

## Desenvolvendo Aplicações com Java Card

Wilton Oliveira Ferreira<sup>1,2</sup>, Ricardo Batista Rodrigues<sup>2</sup>, Julio Cezar Damasceno<sup>3</sup>,  
Josino Rodrigues Neto<sup>2</sup>

<sup>1</sup>Centro de Estudos e Sistemas Avançados do Recife (C.E.S.A.R)  
Rua Bione, 220 - Cais do Apolo - Bairro do Recife 50.030-390 - Recife, PE - Brazil

<sup>2</sup>Centro de Informática, Universidade Federal de Pernambuco (UFPE)  
Av. Jornalista Anibal Fernandes, s/n - Cidade Universitária - 50.740-560 - Recife, PE -  
Brasil

<sup>3</sup>Departamento de Estatística e Informática - Universidade Federal Rural de  
Pernambuco (UFRPE)  
Rua Dom Manoel de Medeiros, s/n, Dois Irmãos 52.171-900 - Recife, PE - Brasil  
{wof,rbr,jcd,jrn4}@cin.ufpe.br

**Resumo.** *Este trabalho tem como objetivo apresentar o conceito de Java Card e os principais métodos, técnicas e ferramentas utilizadas na atualidade por desenvolvedores. A ideia é apresentar a tecnologia e detalhar o processo de como iniciar uma aplicação, apresentar técnicas e ferramentas que podem maximizar as chances da criação de aplicações bem sucedidas e as principais vantagens da tecnologia. Por fim, mostrar o passo a passo do desenvolvimento de uma pequena aplicação, para que seja colocado em prática as técnicas apresentadas.*

**Abstract.** *This paper aims to introduce the concept of Java Card and the main methods, techniques and tools used by developers today. The idea is to present the technology and detail the process of how to start an application, present techniques and tools that can maximize the chances of creating successful applications and the main advantages of the technology. Finally, to show the step by step development of a small application, to be put into practice the techniques presented.*

### 1. Introdução

O foco do mercado nos últimos anos tem sido a emissão de Smart Cards para diversas aplicações. Porém, essa tendência vem se alterando para um maior esforço de melhor utilização dos cartões já existentes no mercado, visto que é possível fazer atualizações no Smart Card sem que seja necessário o retorno ao fabricante ou a emissão de mais cartões. Dessa forma, os setores público, financeiro e de telecomunicações, que são os mais presentes nesse mercado, serão os que mais investirão em novas formas de exploração do Smart Card [Queiroz 2013].

Em novembro de 1996, um grupo de engenheiros da Schlumberger introduziram uma API Java Card para desenvolvimento em seus Smart Cards, propondo também um rascunho inicial para a especificação Java Card. Meses depois Bull, Gemplus e

Schlumberger se unem para fundar o Java Card, formam um consórcio industrial criado para identificar e resolver questões relacionadas a tecnologia Java Card. Finalmente a Sun Microsystems dá suporte oficialmente a especificação Java Card 1.0 que não passava de uma API e não uma Plataforma extensível. Em novembro de 1997, a Sun Microsystems anuncia a especificação Java Card 2.0 com diferenças significativas como uma maneira de se escrever applets utilizando a Orientação a Objetos e especifica mais o Ambiente em Tempo de Execução [Massao 2013].

