Relatório de Segurança em Sistemas Informáticos

Autenticação em cartões electrónicos – Cartão do Cidadão

Bruno Duarte – ei07136
Pedro Barbosa – ei08036
Rúben Veloso – ei11001

Índice

Introdução

Este documento tem como objectivo descrever e demonstrar a segurança associada ao processo de autenticação através de cartões electrónicos, no âmbito da disciplina de Segurança em Sistemas Informáticos do 5º ano do Mestrado Integrado em Engenharia Informática e Computação na FEUP.

Cada vez mais, nos dias de hoje surge a necessidade de se realizar a validação e autenticação das pessoas no mundo informático nas suas transacções.

E no que diz respeito à autenticação de uma entidade, uma assinatura digital ganha especial importância pois tem mais valor que uma assinatura escrita, portanto é necessário estudar-se bem estes problemas e garantir que estes métodos de autenticação são realmente de confiança para os seus utilizadores.

Assim o trabalho escolhido consiste em estudar e demonstrar a segurança associada ao processo de autenticação de utilizadores por cartões electrónicos com micro-processador, mais concretamente o Cartão de Cidadão, um tipo de cartão comum à maioria dos cidadãos portugueses, e sobre o qual são frequentes artigos e reportagens na comunicação social.

A divisão do trabalho está efectuada em três fases:

Inicialmente, uma breve descrição sobre o que é e quais as principais funcionalidades do cartão do cidadão bem como uma descrição com algum pormenor acerca dos processos envolvidos na autenticação.

Na segunda fase podemos encontrar os testes efectuados à aplicação oficial do cartão do cidadão, procurando identificar possíveis vulnerabilidades.

Por fim, na terceira fase estão presentes algumas dicas para que um programador possa utilizar a API fornecida para desenvolver uma aplicação, e ainda uma componente prática com o exemplo de aplicações simples que utilizam a autenticação pelo cartão do cidadão, desenvolvidas durante a elaboração deste trabalho.

Cartão de Cidadão

O que é e para que serve o Cartão de Cidadão?

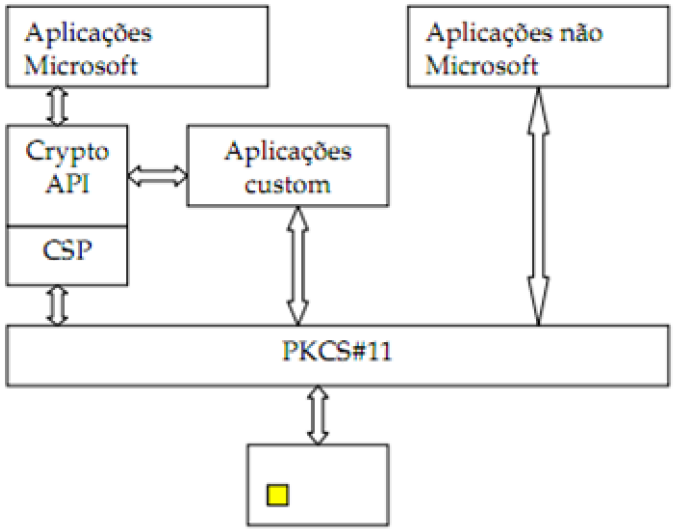
O Cartão de Cidadão é um documento físico e electrónico, fácil de usar, que permite a identificação dos cidadãos através de diversos canais de comunicação com a Administração Pública e Entidades Privadas. Suportando assim interacções presenciais físicas e electrónicas, assim como interacções não presenciais, garantindo, equivalência ao nível da segurança e de valor legal com os meios tradicionais de identificação presencial. O cartão destina-se a facilitar a vida aos cidadãos quando se dirigem aos serviços públicos, presencialmente, pelo telefone ou pela Internet.

O Cartão de Cidadão apresenta-se como um verdadeiro certificado de cidadania, assumindo a forma dupla de um documento físico que identifica visual e presencialmente o cidadão (tal como o Bilhete de Identidade), e um documento digital que permite ao cidadão identificar-se e autenticar-se electronicamente nos actos em que intervenha perante entidades públicas e privadas (através de um PIN pessoal).

