Historia de la programación Orientada a objeto

Los conceptos de la programación orientada a objetos tienen origen en [Simula 67](http://es.wikipedia.org/wiki/Simula), un lenguaje diseñado para hacer simulaciones, creado por [Ole-Johan Dahl](http://es.wikipedia.org/wiki/Ole-Johan_Dahl) y [Kristen Nygaard](http://es.wikipedia.org/wiki/Kristen_Nygaard) del Centro de Cómputo Noruego en [Oslo](http://es.wikipedia.org/wiki/Oslo).

La programación orientada a objetos se fue convirtiendo en el estilo de programación dominante a mediados de los años ochenta, en gran parte debido a la influencia de [C++](http://es.wikipedia.org/wiki/C%2B%2B), una extensión del [lenguaje de programación C](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_C). Su dominación fue consolidada gracias al auge de las [Interfaces gráficas de usuario](http://es.wikipedia.org/wiki/Interfaz_gr%C3%A1fica_de_usuario), para las cuales la programación orientada a objetos está particularmente bien adaptada. En este caso, se habla también de [programación dirigida por eventos](http://es.wikipedia.org/wiki/Programaci%C3%B3n_dirigida_por_eventos).

Historia del lenguaje Java

**Java** es un [lenguaje de programación](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n) [orientado a objetos](http://es.wikipedia.org/wiki/Orientado_a_objetos), desarrollado por [Sun Microsystems](http://es.wikipedia.org/wiki/Sun_Microsystems) a principios de los [años 90](http://es.wikipedia.org/wiki/A%C3%B1os_90).

Java se creó como una herramienta de programación para ser usada en un proyecto de [set-top-box](http://es.wikipedia.org/wiki/Set-top_box) en una pequeña operación denominada *the Green Project* en Sun Microsystems en el año [1991](http://es.wikipedia.org/wiki/1991). El equipo (*Green Team*), compuesto por trece personas y dirigido por [James Gosling](http://es.wikipedia.org/wiki/James_Gosling), trabajó durante 18 meses en Sand Hill Road en Menlo Park en su desarrollo.

El lenguaje se denominó inicialmente *Oak* (por un roble que había fuera de la oficina de Gosling), luego pasó a denominarse *Green* tras descubrir que *Oak* era ya una marca comercial registrada para adaptadores de tarjetas gráficas y finalmente se renombró a *Java*.

Concepto, campo de aplicación de java.

**Java** es un lenguaje de programación orientado a objetos. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria. Con respecto a la memoria, su gestión no es un problema ya que ésta es gestionada por el propio lenguaje y no por el programador.

Su campo de aplicación se ve en:

**Dispositivos móviles y sistemas empotrados**

Desde la creación de la especificación J2ME (Java 2 Platform, Micro Edition), una versión del entorno de ejecución Java reducido y altamente optimizado, especialmente desarrollado para el mercado de dispositivos electrónicos de consumo se ha producido toda una revolución en lo que a la extensión de Java se refiere.

Es posible encontrar microprocesadores diseñados para ejecutar bytecode Java y software Java para tarjetas inteligentes (JavaCard), teléfonos móviles, buscapersonas, set-top-boxes, sintonizadores de TV y otros pequeños electrodomésticos.

### En el navegador web

Desde la primera versión de java existe la posibilidad de desarrollar pequeñas aplicaciones ([Applets](http://es.wikipedia.org/wiki/Applet_Java)) en Java que luego pueden ser incrustadas en una página HTML para que sean descargadas y ejecutadas por el navegador web. Estas mini-aplicaciones se ejecutan en una JVM que el navegador tiene configurada como extensión (*plug-in*) en un contexto de seguridad restringido configurable para impedir la ejecución local de código potencialmente malicioso.