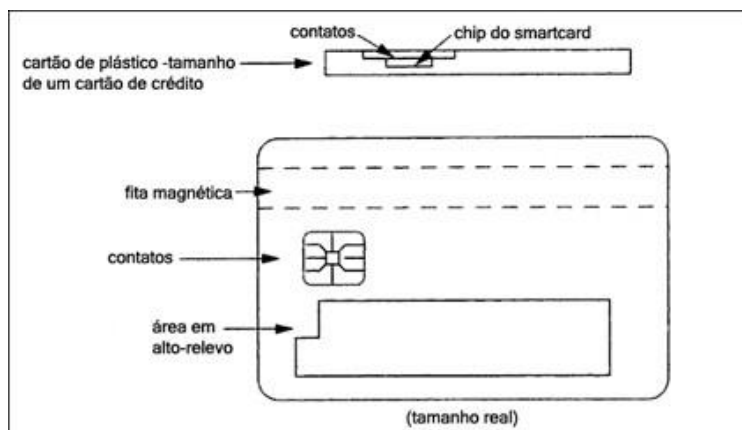
Em Março de 1999 é anunciada a especificação Java Card 2.1, pode ser considerada uma Plataforma completa pois é especificado o Java Card API, JCRE (Java Card Runtime Environment) e o JCVM (Java Card Virtual Machine). Em maio de 2000 é anunciada a especificação Java Card 2.1.1. Em setembro de 2002 é anunciada a especificação Java Card 2.2 o qual introduz a característica RMI. Em outubro de 2003 é anunciada a especificação Java Card 2.2.1. Em março de 2006 é anunciada a especificação Java Card 2.2.2, essa é a última versão 2 e introduziu interfaces para desenvolvimento de Biometria. Em maio de 2009 é anunciada a especificação Java Card 3.0.1 não mais pela antiga Sun Microsystems mais pela então Oracle [Massao 2013].

Smart Cards representam uma das menores plataformas computacionais do mundo, isso quer dizer, são limitados em recursos de processamento e memória. No início, desenvolver para Smart Cards era um processo lento e difícil, pois embora os cartões possuíssem padrões no tamanho e no protocolo de comunicação, cada um funcionava de maneira diferente de fabricante para fabricante [Massao 2013].

As ferramentas de desenvolvimento eram fornecidas pelos fabricantes dos Chips utilizando como base a linguagem Assembler, os SDKs (Software Development Kit) eram dedicados a apenas um tipo ou família de chips, ou seja, o desenvolvimento de uma aplicação era específico para cada fabricante e determinado Chip [Oracle 2013].

## 2. Smart Card

O Smart Card é um cartão de plástico com um chip acoplado que armazena e processa dados. Essa tecnologia é usada em diversas áreas, dentre os quais podem-se citar aplicações na área de saúde, bancária, de entretenimento e de transportes. Tais aplicações se utilizam da segurança que os Smart Cards providenciam, sendo assim um fator chave para sua utilização. Para tal, utiliza-se a tecnologia Java Card, que permite que pequenas aplicações, chamadas applets, sejam empregadas em Smart Cards, utilizando a plataforma Java [Massao 2013].



### Figura 1. Esquema simples de um Smart Card [Queiroz 2013]

Um benefício para sua ampla utilização é seu poder computacional embutido, além de segurança, portabilidade e facilidade de uso que esta tecnologia oferece. Hoje, nos Estados Unidos, Smart Cards são utilizados em grande parte do tempo por seus consumidores para variadas atividades, como visitar bibliotecas, assistir filmes no cinema ou fazer compras, provando-se altamente presente na vida cotidiana atual [Massao 2013].

#### 2.1. Conceitos Fundamentais de Java Card

Java Card é uma Interface de Programação de Aplicativos API desenvolvida e distribuída pela Oracle que permite que aplicativos Java, chamados applets, seja executado nos chips de Smart Cards, acessando seus sistemas de arquivos e microprocessador [Oracle 2013].

A Tecnologia Java Card oferece uma maneira prática de vencer esses obstáculos, ocultando as implementações de Hardware do cartão. Ou seja, desde que o cartão implemente a especificações Java Card, não importa o tipo de processador, ou se é arquitetura Cisc ou Risc, ou a capacidade de memória. Essencialmente a Tecnologia Java Card define a segurança, portabilidade e multiaplicativos que incorporam as várias vantagens da Linguagem Java, como por exemplo, Write once run anywhere (escreva uma vez e execute em qualquer lugar) [Massao 2013].

