1 Introdução

1.1 Evolução tecnológica

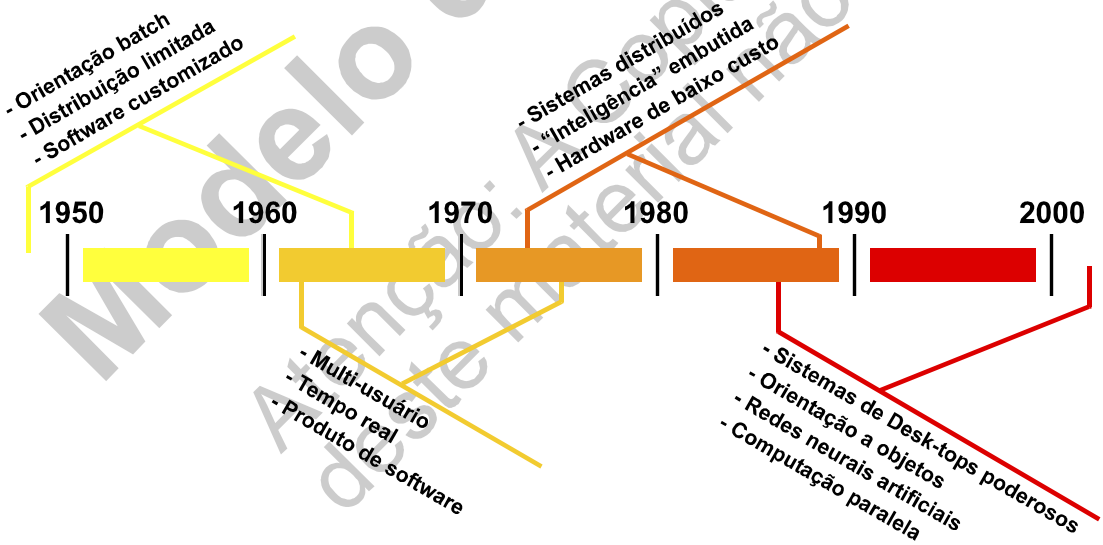
À medida que o mercado exige informatização torna-se cada vez mais necessário o uso de programas mais complexos e pesados, além de aumentar a velocidade de processamento e capacidade de armazenamento do hardware.

Podemos medir a evolução do hardware quantitativamente através do seu poder de processamento (clock do processador, medido em Hertz ou mega Hertz) e da sua capacidade de armazenamento (disco rígido - medido em mega bytes ou mais comumente em giga bytes).

A qualidade do software é medida através de sua confiabilidade, operabilidade, manutenibilidade, extensibilidade, escalabilidade, entre outras métricas. Grande parte das métricas de qualidade do software depende do processo de desenvolvimento do software. Por isto, para que haja maior produtividade e qualidade do software existem cada vez mais softwares de apoio, tais como:

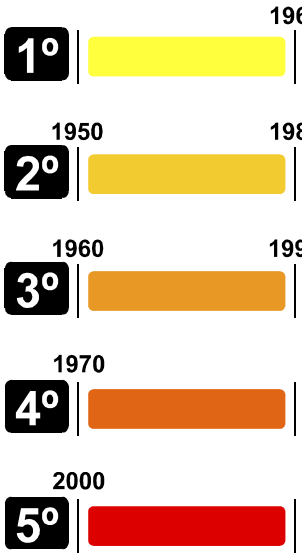
- Compiladores;
- Ambientes de desenvolvimento;
- Servidores de aplicação;
- Banco de dados;
- APIs e frameworks.

