

# **IDedu: Proposta de um Cartão de Identificação Acadêmico baseado no padrão ICAO 9303**

**Jean Everson Martina, Dr.**

**Felipe Coral Sasso**

## **Relatório Final**

### **Contextualização**

A criação de uma federação para o agregamento e distribuição de dados de autenticação de usuários é hoje uma realidade no Brasil. A Comunidade Acadêmica Federada (CAFe) é um exemplo no âmbito acadêmico de tal federação onde as instituições participantes compartilham além dos dados de usuário para a autenticação, também serviços que podem ser usados racionalmente pelos usuários da federação.

Um grande exemplo da abrangência e importância do compartilhamento de dados de usuários dentro de federações é o projeto EDUROAM. As entidades participantes da Federação CAFe também possuem o acesso a rede de conectividade EDUROAM. O EDUROAM é uma rede mundial de compartilhamento de informações de usuários para o provimento de acesso a redes sem fio através das redes das instituições participantes. Com ele é possível que um usuário de uma instituição no sul do país possa simplesmente chegar a uma outra instituição no Norte, ou mesmo fora do país e ter acesso a Internet com seu usuário e senha providos pela sua instituição de origem. Essa é uma realidade positiva do uso de federações em nível mundial.

No entanto as federações em geral ainda possuem espaço para a inovação no uso de suas credências, haja vistas algumas limitações que listamos a seguir:

- As federações no modelo usual requerem que, tanto os recursos, quanto o provimento de identidades seja feito de forma online. Desta forma, tanto o provedor de identidade da instituição de origem, quanto o provedor de serviços da instituição destino devem de alguma forma se comunicar em tempo real para que o usuário possa acessar determinado serviço. Algumas aplicações tais como o empréstimo de livros ou o acesso a determinados ambiente de uso compartilhado dentro de uma universidade podem não requerer tal checagem.
- As identidades providas pela federação requerem o uso de computadores para a sua verificação. Desta forma, o uso de credenciais de identificação fica restrito ao âmbito dos sistemas computacionais, tornando seu uso por agentes humanos difícil ou quase impossível. Claramente, o uso de credenciais verificáveis por agente humanos pode ajudar em vários cenários onde a identificação é corriqueira e necessária, inclusive por agentes que não fazem parte da federação, tal como as bilheterias de cinemas e teatros.

- Alguns dados são considerados de uso restrito e privado, não podendo ser compartilhados pelas instituições de origem através do seu provedor de identidades. Desta forma, não podemos trabalhar com tais dados sob pena de a federação sofrer sanções legais. Um exemplo de uso deste tipo de informação no âmbito acadêmico é o de informações biométricas para o processo de autenticação. Ele é muito útil como segundo fator de autenticação, mas sua coleta, armazenamento e transmissão são atividades muito sensíveis à problemas de privacidade no ambiente federado.

Para responder aos problemas encontrados na federação CAFé expostos acima, esta proposta gostaria de fazer um estudo de caso para propor uma identidade acadêmica federada baseada nos documento ICAO 9303. A ideia é fazermos um estudo para a adaptação dos documentos ICAO 9303 para o ambiente acadêmico e da federação CAFé.

Como aplicabilidade no âmbito acadêmico podemos vislumbrar a integração de um processo eficiente, amplamente disponível e inequívoco de autenticação nos mais variados cenários. Podemos claramente citar os controles de acesso a recursos institucionais ou da federação de autenticação à qual determinada instituição pertence. Também podemos fortalecer os processos de autenticação na conferência de presença dos alunos e na aplicação de provas.

No âmbito das tecnologias já conhecidas e dominadas pelo GT-Gid da RNP, podemos ainda fazer uso desta nova identidade acadêmica para a assinatura digital e para a instanciação de certificados de atributos de interesse das instituições parceiras nos referidos cartões. Não é difícil vislumbrar a emissão de certificados de atributos para Professores, Técnicos Administrativos e Alunos que comprovem seu vínculo com a instituição e contribuam na sua identificação e autorização para o uso de serviços. Desta forma inclusive poderíamos vincular a nova identidade acadêmica federada com as novas tecnologias de identificação civil que estão sendo colocadas em prática pelo governo federal.

Pela motivação acima fica claro que mesmo com a experiência adquirida no projeto submetido ao PGId 2013 ainda é necessário que se estude a fundo as tecnologias utilizada nos cartões no que tange as suas capacidades de hardware e software. O objetivo deste projeto é então o estudo e a implementação de uma carteirinha baseado em smart card seguindo o modelo ICAO 9303 adaptado às necessidades da federação CAFé.

## **Projeto Proposto**

O projeto se propõe a fazer um levantamento das potenciais aplicações acadêmicas para uma carteirinha unificada baseada no padrão ICAO 9303 no âmbito das instituições parcerias da RNP. Em especial queremos fazer um estudo compreensivo das características peculiares dos mecanismos de autenticação presentes nos documentos que seguem o padrão ICAO 9303 e adaptá-los para o uso acadêmico. O projeto foca-se na emissão de um documento completamente digital embarcado em um cartão inteligente. No entanto, seguindo a própria proposta da ICAO, vamos propor modelos intermediários para instituições que não

queriam aderir ao modelo completo.