A segurança é uma preocupação primordial, uma das características, por exemplo, é o nível de acesso a todos os métodos e variáveis que são estritamente controlados, e não existe modo de forjar ponteiros para ativar programas maliciosos na memória. Java Cards permite que sejam instalados vários Applets e cada Applet dentro do cartão é isolado por um firewall, isso evita que uma aplicação hostil comprometa o sistema [Queiroz 2013]. A seguir apresentamos as principais características da API Java Card:

- **Ambiente com baixo poder computacional:** executar applets Java em chips de baixo poder computacional como o de Smart Cards é um desafio. Sendo assim, a tecnologia Java Card implementa a máquina virtual Java de forma dividida entre o sistema do cartão e o sistema operacional do terminal. No cartão, são implementados o modelo de execução, comunicação, segurança e o gerenciamento da memória. Processos como carregamento de classes, verificação de bytecode, e etc. são executados no sistema externo ao cartão.
- **Programação de alto nível:** outra característica implementada pela API é a separação entre o sistema e as aplicações. Assim como a linguagem Java para desktops, a API permite o gerenciamento dos dados através de requerimentos bem definidos e de alto nível.
- **Portabilidade:** como aplicativos em Java para desktops, os applets Java Card são independentes de plataforma. Dessa forma, applets iguais podem ser usados em Smart Cards com capacidade de memória e processamento diferente, além de estruturas e arquiteturas diversas. Isso não se limita a cartões, mas também a ambientes em que há pequeno poder de processamento.
- **Multifuncionalidade:** cartões com a tecnologia Java Card podem conter mais de um applet. A API Java Card gerencia o compartilhamento de dados entre

applets diferentes, protegendo dados internos aos applets e sincronizando o acesso a dados compartilhados.

### 3. Arquitetura Java Card

A Tecnologia Java Card oferece um ambiente seguro para as aplicações que rodam em smart card e outros dispositivos com memória e capacidade de processamento limitada.

Os smart card permitem a implantação de vários aplicativos em um único dispositivo, geralmente os aplicativos são implementados em tempo de fabricação. Porém, novos aplicativos podem ser adicionados a ele mesmo depois de estar sendo utilizado pelo usuário final. Java Card também inclui um conjunto de ferramentas únicas para o desenvolvimento de novos produtos. Por causa das limitações de hardware a plataforma Java Card é dividida em duas partes, sendo uma, presente no smart card e a outra no computador convencional, utilizado para o desenvolvimento dos aplicativos Java Card [Rankl and Effing 2002]. A plataforma completa de desenvolvimento do Java Card, consiste das partes a seguir devidamente documentada em especificações.

- A especificação JCVM (Java Card Virtual Machine), define um subconjunto da linguagem de programação Java e definição também a JCVM adequada para o desenvolvimento de aplicação smart card [Oracle 2013].
- A especificação JCRE (Java Card Runtime Environment), descreve precisamente como deve comportar-se as aplicações Java Card em tempo de execução. Nesta documentação também incluem gerenciamento de memória, gerenciamento dos Applets e também de outros recursos em tempo de execução smart card [Oracle 2013].

#### 3.1. API Java Card (Subconjunto da Linguagem Java)

A tecnologia Java Card suporta apenas um subconjunto das funcionalidades da linguagem Java. Do mesmo modo, a JCVM (Java Card Virtual Machine) suporta apenas os recursos que são necessários para o subconjunto da linguagem Java Card. Quaisquer características da linguagem Java não suportados pelas JCVMS e utilizadas em um Applet são detectados pelo conversor no momento da conversão.

A plataforma Java Card apresenta um subconjunto seletivo das funcionalidades da linguagem Java convencional. Esta característica se dá em razão das limitações dos dispositivos que utiliza esta plataforma. As funcionalidades selecionadas para compor a API Java Card têm como principal característica, o tamanho reduzido de ocupação de memória. A Tabela 1 destaca as principais funcionalidades suportadas, e as notáveis funcionalidades não suportadas.

**Tabela 1: Funcionalidades suportadas e não suportadas do Java Card**

<b>Funcionalidades Suportadas</b>	<b>Funcionalidades não Suportadas</b>
<ul style="list-style-type: none"><li>•Pequenos tipos de dados primitivos: boolean, byte, short</li><li>•Array unidimensional</li><li>•Pacotes java, classes, interfaces e</li></ul>	<ul style="list-style-type: none"><li>•Grandes tipos de dados primitivos: long, double, float</li><li>•Caracteres e Strings</li><li>•Carregamento de Classes dinâmicas</li></ul>

