Construcción de Sistemas Distribuidos

“Middleware”

Rogelio Ferreira Escutia

Contenido

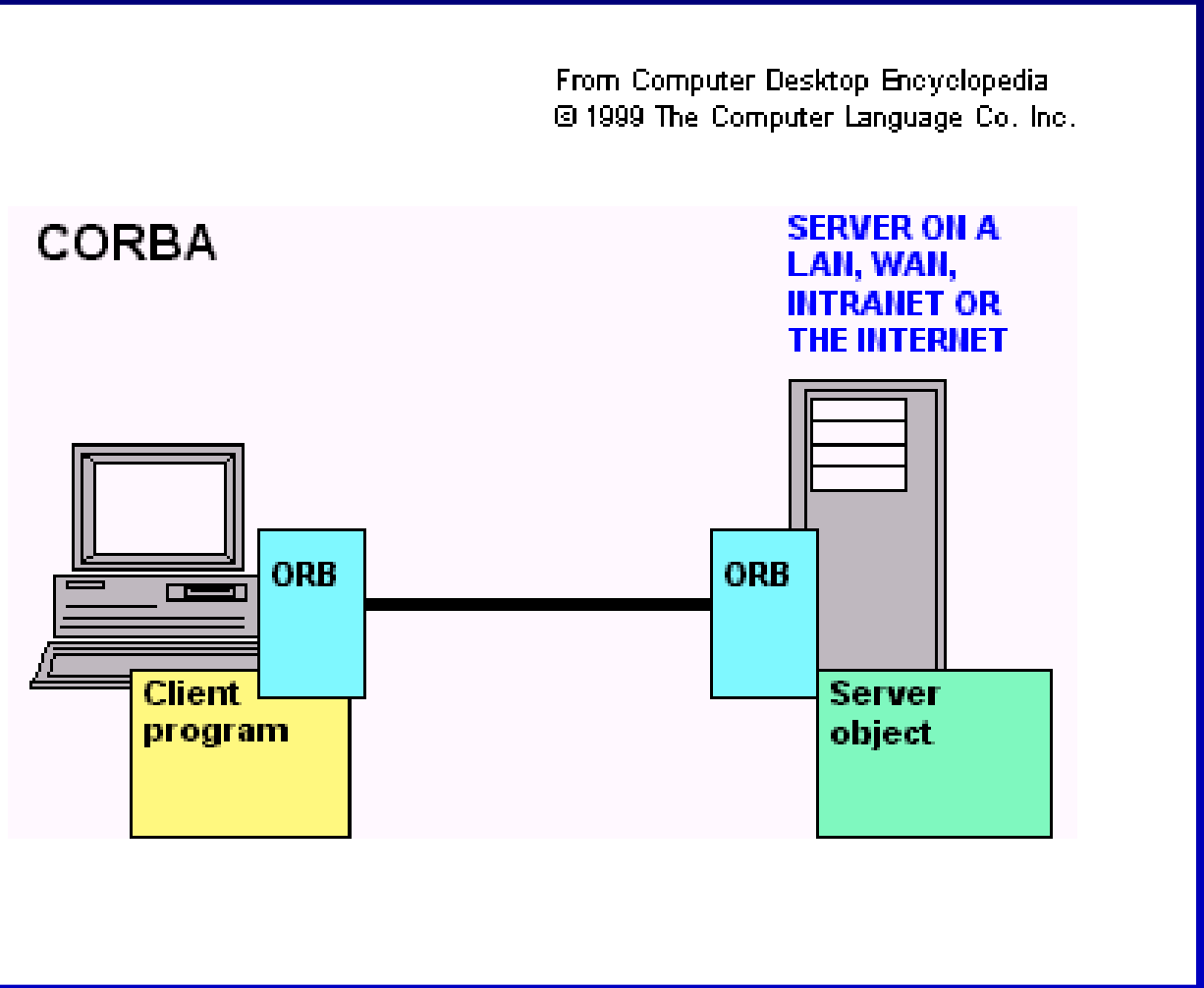
1) Sockets
2) RPC´s
3) CORBA
4) .NET
5) Java