Evolução tecnológica



Anotações

Evolução das linguagens de programação



- Código de Máquinas - Representa o mais baixo nível de abstração com o qual um programa pode ser representado	Assembler
- Base para todas as linguagens modernas - Amplo uso - Enormes bibliotecas de software	Fortran, Cobol, Algol, Basic
- Linguagem de programação estruturada - Linguagem Orientada a Objetos	C, Pascal, C++, Java, Smalltalk, Lisp, Prolog
- Linguagem de consulta - Linguagem Orientada a Objetos - Geradores de código	SQL, Java
Próximas linguagens reconhecerão vozes e gestos	

1.2 Paradigmas de programação

Dois paradigmas de programação bastante comuns são o Procedural e o Orientado a Objetos. As principais diferenças entre estes paradigmas são observadas na organização do código e nas técnicas de modelagem do problema.

1.2.1 Paradigma Procedural

Algumas características comuns deste paradigma são:

- Conjunto de instruções organizado em blocos para executar determinada tarefa (funções).
- Conjuntos de funções agrupadas por funcionalidade em bibliotecas.
- Modelagem através de fluxograma e grande utilização de algorítmos.
- Extremamente técnica.

Exemplos:

- Pascal;
- C;
- Perl;
- Basic

1.2.2 Paradigma Orientado a Objetos

A orientação a objetos é o paradigma de programação predominante atualmente e está aos poucos substituindo a programação procedural criada no inicio da década de 60.

Neste paradigma o programa é composto por objetos com propriedades e operações que podem ser executadas por eles, ou seja, a estrutura de dados é definida juntamente com as funções (métodos) que poderão ser executadas.

Todas as funcionalidades e atributos de cada entidade do sistema são armazenados em classes que representam esta entidade.

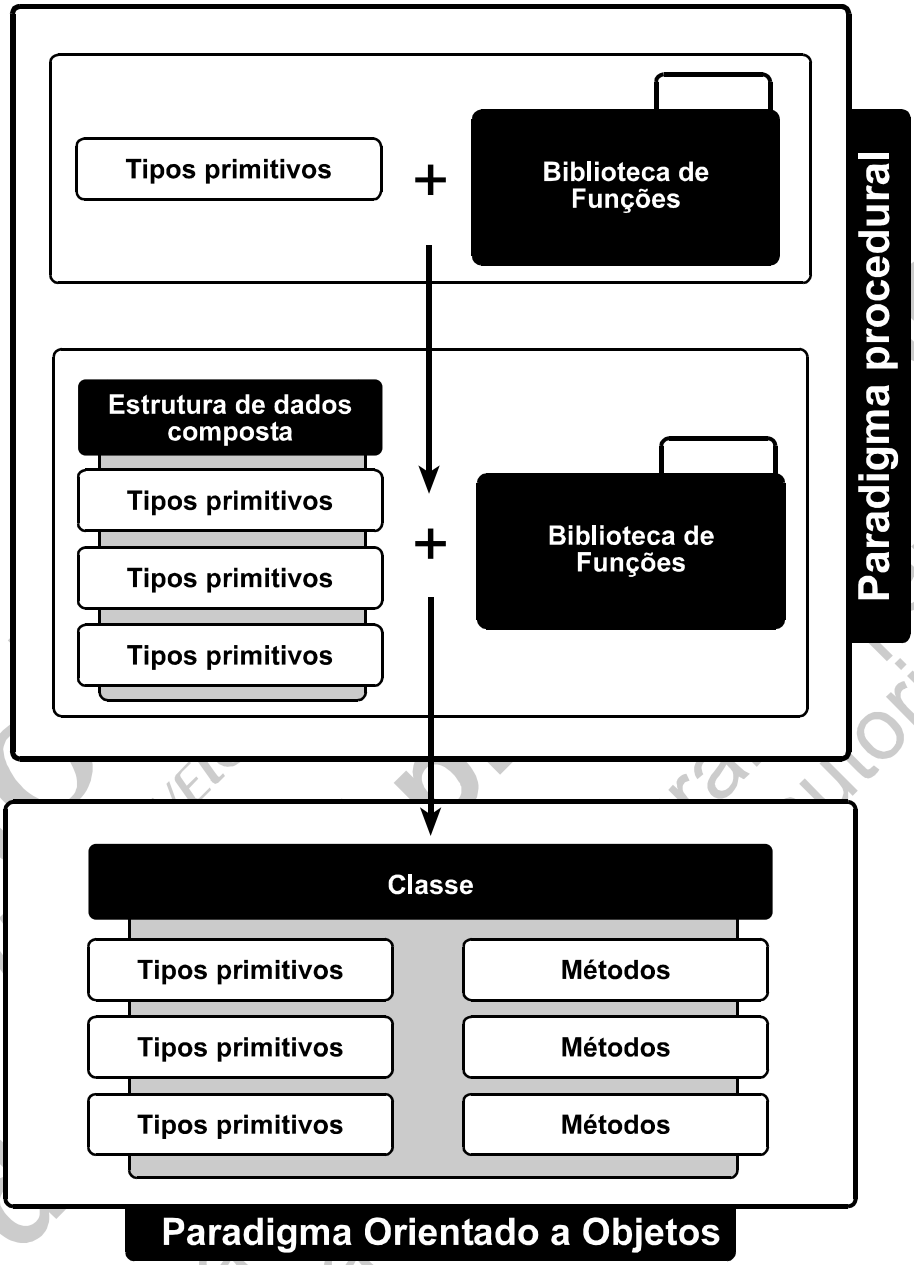
Normalmente, utilizamos a Unified Modeling Language (UML) para modelar soluções orientadas a objetos.