Tecnologicamente, o Cartão de Cidadão encontra-se alinhado com os standards internacionais relevantes, em especial ao nível do espaço da União Europeia. Assume a forma de um smartcard, um cartão com microchip incorporado com capacidades de armazenamento de informação e de processamento criptográfico, que assegura os mais elevados padrões de segurança na protecção da confidencialidade e integridade da informação pessoal do cidadão, no respeito pela legislação nacional e as normas europeias correspondentes.

Arquitetura do sistema

Para as aplicações standard Microsoft® (Office, Outlook) é criado um Cryptographic Service Provider (CSP) que implementa as operações criptográficas do smartcard. Uma aplicação nunca chamará esta implementação directamente mas sim através de uma interface standard chamada Crypto API. A implementação CSP utiliza a segunda interface implementada, PKCS#11. Esta interface é também usada por aplicações não standard Microsoft.



Processos envolvidos na autenticação do Cartão de Cidadão

Interfaces que podem ser utilizadas para manipular o cartão do cidadão em diferentes SO (Windows, Linux, Mac Os)

- PKCS#11
- eID lib (= a 'SDK' ou 'Software Development Kit')

PKCS#11

PKCS é um grupo de padrões de criptografia da chave pública relativa à segurança desenvolvida pela RSA Laboratories.

O PKCS#11 trata-se de uma dessas normas que define uma API que facilita a interacção com smart cards. Esta API foi desenvolvida de forma a suportar os métodos criptográficos mais utilizados (RSA keys, X.509 certificates, DES/Triple DES keys, etc).

A utilização deste método para implementar uma aplicação que utilize o cartão do cidadão requer um estudo da API e do cartão do cidadão, pelo que o seu uso não é recomendado em pequenos projectos. Actualmente já existe uma biblioteca que permite a abstracção de todas estas técnicas (*pteidlibj.jar*) que pode ser utilizada em projectos Java.

eID (Electronic Identity Card) lib

Esta biblioteca é no fundo o "SDK" usado como base para o desenvolvimento de aplicações sobre cartões electrónicos, sendo utilizada em vários países além de Portugal como base para o seu próprio sistema de cartões electrónicos. Pode servir igualmente como base ao desenvolvimento de aplicações próprias sobre cartões electrónicos, tal como será detalhado posteriormente neste documento.

Teste das aplicações oficiais do C.C.

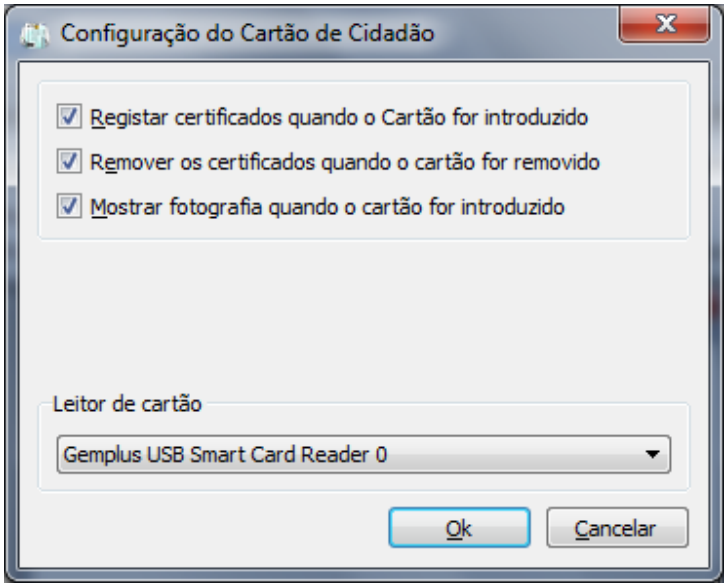
Condições de teste:

- Aplicação CC de 32 bits em Linux Ubuntu 11.10
- Aplicação CC de 32 bits em Windows 7
- Aplicação CC de 64 bits em Windows 7

A aplicação *eID GUI*, presente no '*middleware'*, pode ser usada para ver e gerir a informação no cartão do cidadão, efectuando as seguintes tarefas: mostrar informação sobre o cidadão e fotografia, mostrar e alterar a morada do cidadão, ler os certificados do governo (*ECRaizEstado*) e do cidadão (*EC* de Assinatura digital, *EC* de Autenticação do cartão do cidadão), registar os certificados do governo e do cidadão (apenas disponível na versão Windows), gestão de códigos *PIN* (testar e alterar *PIN* de autenticação, morada ou assinatura) e ainda a gestão da informação guardada no cartão sob a forma de texto pelo utilizador.