Em nosso estudo queremos levantar os requisitos necessários para a confecção, para a assinatura e reconhecimento dos dados presentes de tais documentos. O estudo também vai vislumbrar as formas de interagirmos com documentos ICAO 9303 de forma a utilizar os dados presentes no processo de autenticação. O uso das características biométricas presentes nos cartões também serão estudadas.

Em suma, a nossa proposta é fazer um levantamento dos requisitos voltados à realidade da RNP e de seus parceiros, levando em conta, por exemplo, a infra-estrutura tecnológica disponível, a natureza dos serviços, bem como a sua realidade financeira e de seus parceiros. Após este levantamento de requisitos vamos fazer algumas implementações que nos ajudem a comprovar a viabilidade da proposta.

O nosso estudo resultará em um relatório técnico, escrito com o objetivo de auxiliar os gerentes de TI em geral, e os membros do GT-GId, em particular, a tomar decisões relativas as formas de emitir documentos que seguem o padrão ICAO 9303 e especial com as funcionalidades adaptadas para o ambiente federado CAFé. Também produziremos algumas provas de conceito tecnológico implementadas a partir dos resultados do nosso relatório, que ajudarão à RNP e seus parceiros a entender e a testar as funcionalidades relatadas.

Em virtude da complexidade relativa ao uso dos cartões inteligentes e ao escasso recurso humano disponível ao projeto, não podemos garantir implementações completas e funcionais, haja vista que este continua sendo um objetivo da proposta.

É importante salientar que já contamos com a infra-estrutura de tecnologia e de cartões necessária para a execução dos experimentos. Os cartões que possuímos tem as características técnicas necessárias e podem ser carregados com softwares equivalentes em termos de funcionalidade a um documento ICAO padrão 9303. Vale lembrar que a disponibilidade de cartões e documentação técnica é extremamente restrita, e por isso iremos implementar uma infra-estrutura equivalente de acordo com a documentação disponível. Também faremos uso de outros documentos no padrão ICAO 9303, tais como passaportes, para testes de compatibilidade e performance.

### **Atividades Realizadas**

A seguir, apresentamos as etapas deste projeto:

- **05/05/2014 - Estudo dos documentos ICAO 9303**

Nesta etapa foi realizada uma leitura da documentação 9303 da ICAO, em especial do documento Vol 2 Parte 3. Foram feitos levantamentos das principais características dos documentos, o que pode ser utilizado, o que pode ser descartado e o que pode ser melhorado.

- **06/05/2014 - Projeto de Aplicação**

Após o estudo feito no documento do item anterior, foi definido quais campos presentes na especificação deveríamos utilizar em nossa proposta de carteira acadêmica.

- **10/05/2014 - Estudo da implementação do JMRTD**

Utilizamos a aplicação JMRTD (a mesma utilizada no PGId 2013) para a criação e carteiras acadêmicas. Na aplicação realizamos diversas alterações, a fim de adequa-la a nossa proposta.

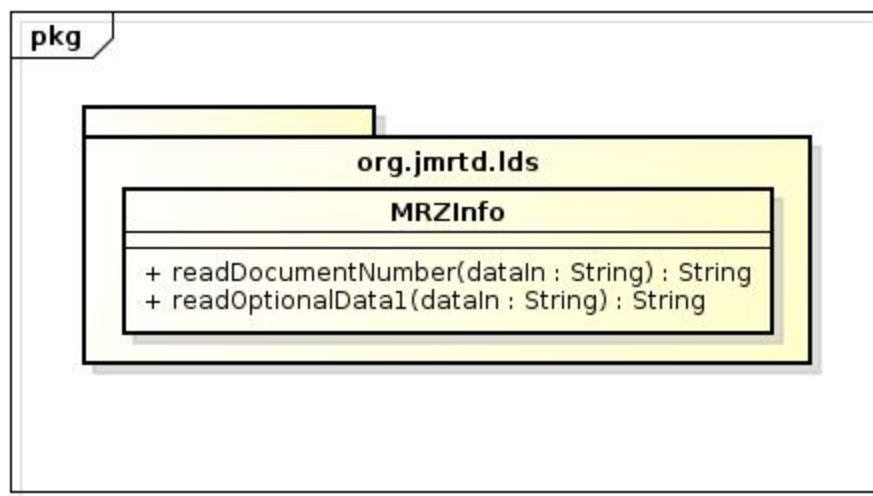
- **15/05/2014 - Instalação de *drivers* para leitura e escrita em cartões**

Nesta etapa trabalhamos com as leitoras que seriam utilizadas para a leitura e escrita nos cartões. A leitora escolhida foi a SDI010, da SCM PC-Card, onde configuramos os *drivers*. Essa leitora possui interface dupla para leitura tanto de cartões sem contato, quanto cartões com contato, como smartcards.