Exemplos:

- Smalltalk;
- Java;
- C++;
- C#.

Anotações

Evolução da estrutura de dados do paradigma procedural para o paradigma orientado a objetos.



1.3 Linguagens x Ambientes de desenvolvimento

Diversas linguagens possuem **ambiente** de **desenvolvimento associado** a elas:

Exemplos:

- Linguagem Basic possui o MS Visual Basic;
- Object Pascal possui o Delphi.

Outras linguagens estão **desvinculadas** de ambiente de desenvolvimento:

Exemplos:

- Java
- PHP
- C/C++

As principais vantagens da linguagem ser **desvinculada** de ambiente são:

- Não há **dependência** de um único **fornecedor** de ambiente de desenvolvimento.
- Maior **controle sobre** o **código** produzido.
- A **concorrência** entre **fornecedores** gera melhorias nas ferramentas.

A desvinculação da linguagem de um ambiente de desenvolvimento "socializa" a linguagem de programação, permitindo que o desenvolvedor possa escolher se quer ou não comprar um ambiente de desenvolvimento.

Apesar das vantagens de uma linguagem ser desvinculada de uma IDE, um ambiente de desenvolvimento proporciona maior facilidade para a escrita do código, depuração e construção de interfaces gráficas.

Alguns ambientes de desenvolvimento Java são:

- JBuilder (www.borland.com)
- NetBeans (http://www.netbeans.org)
- Java Studio Creator (www.sun.com)
- JEdit (www.jedit.org)
- IBM Websphere Studio Application Developer (WSAD) (www.ibm.com)
- Eclipse (www.eclipse.org)
- JDeveloper (www.oracle.com)

Nota: Todos os ambientes de desenvolvimento dependem do J2DKSE instalado.

Anotações

1.4 Linguagens interpretadas x linguagens compiladas

Linguagens de programação são comumente divididas em linguagens interpretadas e compiladas.

Interpretadas

Como o próprio nome diz, são interpretadas linha a linha em tempo de execução.

Exemplos:

- Perl;
- ASP (Active Server Pages);
- JavaScript;
- Basic.

Compiladas

O compilador traduz o programa fonte apenas uma vez para linguagem compilada (executável) não importando quantas vezes o programa irá ser executado. No processo de compilação o código fonte é submetido á análise sintática, léxica e semântica. Caso algum erro seja encontrado o arquivo "executável" pela Virtual Machine (.class) não é gerado, e os erros são apontados pelo compilador.