3) CORBA

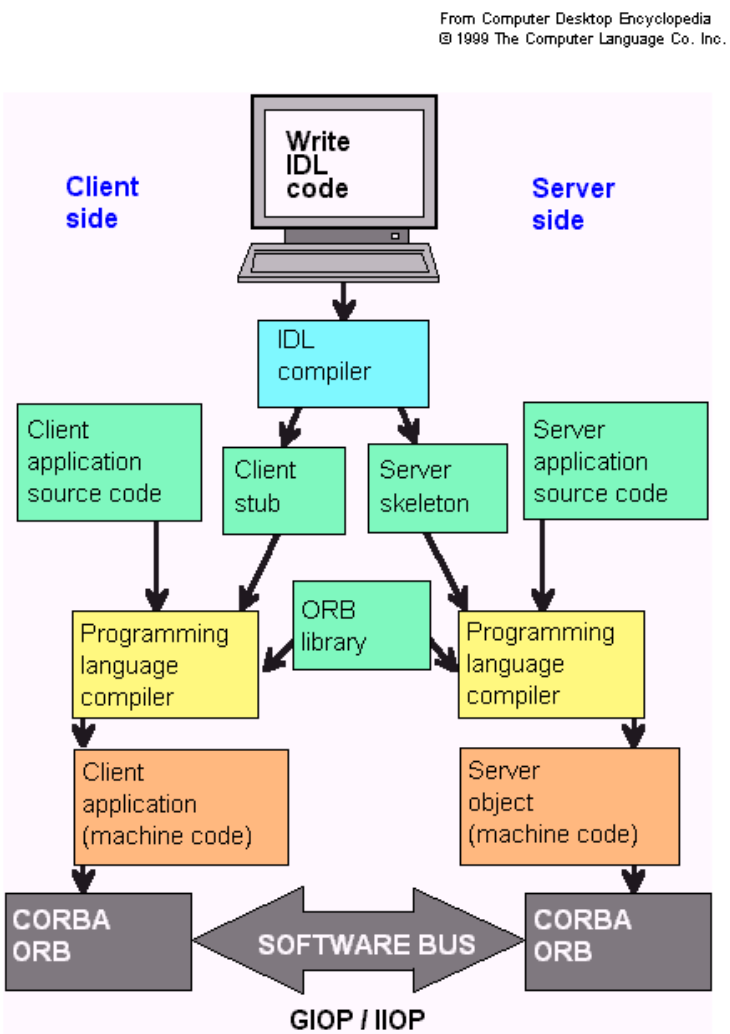
Sistemas Abiertos

- **InicialMente había grandes computadoras (Mainframes) donde se centralizaba la información.**
- **Posteriormente, las computadoras bajaron de precio, por lo que su número aumentó y se les podía encontrar en diferentes lugares de las empresas.**
- **Estas computadoras tenían diferentes tipos de hardware y software, dependiendo de la aplicación para la cual se utilizaran.**
- **Una vez que la cantidad de información aumentó, se requirió de interconectar las diferentes computadoras para compartir información.**

Arquitectura de CORBA



Arquitectura de CORBA



4) .NET

.NET

- Microsoft .NET es la plataforma de Microsoft para la creación y el uso de servicios Web XML (Extensible Markup Language, Lenguaje de marcado extensible).

- Esta plataforma permitirá a los programadores la creación de programas que trasciendan los límites de los dispositivos y aprovechen la conectividad de Internet, además de ayudarles a ser más productivos con su tiempo.

- La plataforma .NET representa también un cambio fundamental en la tecnología de desarrollo.

- En realidad esta plataforma no es algo radicalmente nuevo. Es un conjunto de tecnologías dispersas, que en muchos casos ya existían, que Microsoft ha integrado en una plataforma común con el objetivo de facilitar el desarrollo de este nuevo tipo de servicios de tercera generación.