Existe igualmente a *Tray Applet*, aplicação que é instalada como uma funcionalidade da área de notificação. No Windows, aparece normalmente no canto inferior direito do ecrã, e quando activada (o utilizador pode desactivá-la), verifica se um cartão eID está inserido. Após inserir o cartão será mostrada a fotografia do cidadão durante uns segundos.

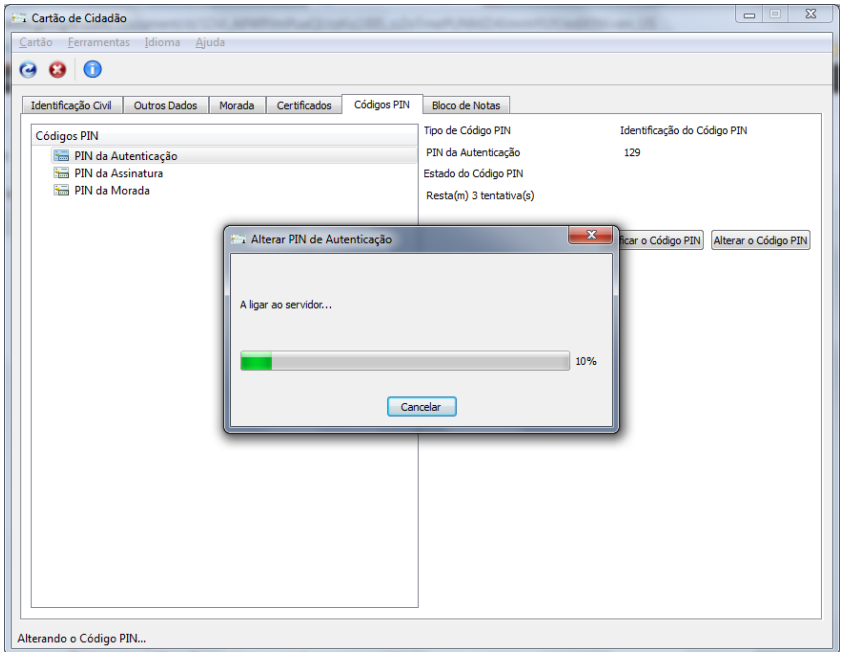
Irá também registar automaticamente (se esta opção estiver activada) os certificados do cartão na Microsoft *certificate store*, caso ainda não estejam registados. Quando o cartão é removido os certificados registados são automaticamente removidos da certificate store (se esta opção estiver activada). Esta funcionalidade a nível de certificados é apenas implementada na plataforma Windows devido às outras plataformas (Mac e Linux) não suportarem o conceito de *certificate stores*.



É importante referir que foram testados tanto a versão Windows como Linux da aplicação, e que o facto de as capturas de ecrã serem em Windows se deve a esta versão ser mais completa (pelas funcionalidades que usam *certificate stores*). Assim, por uma questão de conformidade a escolha recaiu na versão Windows, e as capturas de ecrã foram feitas nessa mesma plataforma.

Outro aspeto importante que só foi detetado durante o teste da aplicação, é que para alterar o PIN de autenticação, ao contrário dos outros PINs, é necessária uma ligação à internet que permita à aplicação comunicar a mudança da palavra-passe a um servidor que lhe dá permissão para alterar o código.

Relativamente aos testes, uma última nota para o facto de a aplicação de 64 bits em Windows estar menos bem conseguida, e em várias situações ser necessário encerrar o processo e voltar a reiniciar a aplicação.