Muitos erros são eliminados durante o processo de compilação.

Exemplos:

- caracteres inválidos.
- nomes de variáveis, métodos, classes inválidas.
- seqüência de comandos inválidos "{" sem "}" correspondente.
- tipos e quantidade de parâmetros, retorno de funções, etc...

Erros lógicos não são capturados no processo de compilação.

Exemplos:

- Divisão por zero;
- Operadores logicamente errados;
- Operações com objetos não construídos (nulos).

Exemplos:

- Pascal
- C/C++
- Java.

2 Plataforma Java

2.1 História do Java

A Sun Microsystems, acreditando no crescimento do uso de pequenos dispositivos eletrônicos destinados ao consumidor final, financiou uma pesquisa interna de codinome Projeto Green em 1991, que resultou na criação de um equipamento chamado Start Seven (*7) e uma linguagem baseada em C e C++ que seu criador James Gosling, batizou de Oak (carvalho) em homenagem a uma árvore que podia ser vista através da janela de seu escritório na Sun.

Algumas características do Start Seven eram:

- Monitor LCD 5” colorido e touchscreen
- Interface PCMCIA – Wireless
- Versão de UNIX rodando em menos de 1MB
- Linguagem segura, robusta, multi-plataforma, com threads, bibliot coleta automática de lixo, distribuída entre outras caracterísitcas.
- Controle remoto
- Permitia distribuição de objetos em uma rede sem fio

Obs: O Duke, maskote do Java até hoje foi criado juntamente com o projeto Star Seven!

Mais tarde descobriu-se que já havia uma linguagem chamada Oak. Quando uma equipe da Sun visitou uma cafeteria local, o nome Java (cidade de origem de um café importado) foi sugerido e aceito.

O mercado para pequenos dispositivos eletrônicos cresceu menos do que o esperado pela Sun Microsystems mas com a explosão da popularidade da World Wide Web em 1993 a Sun previu o imediato potencial do Java para o desenvolvimento de conteúdo dinâmico para a Web.

Por causa do fenomenal interesse pela WWW, a apresentação formal do Java em uma conferência causou interesse imediato na comunidade comercial.

Atualmente Java é muito utilizado para o desenvolvimento de conteúdo dinâmico para a Web, aplicativos corporativos, comerciais, financeiros, aplicativos de alta capacidade de processamento em servidores, aplicativos para pequenos dispositivos como celulares, PDA’s, etc...

