala con SDK del perfil web de Java EE 6.