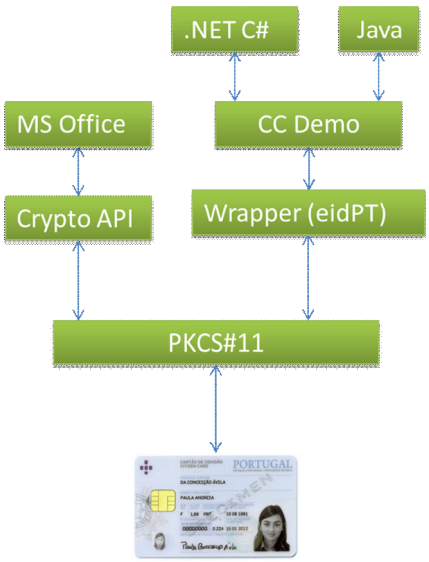


Aplicações realizadas

Inicialmente, decidiu-se implementar uma aplicação em C# que utilizada a eID lib (apenas compatível com Windows) e que permite visualizar informação do utilizador e efectuar as autenticações através dos códigos PIN. Mais tarde, implementou-se uma pequena aplicação em Java que tirou partido da biblioteca *pteidlibj.jar*, sendo esta já uma aplicação multiplataforma, testada tanto em ambiente Windows 7 como em Linux 11.10.

Para obter mais algum detalhe sobre as aplicações, é possível consultar capturas de ecrã de ambos os programas nos anexos, bem como uma pequena explicação do que representam.

.



Princípios básicos de segurança no cartão de cidadão

Para efectuar uma aplicação que transmita alguma confiança aos seus utilizadores e que, acima de tudo respeite os mecanismos de segurança para com os utilizadores, deve ser utilizada a biblioteca *eID lib* que garante desde logo alguns princípios básicos de segurança.

Exemplo disso são as chamadas a funções como obter a morada (PTEID_GetAddr()) ou a assinatura (PTEID_GetCertificates()) que no caso da utilização da biblioteca oficial, chama implicitamente as funções que pedem o respectivo código PIN ao utilizador. No caso de uma aplicação que não use a biblioteca, caso o programador tente mostrar directamente dados ocultos (como a morada) e não peça o respectivo código PIN, é lançada uma excepção (foi experimentando desta forma que se chegou ao erro).

Para a verificação dos códigos PIN, existem ficheiros (com o formato SOD) guardados no cartão, que possuem hashs correspondentes aos diferentes PINs e também aos certificados que cada PIN utiliza.

Ao utilizar certificados, a função PTEID_SetSODCAs() permite definir quais são e qual a localização dos certificados utilizados para assinar o ficheiro SOD, o que é útil no caso de os certificados não terem ainda sido obtidos (ou terem sido mudados de directório). A função PTEID_SetSODCheckin() pode depois ser usada para verificar a presença da hash no ficheiro SOD correta, e assim dar seguimento ao pedido efectuado para mostrar os dados.

Possíveis falhas no Cartão de Cidadão

Após os testes efectuados às aplicações oficiais disponibilizadas, concluiu-se que não existem falhas de segurança na autenticação evidentes. Tentou-se reproduzir o erro relacionado com a utilização de PKCS#11 na versão linux da aplicação oficial, mas tal não foi possível pois o desbloqueio das *keystores* ocorreu normalmente e não foi lançada a suposta excepção que permitiria o acesso às *keys* do cartão, portanto não foi possível demonstrar a vulnerabilidade do cartão.

Para tentar confirmar a existência da mesma, tentamos ainda entrar em contacto com o utilizador que tinha apontado o tal problema, mas não obtivemos resposta após várias tentativas de contacto.

Ainda que a existência de falhas evidentes no cartão do cidadão não tenha ocorrido, convém sempre ao utilizador estar atento a vários factores aquando da utilização do cartão do cidadão. Devido à aplicação do cartão ter uma interface convencional para a inserção de dados, nomeadamente da palavra passe, facilmente esta poderá ser capturada através de um *keylogger*.

Assim, a inclusão de um teclado virtual seria uma solução barata para contornar o problema. Mais seguro ainda seria que os cartões possuíssem uma forma independente do terminal de introduzir os códigos de aceso.

Outro cuidado a ser tido ao assinar um documento, prende-se com a necessidade do utilizador se certificar que está a assinar realmente o documento certo, e não está a ser levado a assinar outro documento que não o pretendido.