**En sistemas de servidor**

En la parte del servidor, Java es más popular que nunca, desde la aparición de la especificación de [Servlets](http://es.wikipedia.org/wiki/Java_Servlet) y JSP ([Java Server Pages](http://es.wikipedia.org/wiki/Java_Server_Pages)).

Hasta entonces, las aplicaciones web dinámicas de servidor que existían se basaban fundamentalmente en componentes [CGI](http://es.wikipedia.org/wiki/Common_Gateway_Interface) y lenguajes interpretados. Ambos tenían diversos inconvenientes (fundamentalmente lentitud, elevada carga computacional o de memoria y propensión a errores por su interpretación dinámica).

Los servlets y las JSPs supusieron un importante avance ya que:

* El [API](http://es.wikipedia.org/wiki/Interfaz_de_programaci%C3%B3n_de_aplicaciones) de programación es muy sencilla, flexible y extensible.
* Los servlets no son procesos independientes (como los CGIs) y por tanto se ejecutan dentro del mismo proceso que la JVM mejorando notablemente el rendimiento y reduciendo la carga computacional y de memoria requeridas.
* Las JSPs son páginas que se compilan dinámicamente (o se pre-compilan previamente a su distribución) de modo que el código que se consigue una ventaja en rendimiento substancial frente a muchos lenguajes interpretados.

La especificación de Servlets y JSPs define un API de programación y los requisitos para un contenedor (servidor) dentro del cual se puedan desplegar estos componentes para formar aplicaciones web dinámicas completas. Hoy día existen multitud de contenedores (libres y comerciales) compatibles con estas especificaciones.

### En aplicaciones de escritorio

Hoy en día existen multitud de aplicaciones gráficas de usuario basadas en Java. El entorno de ejecución Java (JRE) se ha convertido en un componente habitual en los PC de usuario de los sistemas operativos más usados en el mundo. Además, muchas aplicaciones Java lo incluyen dentro del propio paquete de la aplicación de modo que se ejecuten en cualquier [PC](http://es.wikipedia.org/wiki/Computador_personal).

En las primeras versiones de la plataforma Java existían importantes limitaciones en las APIs de desarrollo gráfico ([AWT](http://es.wikipedia.org/wiki/Abstract_Window_Toolkit)). Desde la aparición de la biblioteca [Swing](http://es.wikipedia.org/wiki/Swing_%28biblioteca_gr%C3%A1fica%29) la situación mejoró substancialmente y posteriormente con la aparición de bibliotecas como [SWT](http://es.wikipedia.org/wiki/SWT) hacen que el desarrollo de aplicaciones de escritorio complejas y con gran dinamismo, usabilidad, etc. sea relativamente sencillo.

Caracteristicas:

### Orientado a objetos

Java fue diseñado como un lenguaje orientado a objetos desde el principio. Los objetos agrupan en estructuras encapsuladas tanto sus datos como los métodos (o funciones) que manipulan esos datos. La tendencia del futuro, a la que Java se suma, apunta hacia la programación orientada a objetos, especialmente en entornos cada vez más complejos y basados en red.

### Distribuido

Java proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir sockets y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.

### Interpretado y compilado a la vez

Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador.
Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real.

### Robusto

Java fue diseñado para crear software altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores (la aritmética de punteros), ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.

### Portable

La indiferencia a la arquitectura representa sólo una parte de su portabilidad. Además, Java especifica los tamaños de sus tipos de datos básicos y el comportamiento de sus operadores aritméticos, de manera que los programas son iguales en todas las plataformas.

**Entorno de desarrollo de Java.**

**Eclipse** es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Eclipse fue desarrollado originalmente por [IBM](http://es.wikipedia.org/wiki/IBM) como el sucesor de su familia de herramientas para [VisualAge](http://es.wikipedia.org/w/index.php?title=VisualAge&action=edit&redlink=1). Eclipse es ahora desarrollado por la [Fundación Eclipse](http://es.wikipedia.org/wiki/Fundaci%C3%B3n_Eclipse), una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.