<p>tratamento de exceções</p> <ul style="list-style-type: none"><li>•Funcionalidades da orientação a objeto: herança, métodos virtuais, sobrecarga e criação dinâmica do objeto, escopo de acesso, e regras de vinculo</li><li>•Tipo inteiro de 32 bits é suportado apenas em algumas versões da API.</li></ul>	<ul style="list-style-type: none"><li>•Gerenciamento de Segurança</li><li>•Coletor de lixo e finalização de objetos</li><li>•Threads</li><li>•Serialização de objeto</li><li>•Serialização de objeto</li><li>•Clonagem de objeto.</li></ul>
---	---

### 3.2. JCVM (Java Card Virtual Machine)

A JCVM (Java Card Virtual Machine) apresenta os mesmos conceitos da JVM (Java Virtual Machine), com apenas algumas diferenças. A primeira diferença entre estas duas plataformas é caracterizada pelo fato da JCVM ser dividida em duas partes separadas, como mostra a Figura 2 [Chen 2000]. A parte on-card da JCVM incluem o interpretador de bytecode Java Card. A parte off-card incluem um conjunto de ferramentas para analisar o código java, converter e encapsular as classes em um pacote específico denominado CAP (Converter Applet). O pacote CAP pode ser implantado no smart card ou qualquer outro dispositivo que suporte Java Card, e as classes presentes neste pacote são interpretadas pelo interpretador on-card.

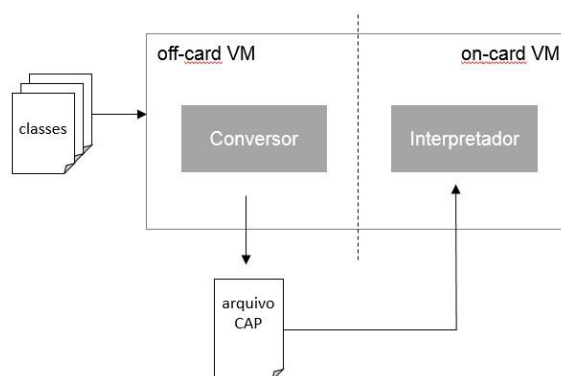


Figura 2. Java Card Virtual Machine

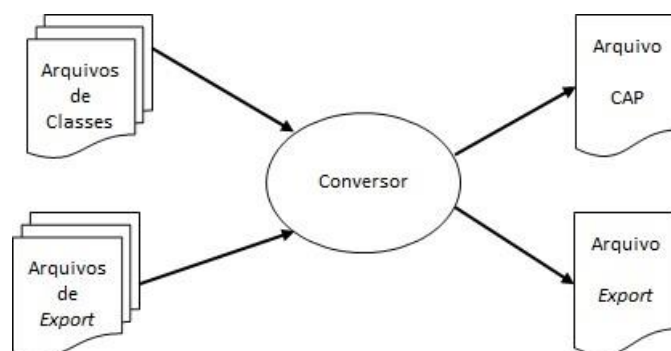
#### 3.2.1. Arquivo CAP e Arquivo Export

A tecnologia Java Card introduz dois novos formatos de arquivos binários, que permitem o desenvolvimento distribuído, independente da plataforma de execução. O arquivo CAP (Converter Applet) contém a representação binária executável das classes em um pacote Java. O arquivo CAP é um arquivo JAR (Java Archive) que contém um conjunto de componentes também encapsulado de forma individual em arquivos JAR [Chen 2000].

Cada componente descreve um aspecto do conteúdo do arquivo CAP, tais como informações das classes, bytecodes executáveis, links de informações, verificação de informações e assim por diante. O formato de arquivo CAP é otimizado para pequenos dispositivos, usando estruturas de dados compactos. Além do arquivo CAP também é gerado o arquivo Export, que auxilia a criação do próprio CAP.

Os arquivos Exports é utilizado na fase de pré-compilação do projeto Java Card. Esta pré-compilação utiliza recursos que não fazem parte do smart card, este conjunto de recurso é denominado artefatos off card. Os artefatos off card são auxiliares, e sua utilização só ocorre na fase de formação do CAP. O Export é composto pelas várias bibliotecas necessárias para a compilação inicial, e fase de montagem do arquivo CAP. Este arquivo poderá ser descartado imediatamente após realizada a conversão (compilação e empacotamento) do projeto Java Card em pacote CAP.