.NET

Microsoft .NET es una plataforma para construir, ejecutar y experimentar la tercera generación de aplicaciones distribuidas, que consiste en los siguientes elementos:

- Un modelo de programación basado en XML.
- Un conjunto de servicios Web XML, como *Microsoft .NET My Services* para facilitar a los desarrolladores integrar estos servicios.
- Un conjunto de servidores que permiten ejecutar estos servicios (como *.NET Enterprise Servers*).
- Software en el cliente para poder utilizar estos servicios (como Windows XP, agendas electrónicas, etc.)
- Herramientas para el desarrollo como *VisualStudio.NET*.

.NET

- Una parte importante de esta plataforma es el software de los dispositivos clientes y servidores, que ha sido el mercado habitual de Microsoft.

- Para los dispositivos clientes, Microsoft planea integrar .NET en cualquier dispositivo imaginable, como PCs con Windows, agendas electrónicas con Pocket PC, teléfonos móviles, su consola de videojuegos X-Box, en WebTV, etc.

- Esto supone para las empresas aumentar el numero de potenciales clientes que puedan utilizar su servicios (ya no están limitados al PC).

- Para poder ejecutar estos servicios, Microsoft introduce una serie de software englobado dentro de los .NET Enterprise Servers, como es el *Application Center*, *Commerce Server*, etc.

Arquitectura .NET

- **Una definición general de la arquitectura .NET podría ser la siguiente: "Una plataforma independiente del lenguaje para el desarrollo de servicios Web".**

- **La arquitectura .NET (*.NET Framework*) es el modelo de programación de la plataforma .NET para construir y ejecutar los servicios .NET.**

- **El objetivo de esta arquitectura es la de reducir la complejidad en el desarrollo de este tipo de aplicaciones, permitiendo a los desarrolladores centrarse en escribir la lógica específica del servicio a desarrollar.**

Arquitectura .NET

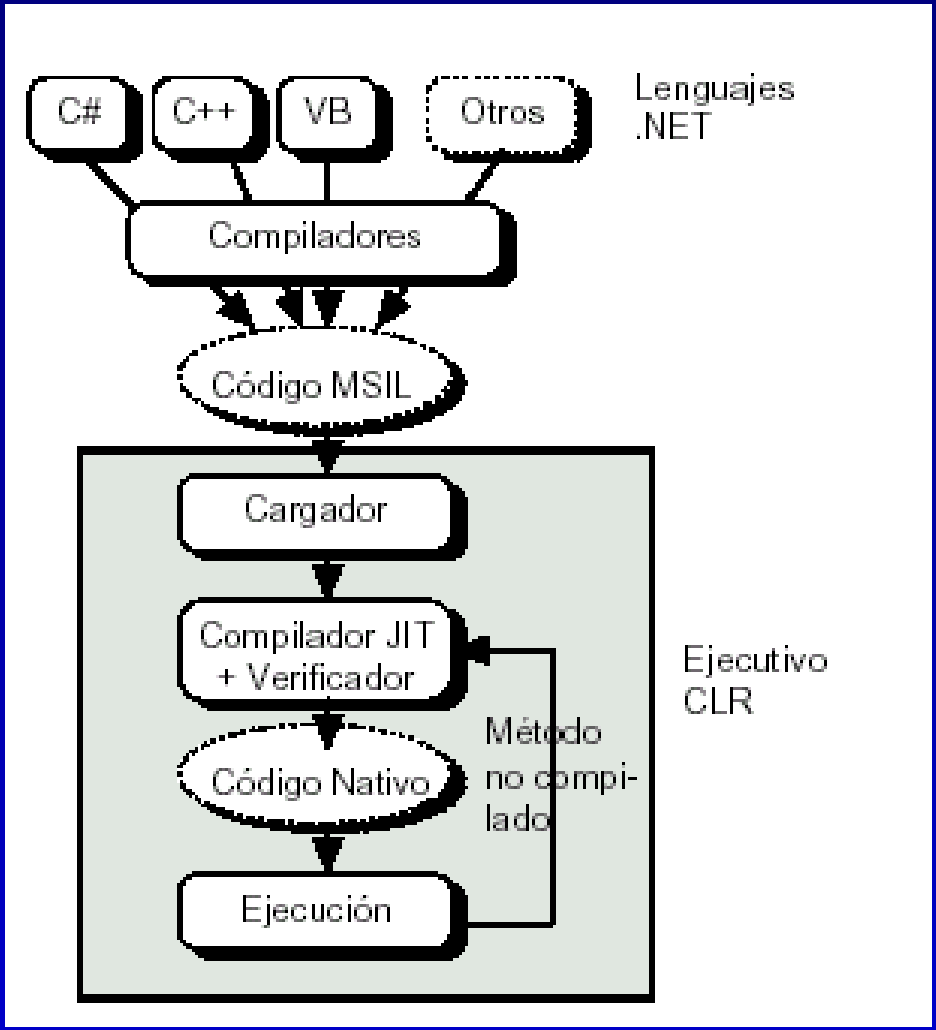
- **Esta arquitectura está compuesta por librerías:**