Conclusão

O suporte e documentação disponível sobre a API do cartão de cidadão são muito escassos. Existe um wrapper .NET que facilita o desenvolvimento em C# ou VB e ainda um fórum de discussão e demo bastante completo sobre a utilização deste wrapper, sendo este o principal motivo que levam os programadores a escolherem estas ferramentas. Noutras plataformas, JAVA por exemplo, também existe um *wrapper* disponível, no entanto não existem exemplos nem demos.

As falhas do cartão de cidadão terão que ser levadas a sério, pois uma assinatura digital tem mais valor do que uma assinatura manuscrita. No entanto estas últimas têm muitos outros riscos e são mais fáceis de ser ultrapassados, pelo que a utilização de assinaturas digitais não será pior do sistema actualmente em uso.

A aplicação em si do cartão de cidadão é segura, e mesmo testando com a aplicação externa não conseguimos reproduzir a excepção que levaria ao erro relatado. Tentamos ainda por várias vezes contactar com o autor da denúncia, mas o mesmo não respondeu.

Pode-se concluir no entanto que se o utilizador apenas usar a aplicação oficial do cartão do cidadão não terá problemas. Apenas se usar o cartão numa aplicação externa com intenções menos boas poderá ter problemas.

Assim, uma boa forma de assegurar a segurança dos cartões de cidadão seria criar uma forma de certificar que o programa utilizado usa a biblioteca eID lib, ou outra que garanta uma ordem correcta de operações sobre o cartão. Neste caso, o utilizador teria uma forma de saber á partida se a aplicação é certificada e poderia utilizar a aplicação de forma segura.

Bibliografia

Site Oficial (Manuais, Aplicações Oficiais)
http://www.cartaodecidadao.pt/ (último acesso em 01/12/2011)

eIDPT - Cartão de Cidadão .NET Wrapper
http://cartaodecidadao.codeplex.com/ (último acesso em 01/12/2011)

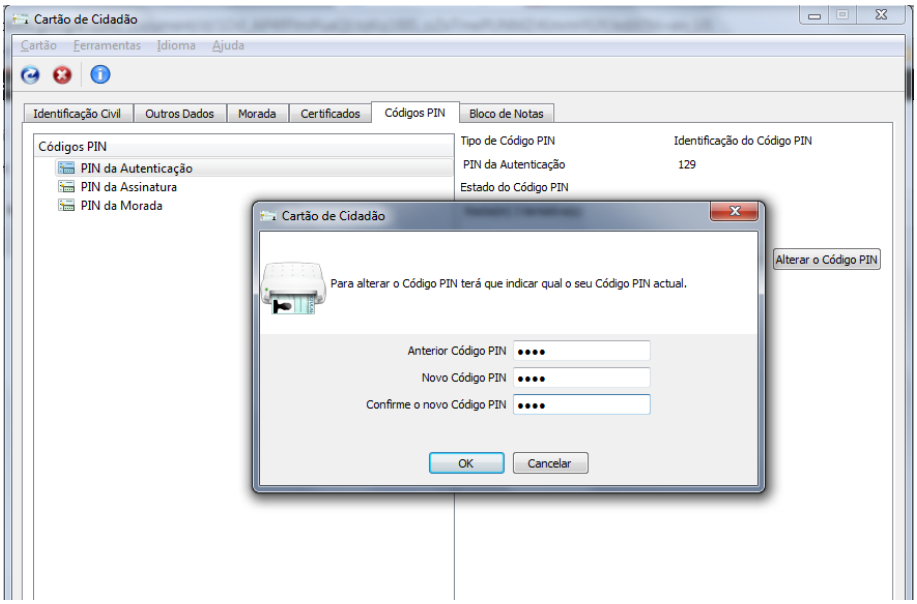
Serviços Online com o Cartão de Cidadão
http://www.senha001.gov.pt/

CUC Manual Técnico Middleware do Cartão do cidadão(último acesso em 20/11/2011)
http://www.cartaodecidadao.pt/index.php?option=com_content&task=view&id=115&Itemid=100&lang=pt