- **20/05/2014 - Entrega do *roadmap***

- **21/05/2014 - Projeto das alterações**

Após o estudo da implementação do JMRTD, iniciou-se as alterações necessárias no código fonte da aplicação. Aqui fizemos um diagrama de classe demonstrando as alterações. Basicamente a alteração mais significativa foi realizada na classe MRZInfo onde foi adicionado duas novas funções para leitura do número do documento e dados opcionais do cartão.



- **21/07/2014 - Entrega de relatório intermediário**

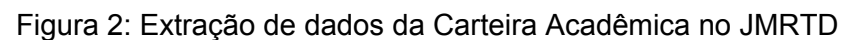
Conforma cronograma do Programa, foi realizada a escrita e entrega de um relatório, onde se apresentou as atividades realizadas até então.

- **22/07/2014 - Inicio dos Testes**

Após as implementações e alterações, iniciou-se os testes. Para as alterações no JMRTD foi utilizado a IDE Eclipse no sistema Operacional Linux Ubuntu 12.04. A maioria das alterações foram feitas na classe *MRZInfo*, onde foram criados dois

[illegible]

Após a criação, foi feita a importação da Carteira Acadêmica na aplicação, com o intuito de demonstrar como é a leitura da Carteira Acadêmica na aplicação, onde tem-se vários dados que podem ser utilizados em outras aplicações. A figura 2 demonstra tais dados, como a identidade do portador da carteira, e informações do certificado digital do emissor.



Foi testado também a escrita dos passaportes em um smart card. O smartcard testado foi um SmartCard JCop com interface sem contato.

Para o teste de escrita, foi criado no JMRTD uma Carteira Acadêmica com vários dados e foto e foi feito o *upload* dessa carteira no no SmartCard. Para o teste de leitura, primeiramente foi realizado a leitura do cartão no próprio JMRTD, tanto utilizando realizando a leitura da Interface sem Contato, quanto a leitura da Interface com Contato do cartão.

Outro teste realizado foi a leitura do cartão em smartphones com a tecnologia NFC. Os smartphones utilizados no Teste possuem o sistema operacional Android, e também a aplicação NFC Passport Reader<sup>1</sup> instalada. A figura 3 a seguir demonstra a leitura feita no dispositivo.

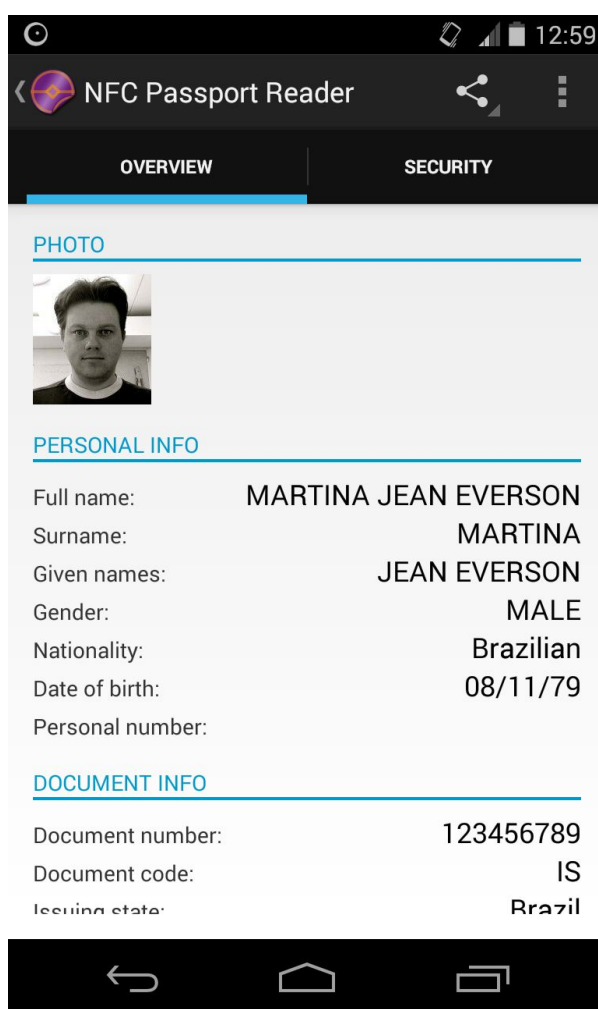


Imagem 3: Leitura da carteira na aplicação NFC Passport Reader

Foi realizado também a escrita de um pequeno tutorial de instalação do applet do JMRTD e como fazer o upload da carteira acadêmica no smartcard JCOP.

<sup>1</sup> [https://play.google.com/store/apps/details?id=nl.novay.nfcpassportreader&hl=pt\\_BR](https://play.google.com/store/apps/details?id=nl.novay.nfcpassportreader&hl=pt_BR)

A seguir, descrevemos em detalhes as alterações realizadas no padrão da documentação 9303.