Anotações

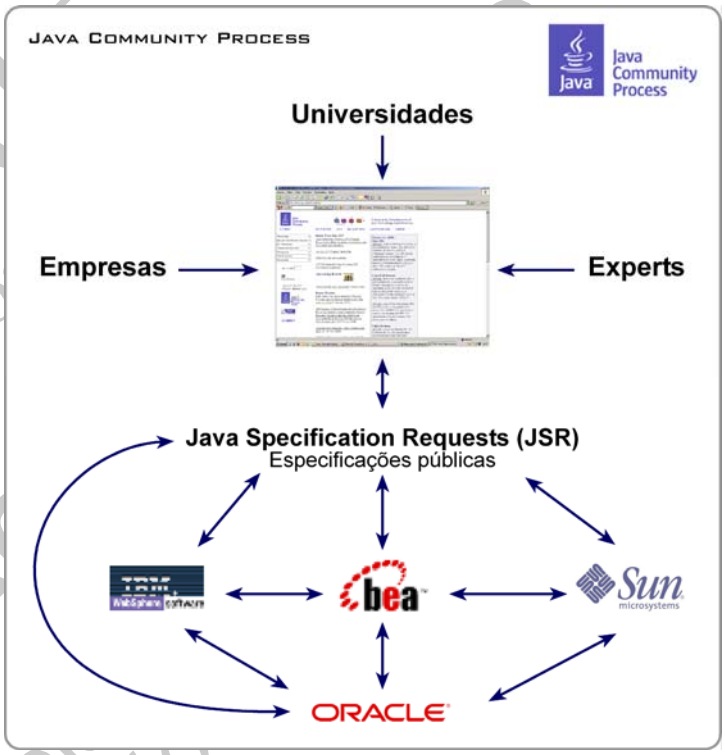
Histórico resumido da tecnologia Java

23 de maio, 1995	Lançamento da tecnologia Java
23 de janeiro, 1996	Lançamento do JDK 1.0
29 de maio, 1996	1º JavaOne
Setembro, 1996	83000 páginas web utilizando Java
09 de dezembro,1996	Lançamento JDK1.1 beta
04 de março, 1997	Lançamento Java Web Server beta e Java Servlet Developers Kit
02 de abril, 1997	JavaOne atinge o número de 10.000 inscritos tornando-se a maior conferência de desenvolvedores do mundo. Sun anuncia a tecnologia Enterprise Java Beans (EJB)
Março, 1998	Lançamento do JFC(Java Foundation Classes) / “Projeto Swing”
24 e março, 1998	JavaOne atinge o número de 15.000 inscritos
08 de dezembro, 1998	Formalização do Java Community Process (JCP)
25 de janeiro, 1999	Anuncio da tecnologia JINI
02 de junho, 1999	Lançamento de Java Server Pages (JSP)
15 de junho, 1999	JavaOne atinge 20.000 inscritos Sun anuncia três edições da plataforma Java: J2SE, J2EE, J2ME
25 de agosto, 1999	Lançamento J2SE 1.3 beta
30 de setembro, 1999	Lançamento J2EE beta
26 de maio, 2000	Existem mais de 400 grupos de usuários Java (JUG)
Junho 2001	Lançamento do JDK1.4

2.2 Mitos da linguagem

Java é da SUN?

A especificação Java foi criada pela SUN, no entanto, a linguagem é mantida pelo Java Community Process (JCP) que reúne Java experts, empresas e universidades que através de processos democráticos definem a evolução da linguagem.



Java é uma linguagem direcionada para Web?

Java não é apenas uma linguagem direcionada para Web, apesar de ser atualmente bastante conhecida e divulgada por seus "dotes" para desenvolvimento de aplicações Web, Java é uma linguagem completa como: C++, Pascal e Basic.

Java é igual a JavaScript?

Não. Java é compilada e JavaScript é interpretada pelo interpretador contido no browser.
Java é uma criação da SUN e JavaScript é uma criação da Netscape. A linguagem JavaScript originalmente chamava-se LiveScript, mas um acordo entre a Sun e a Netscape acabou fazendo com que LiveScript viesse a se chamar JavaScript.