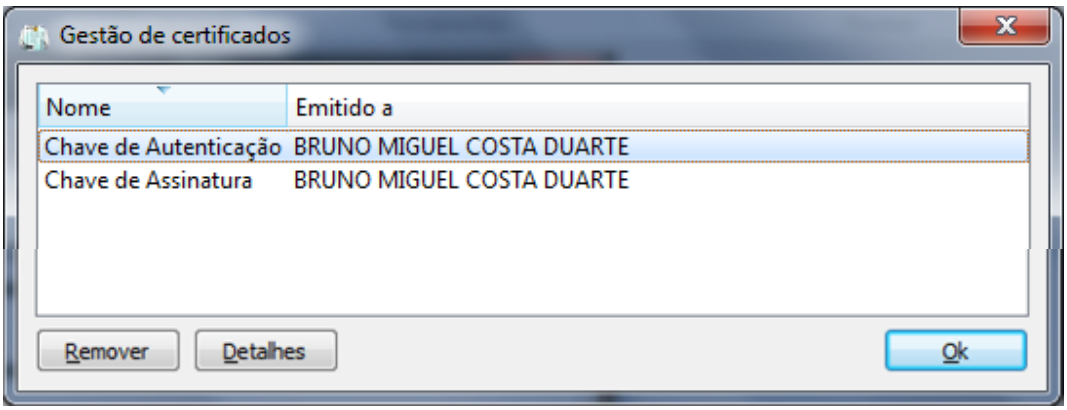
Anexos

Outros testes da aplicação oficial:

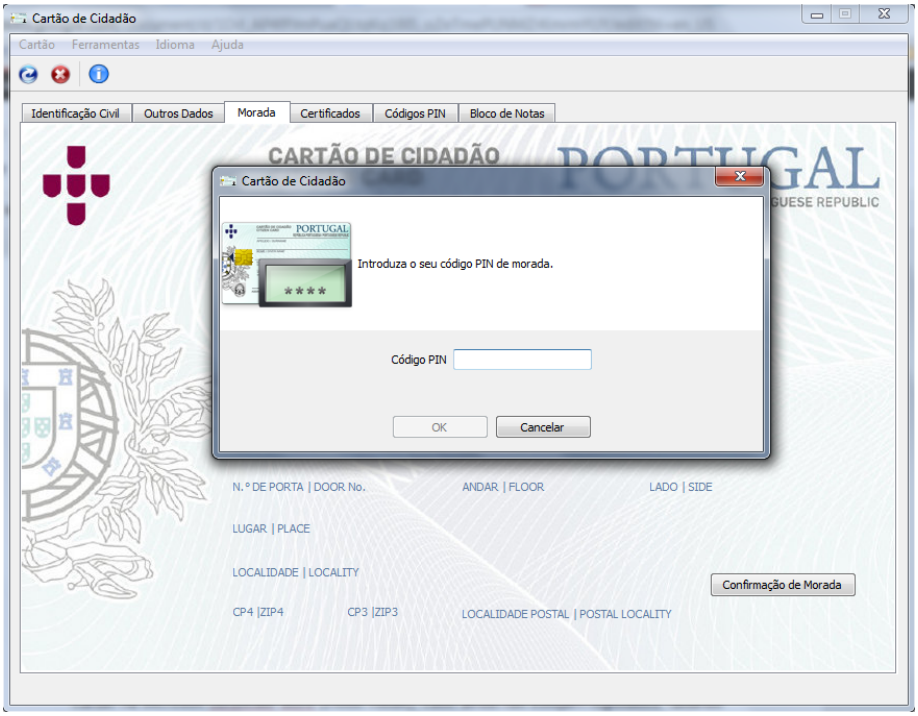
Exemplo de alteração do código PIN: se código antigo não corresponder é lançada uma excepção e o utilizador tem menos uma tentativa para introduzir o PIN correcto; quando os códigos novos inseridos não são iguais é mostrada uma mensagem de erro, caso contrário é alterado o PIN.