### MRZ da Carteira Acadêmica

O MRZ (sigla para Machine Readable Zone, ou Zone Legível por Máquina), como o próprio nome já diz, se refere a uma zona com OCR (Optical Character Recognition, ou Reconhecimento ótico de caracteres) do passaporte que pode ser lida por leitoras específicas para leitura de passaportes.

Nossa proposta de MRZ para a carteira acadêmica consiste em manter o número de dígitos para o Tipo de Documento e Estado Emissor. Haverá dois tipos de documentos, o Id Student (IS) e o Id Academic (IA). Isso permitirá que o documento seja avaliado por agentes humanos para identificar estudantes.

Na posição do Estado Emissor, haverá o país da instituição que emitiu o cartão. Isso permitirá que qualquer faça a conexão à federação nacional correta para uma avaliação eletrônica das credenciais. O número de documento será de tamanho variado e conterá um dígito de verificação. Os Dados Opcionais da primeira linha conterão a sigla da instituição, que será separada do Número do Documento por um caractere de preenchimento (<). A figura 4 demonstra detalhadamente como será o MRZ da carteira acadêmica.

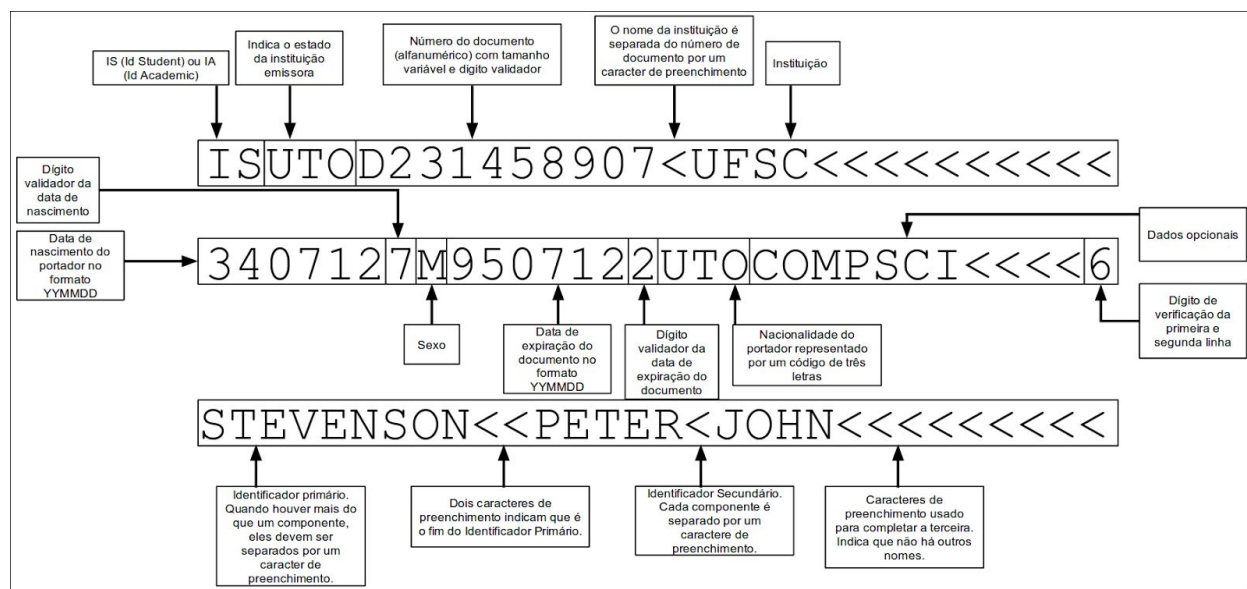


Figura 4: MRZ da carteira acadêmica

É importante ressaltar que iremos manter o MRZ devido ao novo Registro de Identidade Civil (RIC), no qual também está sendo baseado no padrão dos passaportes e também possuirá o MRZ. Com isso, haverá uma grande demanda de utilização de leitoras de passaportes para a leitura do MRZ do RIC, que também poderão ser utilizadas para a leitura das carteiras acadêmicas.

### Zona de Inspeção Vizual da Carteira Acadêmica

O VIZ (*Visual Inspection Zone* ou Zona de Inspeção Vizual) é uma página do passaporte que inclui dados pessoais, como o nome do titular, número de passaporte, país emissor, data de emissão e validade, mas não é necessário um computador ou leitora para a leitura desses dados, podendo ser lida por agentes humanos.

Todos os dados contidos no MRZ, com exceção dos dígitos de validação também estarão impressos no documento, permitindo a verificação visual. A figura 5 demonstra como será a frente da carteira acadêmica.