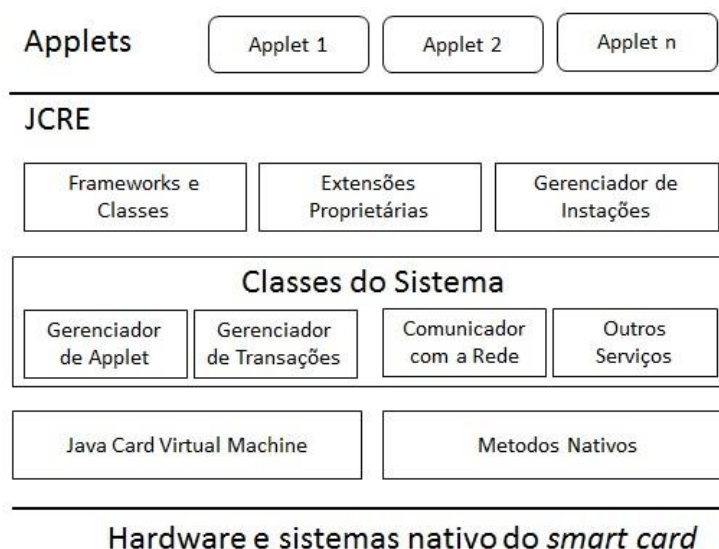
Ao contrário da máquina virtual Java, que processa uma classe de cada vez, a unidade de conversão processa o pacote por completo. Os arquivos das classes são produzidos por um compilador Java a partir do código fonte. Então, o conversor pré-processa todos os arquivos de classe que compõem um pacote Java e converte o pacote em um arquivo CAP. Na Figura 3 é mostrado a estrutura completa do processo de conversão.



**Figura 3. Conversor de pacotes**

### 3.3. Java Card Runtime Environment (JCRE)

O JCRE (Java Card Runtime Environment) consiste em um componente Java Card que executa dentro do smart card. O JCRE é responsável pela gestão dos recursos do smart card, tais como comunicações com a rede, a execução dos Applet e dos sistemas de segurança do/para o Applet. Este componente é essencial para a solução smart card. A Figura 4 apresenta a posição onde se localiza o JCRE dentro da arquitetura da solução smart card. Como pode ser observado, o JCRE oferece os vários serviços de Frameworks, implementação de especificações proprietária e ferramentas para gerir a implantação dos Applets no smart card.



**Figura 4. Arquitetura Smart Card**

Como pode ser observado na Figura 4, a camada superior à camada de hardware é composta pela JCVM e por métodos nativos a qual atua como um mediador entre o hardware e as camadas de software superiores a ela. Logo, imediatamente superior, temos as Classes do Sistema que mais atua em prol do apoio à infraestrutura como um todo, dentre os vários serviços desta camada, se destacam: gerenciador de applet, gerenciador de transações e gerenciador das comunicações de redes. Acima da camada de Classes do Sistema está localizada a JCRE (Java Card Runtime Environment) que por sua vez conta com alguns pacotes específico que são utilizados para auxiliar a execução dos Applets que está na camada imediatamente superior.

## 4. Desenvolvendo uma Aplicação com Java Card

O desenvolvimento de uma aplicação Java Card precede os mesmos cuidados inerentes a programação na linguagem Java convencional. A seguir serão apresentados algumas das necessidades e exigências desta tecnologia.

### 4.1. Configuração do Ambiente de Desenvolvimento

O ambiente de desenvolvimento destinado à tecnologia Java Card pode ser montado utilizando ferramentas como eclipse e plug-ins específicos para tal [Eclipse 2013][Youssef 2013]. Um plug-in para o desenvolvimento Java Card bastante eficiente e o Eclipse JCDE (Eclipse Java Card Development Environment) [Youssef 2013]. Observando apenas a necessidade de além do JDK (Java Development Kit) e JRE (Java Runtime Environment) também é necessário o JCDK (Java Card Development Kit).