Arquitectura .NET

- Las *librerías básicas* proporcionan una serie de funcionalidades que son necesarias a la hora de desarrollar los servicios Web.
- Las *clases básicas* gestionan las operaciones más básicas como las comunicaciones, entrada/salida, seguridad, etc. Las *clases XML y de datos* gestionan el acceso a base de datos y la gestión de datos en XML.
- El objetivo de las librerías *Servicios Web XML* es la de dar soporte para el desarrollo de aplicaciones distribuidas que ofrezcan servicios XML a otras entidades.
- Las *Web forms* permiten desarrollar la parte gráfica de una aplicación para la Web, mientras las *Windows Forms* están orientadas a implementar la parte gráfica de las aplicaciones clásicas para Windows.

Arquitectura .NET

- **Modelo de ejecución de los programas .NET**



Arquitectura .NET

- Los compiladores producen código MSIL (*MicroSoft Intermediate Language*), que es un lenguaje intermedio que se puede ejecutar en la máquina virtual.
- Este código no es interpretado por el ejecutivo, sino que es compilado de nuevo en tiempo de ejecución (JIT: *Just in Time*) al código nativo de la máquina.
- Este código compilado no se ejecuta independientemente sino dentro de este ejecutivo.
- Esto se denomina código manejado, lo cual permite que el ejecutivo controle ciertos aspectos de la aplicación que ejecuta como son seguridad, gestión de Memoria, compartición de datos, etc.

Arquitectura .NET

Ventajas de .NET:

- Tiene una historia de mejores kits de herramientas
- Puede manejar múltiples lenguajes en una aplicación.
- Modelo más simple de programación, lo que permite que programadores promedio puedan desarrollar más rápidamente pero con menos control.
- Alta integración con el sistema operativo.
- Es una apuesta muy fuerte del fabricante de software líder en ventas.
- Es más eficiente que las anteriores plataformas de Microsoft, y las aplicaciones creadas sobre .Net son más fáciles de instalar.
- C# es un buen lenguaje, y la migración desde Visual Basic a Visual Basic .Net, aunque no es inmediata, es factible.
- La curva de aprendizaje es relativamente suave.
- Microsoft ha desarrollado buenas herramientas 'visuales' de desarrollo (Visual C# y Visual Studio .Net)

Arquitectura .NET

Desventajas de .NET:

- **Dependencia de un solo proveedor.**
- **Por ser un cambio muy fuerte en arquitectura, puede contener los problemas de primeras versiones.**
- **Al poder combinar múltiples lenguajes, puede dar lugar a código mantenible sólo por ciertas personas.**
- **Poco reaprovechamiento de la experiencia de recursos humanos especializados en Microsoft, ya que cambia drásticamente la plataforma.**

5) Java

¿Qué es Java?