UTOPIA		ID Student	
	Name: STEVENSON, PETER		
	Sex: Male	Nationality: UTO	Date of Birth: 12 JUL 34
	University: UFSC		
	Document Number: D23145890	Date of Expiry: 12 JUL 15	
	Computer Science - CompSci		
			

Figura 5: Frente da carteira acadêmica

Além do MRZ, a carteira acadêmica também possuirá um QR Code, que conterá todos os dados do MRZ. Isso permitirá que instituições criem suas próprias aplicações para leitura do QR Code. Assim, as universidades têm a opção de evitar o aumento de gastos com a compra de leitores próprias para passaportes, que possuem um alto custo, e são complicadas de gerenciar. Na figura 6 é demonstrado o verso da carteira acadêmica.





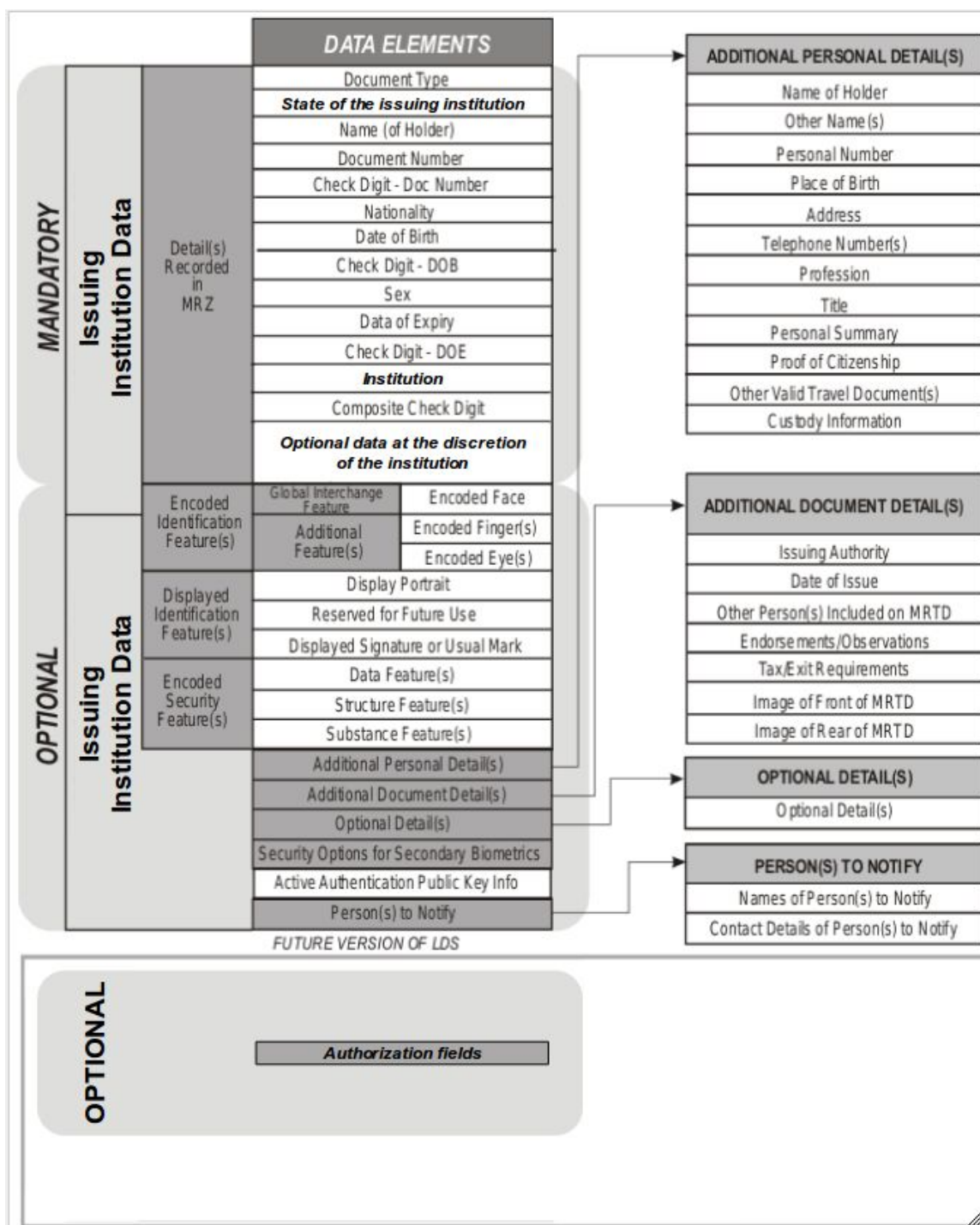


Figura 7: LDS da Carteira Acadêmica

## Cronograma

Cronograma				
Data	Fase	Detalhamento	Entregas	Descrição
5/5/2014	Estudo Bibliográfico	Estudo dos documentos ICAO9303	-	Foi feita a leitura do documento MRTD Vol 2 Parte 3. Foram feitos levantamentos das principais características dos documentos, que serão utilizados na implementação.
6/5/2014	Estudo Bibliográfico	Projeto de aplicação	-	Definição de quais campos do MRTD serão utilizados no documento.
10/5/2014	Estudo Bibliográfico	Estudo da Implementação do JMRTD	-	Está sendo usada a aplicação utilizada no PGID 2013 (JMRTD).
15/5/2014	Estudo Bibliográfico	Instalação de drivers para leitura e escrita de cartões no MS Windows	-	Configuração da leitora de cartões em ambiente Windows.
20/5/2014		Entrega de documento com relatório de detalhamento das atividades (roadmap)	Relatório previsto pelo programa	
21/5/2014		Projeto das alterações do JMRTD	-	Serão feitas algumas alterações na aplicação para adaptá-la à proposta. Documentação, diagrama de classe das alterações, análise de requisitos.
26/5 - 30/5		1ª reunião de acompanhamento (uma por projeto)		
1/6/2014		Alteração do JMRTD	-	Realizar as alterações necessárias no JMRTD para permitir a escrita no cartão.
21/7/2014		Entrega de relatório intermediário	Relatório previsto pelo programa	
22/7/2014		Testes unitários.	-	Teste de criação de passaportes no JMRTD e escrita no cartão.
28/7 - 8/8		2ª reunião de acompanhamento (uma por projeto)		
19/9/2014		Entrega do Relatório Final	Relatório previsto pelo programa	
<b>Legenda</b>	Concluído			
	Em Andamento			
	Atrasado			
	Não iniciado			

## Considerações Finais

As federações são uma realidade no mundo virtual. No entanto sua necessidade de estarmos 100% online, de confiarmos em ambientes computacionais complexos e da impossibilidade de utilizarmos dados privados retira boa parte de usabilidade

Este trabalho teve como objetivo o estudo e implementação de uma nova carteira acadêmica. Foi realizado um estudo profundo da documentação ICAO 9303 com o intuito de excluir o que não poderia ser aproveitado pela carteira acadêmica, e o que poderia estar presente no nosso padrão de carteira acadêmica. Após a realização do estudo, foi iniciado o trabalho de implementação, tanto da alteração do JMRTD quanto os trabalhos com os cartões, onde foi realizada toda a configuração para instalação do *applet* do JMRTD e da carteira criada na aplicação.