Durante o processo de configuração do ambiente, deve ser observado a compatibilidade do smart card com a versão do JCDK, a ISO/IEC (International Organization for Standardization / International Electrotechnical Commission) cuida da regulamentação dos smart cards em todos seus aspectos. Que vai desde a estrutura física do dispositivo até os padrões de comunicação incluindo o suporte ao Java Card. Já o responsável por manter a compatibilidade do Java Card com os padrões ISO/IEC fica a par da Oracle [Oracle 2013] atual mantenedora da tecnologia.

## 4.2. Emulação de Smart Card

O desenvolvimento dos aplicativos Java Card é facilitado pela funcionalidade padrões da ferramenta Eclipse [Eclipse 2013]. Desta forma, desenvolver os Applets para Java Card será algo trivial do ponto de vista de codificação. Do ponto de vista da semântica da linguagem, deve ser observado as limitações quanto aos tipos de dados suportados. Há frequentes casos que, por exemplo, a compilação é permitida porém a execução dos testes geram uma exceção. Detectar uma exceção em um Applet Java Card é extremamente complicado, isto, porque, quando o mesmo está sendo executado diretamente na mídia smart card, não é possível analisar o log muito menos visualizar a execução. Para estes casos só é possível contar com as várias mensagens contendo códigos de erros que podem ser facilmente interpretados com base na ISO 7816 [ISO/IEC 2013].

Durante o processo de desenvolvimento, os códigos gerados podem ser testados utilizando ferramentas que emulam o ambiente smart card. Desta forma, os resultados obtidos a partir da emulação do smart card aproxima, em muito, as características do hardware real. Além de manter a facilidade na ausência da necessidade da implantação do Applet em uma mídia smart card, a ferramenta de emulação também permite facilmente a depuração do código, o que reduz a perda de tempo com frequentes testes durante o desenvolvimento.

## 5. Conclusão

O estudo apresentado teve como desafio apresentar as características básicas da tecnologia Java Card. Bem como instruir aos interessados, como realizar pesquisas envolvendo esta tecnologia. Foram apresentadas as principais características da arquitetura e também algumas aplicações que admitem o uso de tal tecnologia. Resumidamente, pode ser dito que os padrões de segurança oferecido pela plataforma smart card/Java Card é extremamente robusta no que se refere a capacidade de proteção. Todas as funcionalidades de segurança do smart card podem ser estendidas através do uso de applets. Alguns casos de sucesso, bem conhecidos, são os cartões de créditos e tokens utilizado em operações financeiras. Além disso, esta tecnologia permite que um smart card possa ser atualizado após ser liberado pelo fabricante. Quando modificamos uma senha de um cartão bancário por exemplo, o fazemos alterando informações interna do smart card.

A tecnologia Java Card ainda é recente, logo ainda existem muitos desafios de pesquisa a serem explorados. Ainda existe uma certa dificuldade enfrentada para os pesquisadores/desenvolvedores que estão se envolvendo com tal tecnologia no que se refere a conteúdo de apoio. Algumas referências como [Chen 2000], [Finkenzeller 2003], [Rankl and Effing 2002], [Haghiri and Tarantino 2002], [Ferrari et al. 1998], podem ajudar a quem de fato pretende se empenhar nos estudos do Java Card para smart cards.

## Referências

- Chen, Z. (2000). Java Card Technology for Smart CCard – Architecture and Programmer’s Guide. II. Addison-Wesley, first edition edition.
- Eclipse (2013). Eclipse.



- Ferrari, J., Mackinnon, R., Poh, S., and Yatawara, L. (1998). Smart Cards: A Case Study. IBM, first edition.
- Finkenzeller, K. (2003). RFID HANDBOOK - Fundamentos and Applications in Contactless Smart Cards and Identification. WILEY, second edition.
- Haghir, Y. and Tarantino, T. (2002). Smart Card Manufacturing - A Practical Guide. WILEY.
- ISO/IEC (2013). Iso/iec 7816 - cards and personal identification.
- Massao, R. (2013). Planet smart cards.
- Oracle (2013). Oracle – java card technology. Technical report, Oracle.
- Queiroz, A. R. (2013). Smar card e java card.
- Rankl, W. and Effing, W. (2002). Smart Card Handbook. Wiley, third edition edition.
- Youssef, A. (2013). Eclipsejcdc user guide.