Plataforma Java

- La plataforma Java es una nueva forma de trabajar con computadoras, basado en el poder de las redes y en la idea de que el mismo software pueda correr en diferentes tipos de computadoras.
- La idea es crear un software que trabaje desde pequeños dispositivos hasta supercomputadoras, el único requisito es el dispositivo soporte la plataforma Java.
- En la actualidad, la plataforma Java es soportada por teléfonos, tarjetas inteligentes, electrodomésticos, etc.

- What is the Java Platform, http://java.sun.com/java2/whatis/1996/, marzo 2004

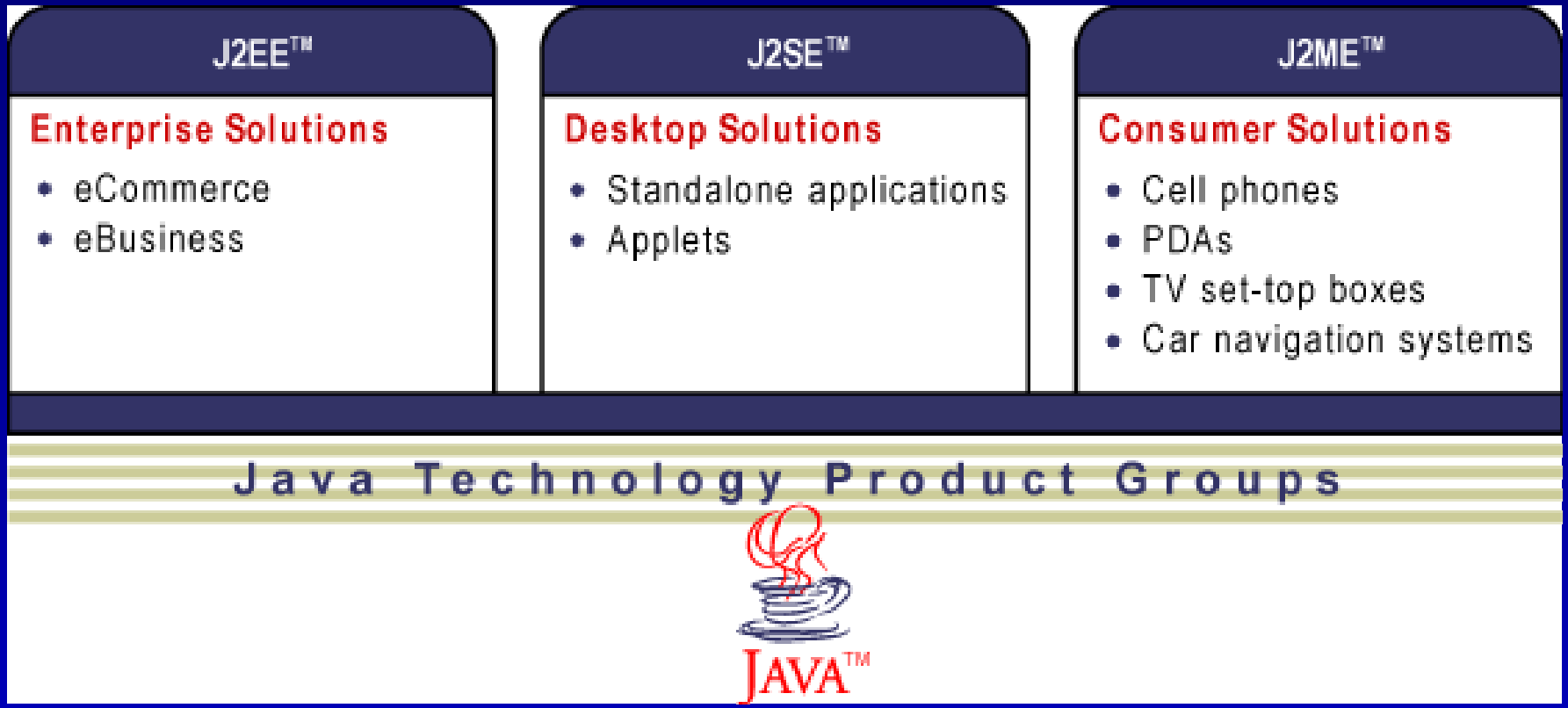
Plataforma Java

Plataforma Java

- La plataforma SUN esta formado por los siguientes componentes:

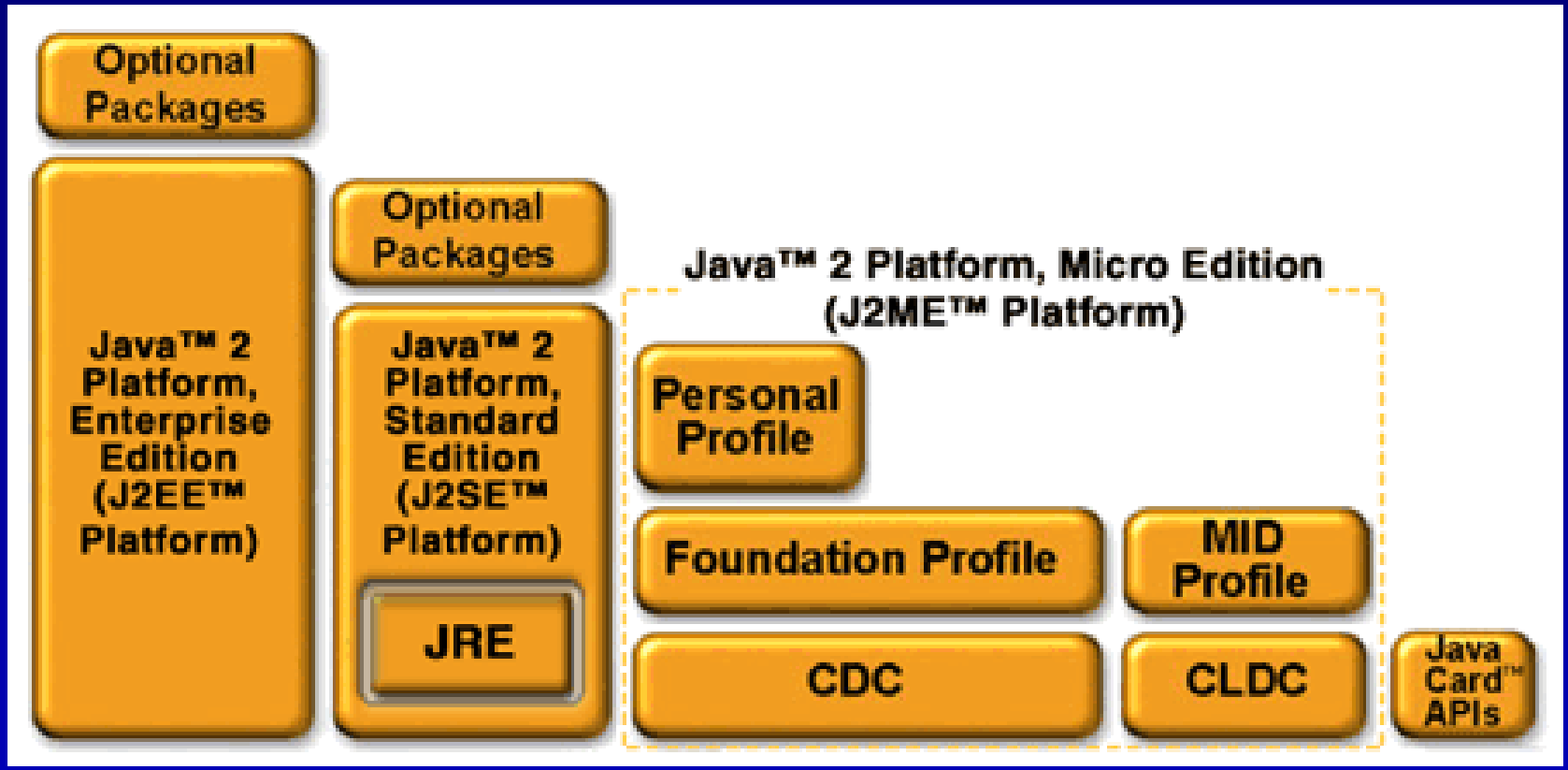
  J2SE: Aplicaciones de escritorio.
  J2EE: Aplicaciones empresariales.
  J2ME: Aplicaciones incrustadas (Embedded Systems).
  Java Card: Tarjetas inteligentes.