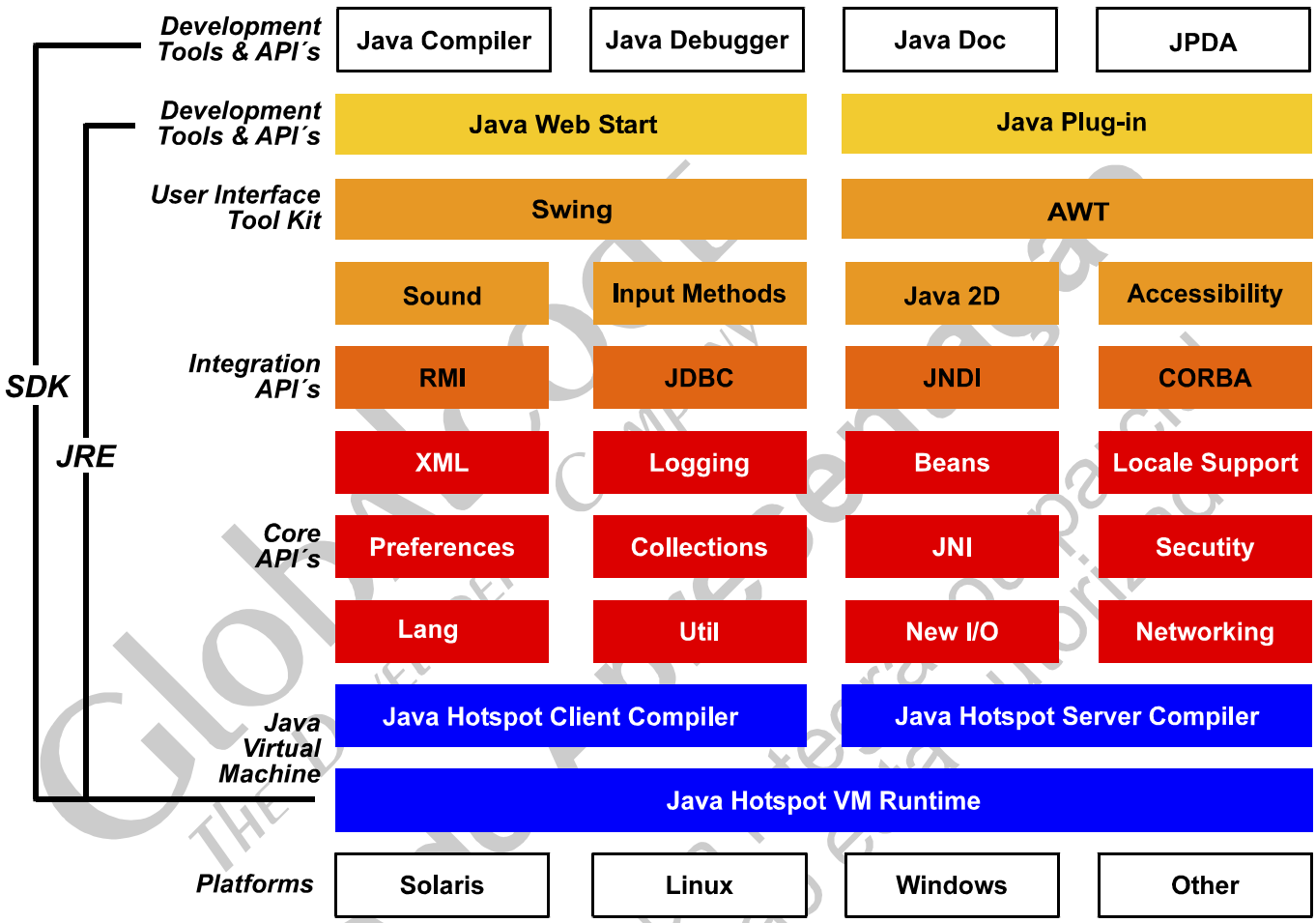
Java é lento?

Java, como aplicação stand alone, é mais lento que uma linguagem compilada com código nativo (por exemplo, linguagem C), pois para ser portável não interage diretamente com o servidor gráfico do sistema operacional. No entanto, a afirmação de que "Java é lento" é completamente falsa para softwares distribuídos (em servidores), onde bibliotecas gráficas não são necessárias para gerar respostas aos usuários. (Servlet, JSP, RMI).

Anotações

2.3 Java 2 Standard Edition (J2SE)

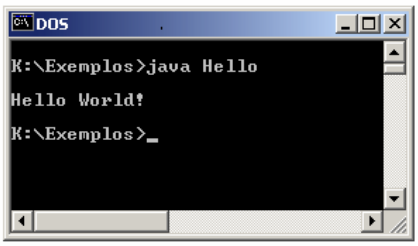
É a especificação do Java que contém APIs com as funções básicas do Java como I/O, multithread, network, conectividade com bancos de dados entre outras mais.