Como resultados deste projeto, temos uma aplicação já pronta para geração de cartões acadêmicos, um relatório detalhado de como realizar o upload destes cartões em alguns smartcards e por fim, uma carteira acadêmica no formato físico criada na aplicação JMRTD seguindo o padrão aqui proposto. Além é claro, foi obtido um grande aprendizado sobre o padrão dos passaportes e smartcards.

Este projeto também resultou em um artigo, que foi apresentado no *International Conference on Availability, Reliability and Security (ARES 2014)*<sup>2</sup>, no qual está em publicação. Artigo foi iniciado juntamente com a proposta inicial entregue ao PGId.

Com essa proposta, abre-se uma gama de opções de aplicações da carteira acadêmica, como:

- **Controle de Presença:** A universidade pode usar leitores para controlar a presença de acadêmicos que têm um cartão seguindo a proposta. O aluno só precisa deslizar o seu

<sup>2</sup> <http://www.ares-conference.eu/conference/>

cartão no leitor. A biométrica, embutido no cartão pode ser utilizada como uma prova de presença assinada pelo cartão;

- **Autenticação em MOOCs:** Massive Open Online Courses (MOOCs) é um termo que surgiu em 2008, para um determinado tipo de formato de curso online aberto. Um exemplo da utilização de autenticação em MOOCs seria o aluno ter em mãos um leitor de cartão. Assim, o sistema poderia utilizar o retrato contido no cartão acadêmico e com isso executar uma autenticação contínua, comparando a imagem presente no cartão com a imagem do aluno que está fazendo o curso. Isso garante autenticação e dedicação exclusiva do acadêmico à tela do computador. Dessa forma é fácil de avaliar a performance do estudante;
- **Acesso a Locais controlados:** O estudante também pode acessar outros lugares controlados além da sala de aula utilizando o cartão acadêmico, como o restaurante universitário, onde o cartão poderia armazenar seus bilhetes ou créditos para serem utilizados ali. Outra idéia é usar este cartão para acessar conferências, eventos, bibliotecas e assim por diante;
- **Emissão de Certificados Digitais:** Outra área de aplicação é a emissão automática de certificados usando o cartão. Este processo consiste em gerar uma solicitação de certificado, extraíndo as credenciais do usuário do cartão. Isso é possível porque a instituição já verificou estes dados quando ele foi registrado na federação. Estes certificados podem ser utilizados em curto prazo, como em conferências de acesso a redes sem fio. Ele também pode ser utilizado para aplicações mais clássicas, como a autenticação de alunos e funcionários nos diversos sistemas acadêmicos.

## ANEXO - Tutorial para a instalação do *applet* do JMRTD em cartões compatíveis com o *Global Platform*

Aqui explicamos detalhadamente como realizar a instalação do *applet* do JMRTD em cartões compatíveis com o *Global Platform*.