Plataforma Java



- SUN - http://www.sun.com – marzo 2004

Plataforma Java



- Java 2 Platform, http://java.sun.com/java2/whatis/index.html, marzo 2004

Java Card

- Una tarjeta inteligente es una tarjeta del tamaño de una tarjeta de crédito con un circuito integrado en su interior.
- Un circuito contiene un microprocesador y memoria, los cuales le dan la habilidad a la tarjeta de procesar y almacenar información.
- En el caso de la plataforma Java Card, las aplicaciones en forma de byte-code son cargadas en la zona de memoria, para después ser ejecutadas por la máquina virtual.
- El código ejecutable es independiente de la plataforma, por lo que cualquier tarjeta que tenga incorporada una máquina virtual podrá ejecutar el código.

• Java 2 Platform, http://java.sun.com/java2/whatis/index.html, marzo 2004

Historia de Java

Java - Historia

- Es un lenguaje que se originó en 1991 como parte de un proyecto de investigación ("Green Team", formado por Patrick Naughton, Mike Sheridan, y James Gosling) para desarrollar un nuevo lenguaje de programación llamado "Oak".
- Este lenguaje se enfocó para comunicar dispositivos caseros como televisiones y videocaseteras, los cuales pudieran compartir el mismo software pero utilizando diferentes microprocesadores.
- Este lenguaje no prospero en el mercado de dispositivos caseros, por lo que se cambió el mercado en el cual se emplearía, optando utilizarlo en el naciente Web, para el desarrollo de componentes multimedia en páginas web.
- Estas aplicaciones fueron denominadas "applets" y fue en ese entonces que se cambió el nombre por Java.

Java - Historia

- **"Green Team" de izquierda a derecha: *Al Frazier, Joe Palrang, Mike Sheridan, Ed Frank, Don Jackson, Faye Baxter, Patrick Naughton, Chris Warth, James Gosling, Bob Weisblatt, David Lavallee, and Jon Payne.***

- JAVA TECHNOLOGY: THE EARLY YEARS, http://java.sun.com/features/1998/05/birthday.html, marzo 2004

Java - Historia

- **Dispositivo e interface del prototipo denominado *7 (Star Seven) con el cual se hizo una demostración de la plataforma Java (aún no denominada con este nombre).**

- JAVA TECHNOLOGY: THE EARLY YEARS, http://java.sun.com/features/1998/05/birthday.html, marzo 2004

Características de la Plataforma Java

Java - Características

- El lenguaje de programación Java fue desarrollado para tener las siguientes características:

  1) Orientado a Objetos.
  2) Distribuido.
  3) Simple.
  4) Multihilo.
  5) Seguro.
  6) Independiente de la plataforma.