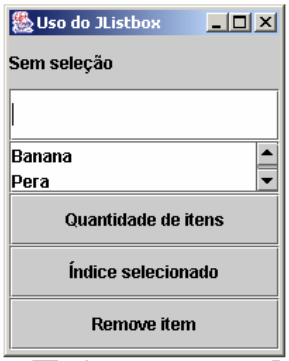
J2SE é composto por classes para atender as seguintes necessidades:

- Classes essenciais
- Applets
- Networking
- Internacionalização
- Segurança
- Serialização de objetos
- Java Database Connectivity (JDBC)
- Utilitários

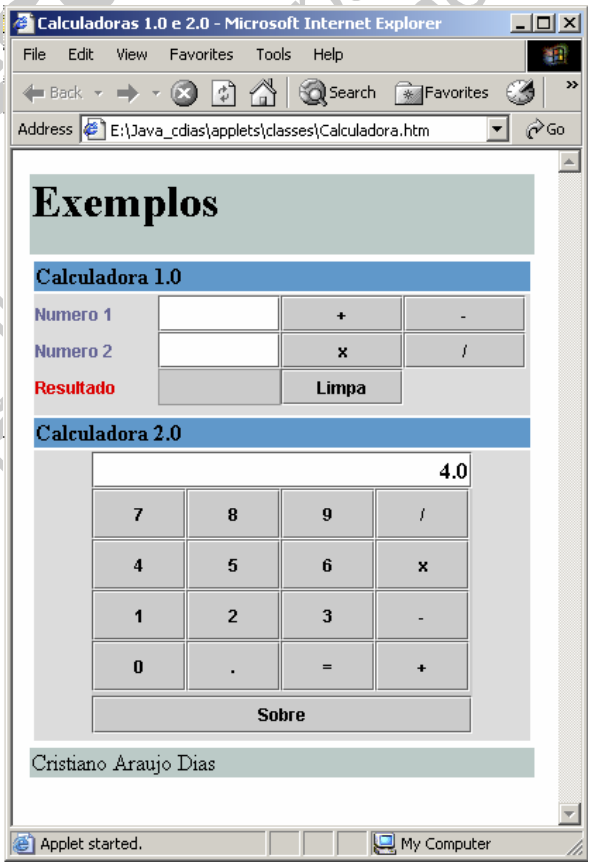
Veja alguns exemplos de aplicações implementadas utilizando a plataforma J2SE:



Console



AWT / Swing

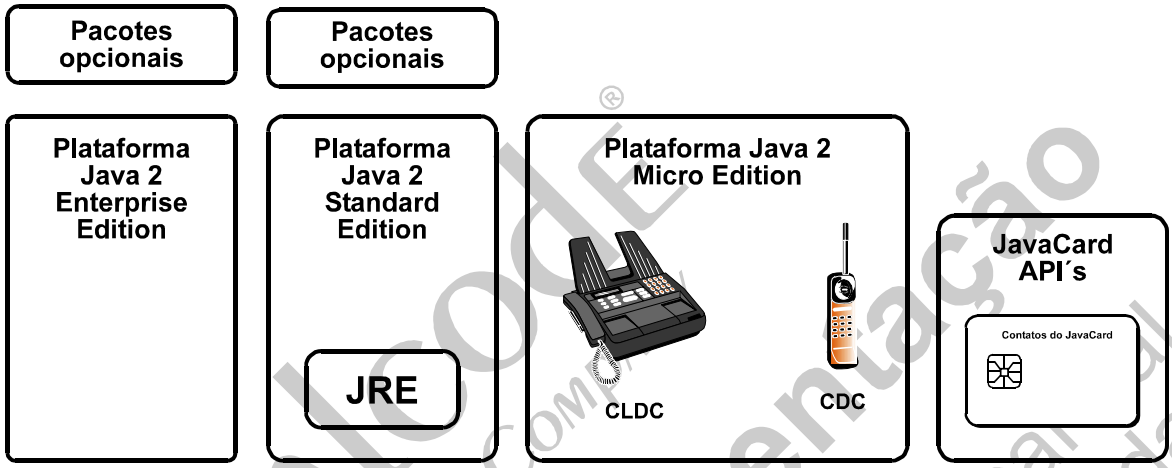


Applet

Anotações

2.4 Java 2 Micro Edition (J2ME)

É a especificação Java que contém APIs com funcionalidades para desenvolvimento de aplicações para pequenos dispositivos como: agendas eletrônicas, telefones celular, palmtop e aparelhos eletrônicos em geral que possuam uma KVM (Máquina Virtual para pequenos dispositivos).



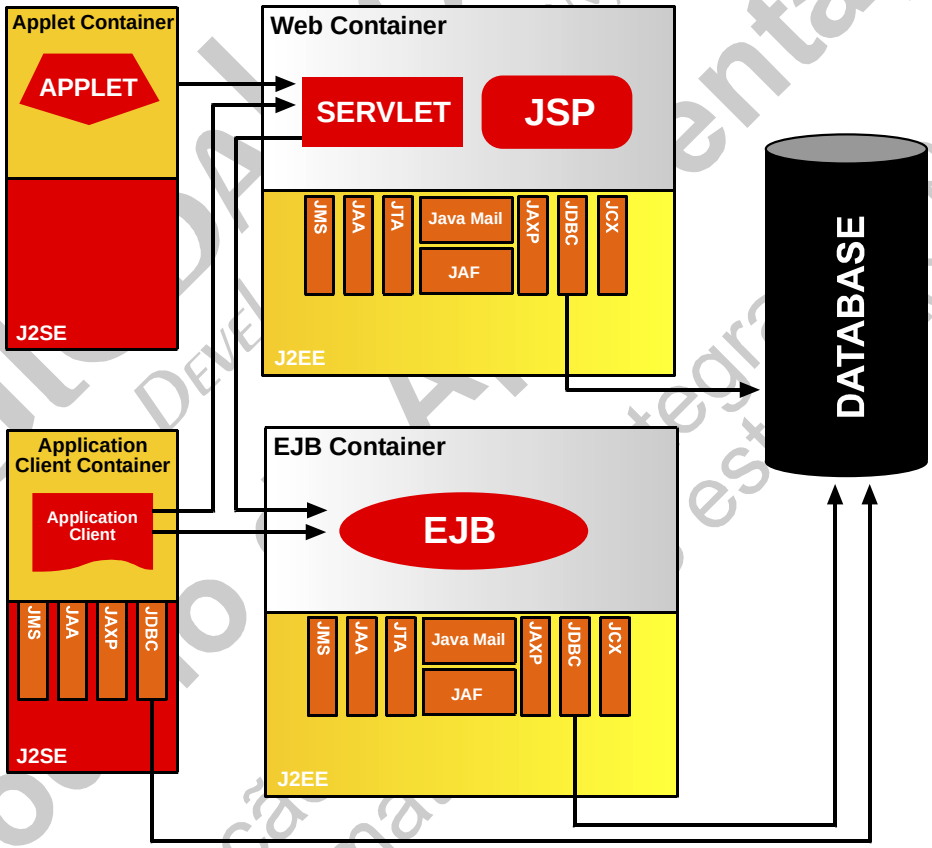
Emulador para desenlvolvimento de aplicações J2ME, desta forma você não precisa ter um celular que roda J2ME para desenvoler e testar suas aplicações.

2.5 Java 2 Enterprise Edition (J2EE)

É a especificação Java que contém APIs com funcionalidades específicas para o desenvolvimento de aplicações para servidores tais como, Servlets, JSP, EJB e JMS.

O J2EE é uma extensão ao J2SE e acompanha um servidor Web, um servidor de componentes transacionais de negócio, servidor de banco de dados, e um servidor de filas de mensagens.

Veja abaixo uma ilustração da arquitetura J2EE completa:



Anotações