Primeiramente deve-se realizar o download do SDK do Java Card, versão 2.2.2 em [http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javame-419430.html#java\\_card\\_kit-2.2.2-oth-JPR](http://www.oracle.com/technetwork/java/javasebusiness/downloads/java-archive-downloads-javame-419430.html#java_card_kit-2.2.2-oth-JPR).

Após o download, extrair o conteúdo do arquivo em qualquer local que o usuário desejar. Dentro da pasta *java\_card\_kit-2\_2\_2* há ainda outros arquivos que devem ser extraídos. Depois da extração dos arquivos, setar na variável de ambiente JC\_HOME o local de extração, alterando o arquivo */etc/bash.bashrc*. Por exemplo, se o local de extração do arquivo for na pasta */opt/javacard/*, então teremos que setar a variável da seguinte maneira:

```
export JC_HOME=/home/user/javacard/java_card_kit-2_2_2/
```

Após isso, devemos baixar o código fonte do Applet JMRTD, disponível em <http://sourceforge.net/projects/jmrttd/files/passportapplet/0.0.2/> e extrair. Na classe *PassportApplet*, no pacote *sos.passportapplet*, alterar o método *install*, como segue, com isso, evita-se o problema do cartão não suportar curvas elípticas:

```
/**
 * Installs an instance of the applet. The default crypto mode is now
 * PERFECTWORLD_MODE as the new JCOP41 cards support all required
 * crypto.
 *
 * @param buffer
 * @param offset
 * @param length
 * @see javacard.framework.Applet#install(byte[], byte, byte)
 */
public static void install(byte[] buffer, short offset, byte length) {
    (new PassportApplet(PassportCrypto.CREF_MODE)).register();
}
```

As classes *EvilInterface.java* e *EvilPassportApplet.java*, presentes no mesmo pacote da classe anterior, devem ser removidas, pois o SDK 2.2 ainda não suporta esse tipo de Interface.

Na pasta onde foi extraído o Java Card, copiar a pasta *lib* da pasta *java\_card\_kit-2\_2\_2-rr-bin-linux-do* para a raiz do projeto do JMRTD.

Feito isso, instalar o Ant, executando o seguinte comando.

```
sudo apt-get install ant
```

Após a instalação do Ant criar na raiz do projeto do JMRTD um arquivo com o nome *build.xml*, com o seguinte conteúdo:

```
<?xml version="1.0" encoding="UTF-8" ?>
<project default="convert" name="javacard-starter">
  <property environment="env" />

  <!-- Build specific properties -->
  <property name="target.classes" location="${basedir}/target/classes" />
  <property name="target.javacard" location="${basedir}/target/javacard" />
  <property name="source.java" location="${basedir}/src/sos/passportapplet" />

  <!-- Must point to the folder containing the JAR files from the JCDK -->
  <property name="javacard.libs" location="${basedir}/lib" />

  <!-- Must point to the folder containing the JAR file from the JCDK ant task -->
  <property name="javacard.ant-task" location="${basedir}/lib" />

  <!-- Must point to the folder containing the API export files from the JCDK -->
  <property name="javacard.export"
location="${env.JC_HOME}/api_export_files" />

  <property name="javacard.home" location="${env.JC_HOME}" />

  <property name="verbose" value="true" />
  <property name="noverify" value="false" />

  <!-- Path for JC tasks -->
  <path id="classpath">
    <fileset dir="${javacard.ant-task}">
      <include name="*.jar" />
    </fileset>
  </path>
</project>
```

```

        </fileset>
        <fileset dir="${javacard.libs}">
            <include name="*.jar" />
        </fileset>
    </path>

    <!-- set the export path to the Java Card export files -->
    <path id="export" description="set the export file path">
        <fileset dir="${javacard.export}">
            <include name="**/*.exp" />
        </fileset>
        <pathelement path="${javacard.export}" />
        <pathelement path="${target.classes}" />
        <pathelement path="${target.javacard}" />
    </path>

    <!-- Definitions for tasks for Java Card tools -->
    <taskdef name="apdu"
classname="com.sun.javacard.ant.tasks.APDUToolTask" classpathref="classpath" />
    <taskdef name="capgen"
classname="com.sun.javacard.ant.tasks.CapgenTask" classpathref="classpath" />
    <taskdef name="deploycap"
classname="com.sun.javacard.ant.tasks.DeployCapTask" classpathref="classpath" />
    <taskdef name="convert"
classname="com.sun.javacard.ant.tasks.ConverterTask" classpathref="classpath" />
    <taskdef name="verifyexport"
classname="com.sun.javacard.ant.tasks.VerifyExpTask" classpathref="classpath" />
    <taskdef name="verifycap"
classname="com.sun.javacard.ant.tasks.VerifyCapTask" classpathref="classpath" />
    <taskdef name="verifyrevision"
classname="com.sun.javacard.ant.tasks.VerifyRevTask" classpathref="classpath" />
    <typedef name="appletnameaid"
classname="com.sun.javacard.ant.types.AppletNameAID" classpathref="classpath" />
    <typedef name="jcainputfile"
classname="com.sun.javacard.ant.types.JCAInputFile" classpathref="classpath" />
    <typedef name="scriptgen"
classname="com.sun.javacard.ant.tasks.ScriptgenTask" classpathref="classpath" />