Java – Orientado a Objetos

- **Java es un lenguaje de Programación Orientado a Objetos (Object-Oriented Programming), por lo que se busca crear objetos, o piezas de código autónomo, que pueda interactuar con otros objetos para resolver un problema.**

Plataforma Java 2

Plataforma Java 2

Las tres ediciones de la plataforma Java 2 son:

- **Standard Edition (J2SE).- Esta es la plataforma básica y mas extendida, esta misma también conocida como Java Development Kit o JDK. Con ella se pueden hacer applets (pequeña aplicación escrita en Java) y cualquier clase de aplicación no distribuida.**

- **Enterprise Edition (J2EE).- Funciona sobre J2SE, añadiéndole una serie de APIs y que permiten crear aplicaciones de empresa en el lado del servidor. J2EE, Java 2 Enterprise Edition es una especificación acordada entre proveedores para crear, tanto aplicaciones empresariales como las plataformas que las soportan. De tal modo que exista libertad en la selección del proveedor de la plataforma y a la vez se asegure su portabilidad.**

- **Micro Edition (J2ME).- Finalmente esta ofrece un entorno de ejecución altamente optimizado para todo tipo de dispositivos con recursos limitados, como móviles, PDAs, etcétera.**

J2EE

J2EE

- El *objetivo principal de la plataforma J2EE* es la de proveer un estándar simple y unificado para aplicaciones distribuidas a través de un modelo de aplicaciones basado en componentes.

- Las aplicaciones que se pueden desarrollar en J2EE están fuertemente basadas en arquitecturas distribuidas, lo que permite escalabilidad además de su aplicación al Web, Cliente-Servidor, integración a sistemas legados.

- Define un conjunto de lineamientos para los programadores de aplicaciones y otro para los proveedores de servidores de aplicaciones.

- Desarrollar sobre J2EE *(Java 2 Enterprise Edition)* permite concentrarse en el desarrollo de componentes que representen la lógica de un negocio, y brindarle a estos componentes servicios como transacciones, seguridad y mensajería utilizando estándares ampliamente aceptados y probados por la industria.

J2EE

- Diseñar para una arquitectura distribuida como J2EE promueve que las aplicaciones sean escalables y modulares, características muy deseables en las aplicaciones de negocios de hoy día.
- Enterprise Edition (J2EE) es perfecta para desarrollar sólidas aplicaciones empresariales que permiten un desarrollo acelerado.
- Los Servicios de Java para la Web es una herramienta integrada que junto con la plataforma de Java les permite a diseñadores de Java construir, hacer pruebas, desplegar las aplicaciones de XML, servicios de la Web, y aplicaciones de la Web.
- Las tecnologías que comprenden el Java WSDP incluyen los API de Java para XML.

J2EE

Ventajas de J2EE sobre otras plataformas

- 1) Independencia de plataforma.- Las aplicaciones J2EE son portables. Si cambiamos el sistema operativo de nuestros servidores y/o el programa servidor de aplicaciones, podremos seguir utilizándolas sin necesidad de hacer modificaciones a las aplicaciones de J2EE ya desarrolladas.

- 2) Objetos gestionados por los contenedores.- El hecho de que los contenedores gestionen los objetos permite al desarrollador abstraerse de cantidad de aspectos que requieren tiempo y dedicación. Como consecuencia los desarrollos son más rápidos, con menos coste, y se obtiene un código mucho menos propenso a fallos y de más fácil mantenimiento. Además las aplicaciones J2EE son declarativas, lo que quiere decir que se puede modificar su comportamiento sin modificar el código, lo cual contribuye también a un mantenimiento mucho más rápido y sencillo.

J2EE

Ventajas de J2EE sobre otras plataformas

- **3) Reusabilidad.- Las aplicaciones J2EE están constituidas por la unión de componentes, como si se tratase de las piezas de un rompecabezas. Así se consiguen desarrollos más rápidos.**

- **4) Modularidad.- Facilita enormemente el mantenimiento de las aplicaciones, ya que si se desea realizar modificaciones en algún módulo, éstas no deberían afectar al resto.**

J2EE

Desventajas de J2EE

- **1) Dado que pueden participar varios proveedores en una solución, puede haber problemas de integración. Se recomiendan soluciones de un solo proveedor.**
- **2) Herramientas poco integradas entre proveedores.**
- **3) Las aplicaciones de otros lenguajes se consideran fuera de la plataforma.**

Middleware