```

```

<target name="init">
    <mkdir dir="${target.classes}" />
    <mkdir dir="${target.javacard}" />
</target>

<target name="clean">
    <delete dir="${target.classes}" />
    <delete dir="${target.javacard}" />
    <delete dir="${target.javacard.eeprom}" />
</target>

<target name="compile" depends="init" description="Compile source code to
class files">
    <!-- Compile the java code from ${src} to ${classes} -->
    <javac debug="yes" optimize="no" srcdir="${source.java}"
destdir="${target.classes}" source="1.1" target="1.1">
        <classpath refid="classpath" />
    </javac>
</target>

<target name="convert" depends="compile" description="Convert class files to
cap files">
    <convert package="sos.passportapplet"
packageaid="0xA0:0x00:0x00:0x02:0x47:0x10"
majorminorversion="2.2" classdir="${target.classes}"
outputdirectory="${target.classes}"
jca="true"
exp="true"
cap="true"
debug="true"
verbose="${verbose}"
noverify="${noverify}">
        <appletnameaid aid="0xA0:0x00:0x00:0x02:0x47:0x10:0x01"
appletname="PassportApplet" />
        <exportpath refid="export" />
        <classpath refid="classpath" />
    </convert>
</target>

```



```

        <target name="deploy" description="" depends="convert">
            <deploycap outeefile="${target.javacard}/chaining.eeprom"
capfile="${target.classes}/chaining/javacard/chaining.cap" classpathref="classpath"
crefexe="${javacard.home}/bin/cref" />
            <apdu ineefile="${target.javacard}/chaining.eeprom"
outeefile="${target.javacard}/chaining.eeprom" classpathref="classpath"
crefexe="${javacard.home}/bin/cref" />
            <!-- <scriptgen outfile="${target.javacard}/chaining.scr"
capfile="${target.classes}/chaining/javacard/chaining.cap" classpathref="classpath" />
-->
        </target>

        <target name="all" depends="clean, convert" />
</project>

```

Após a realização dessas alterações, navegar até a pasta do projeto e executar o seguinte comando no terminal:

```
ant all
```

Após esse passo, será gerado uma estrutura de pastas contendo o .jca, .exp e o .cap do applet.

### Instalação do gpshell

Antes de realizarmos o upload para o cartão, precisamos instalar o GPShell, começando pelos pacotes necessários, que pode ser feito com o seguinte comando:

```
$ sudo apt-get install pcscd libccid libpcsc-lite-dev libssl-dev libreadline-dev autoconf
automake build-essential docbook-xsl xsltproc libtool libltdl-dev
```

Realizar a instalação do *GlobalPlatform Library* e do *GPShell*, fazendo o download no seguinte link: <http://sourceforge.net/projects/globalplatform/files/> extrair e executar os seguintes comandos:

```
$ ./configure
$ make && sudo make install
```

Fazer o download do *GPPCSCConnectionplugin* no link [http://en.sourceforge.jp/projects/sfnet\\_globalplatform/downloads/GlobalPlatform%20Library/GlobalPlatform%20Library%206.0.0/gppcscconnectionplugin-1.1.0.tar.gz](http://en.sourceforge.jp/projects/sfnet_globalplatform/downloads/GlobalPlatform%20Library/GlobalPlatform%20Library%206.0.0/gppcscconnectionplugin-1.1.0.tar.gz) extrair e executar o comando:

```
$ ./configure  
$ make && sudo make install
```

Baixar e instalar o OpenSC, seguindo os seguintes comandos:

```
$ git clone https://github.com/joelhockey/OpenSC.git  
$ cd OpenSC/  
$ ./bootstrap  
$ LIBS=-lntdl ./configure --prefix=/usr --sysconfdir=/etc/opensc  
$ make && sudo make install
```

Após realizar todas as instalações necessárias, pode-se fazer o upload do .cap do JMRTD no smartcard com o seguinte código do gpshell:

```
mode_211  
enable_trace  
enable_timer  
  
establish_context  
card_connect  
select -AID A0000001510000  
open_sc -security 3 -keyind 0 -keyver 0 -mac_key  
404142434445464748494a4b4c4d4e4f -enc_key  
404142434445464748494a4b4c4d4e4f // Open secure channel  
delete -AID A0000002471001  
delete -AID A00000024710  
install -file target/classes/sos/passportapplet/javacard/passportapplet.cap  
-nvDataLimit 2000 -instParam 00 -priv 2  
card_disconnect  
release_context
```

Talvez o script precise de pequenas alterações conforme o manual do fabricante correspondente. Após esses passos, o applet deve ser instalado no cartão com sucesso.