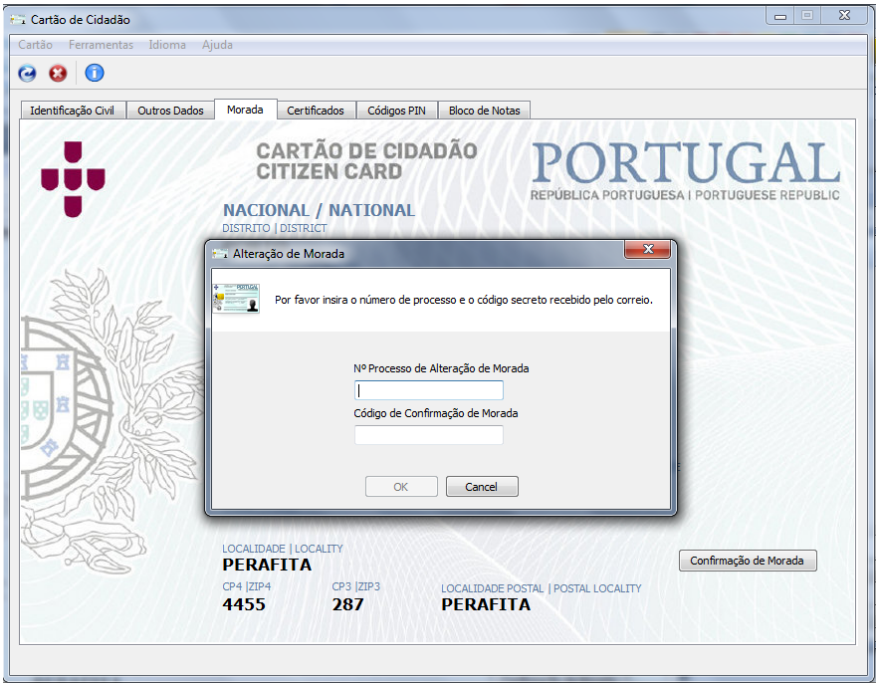
Exemplo de certificados validados na aplicação



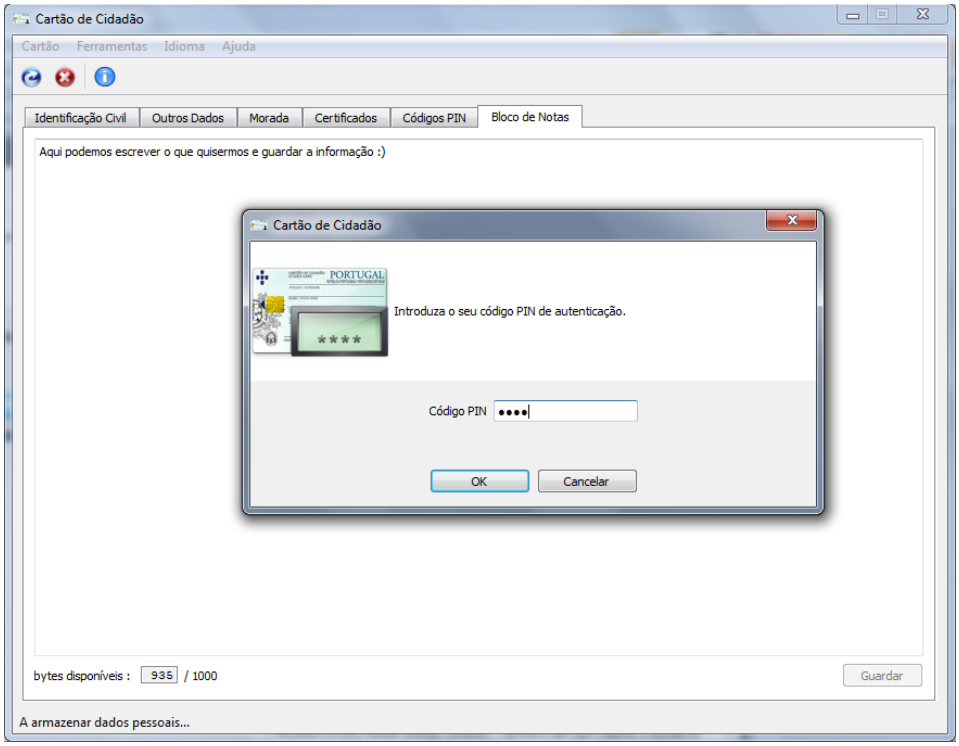
Pedido do Pin de morada



Alteração da morada



Inserção de informação a ser guardada no cartão



Demonstração do uso da API eIDPT

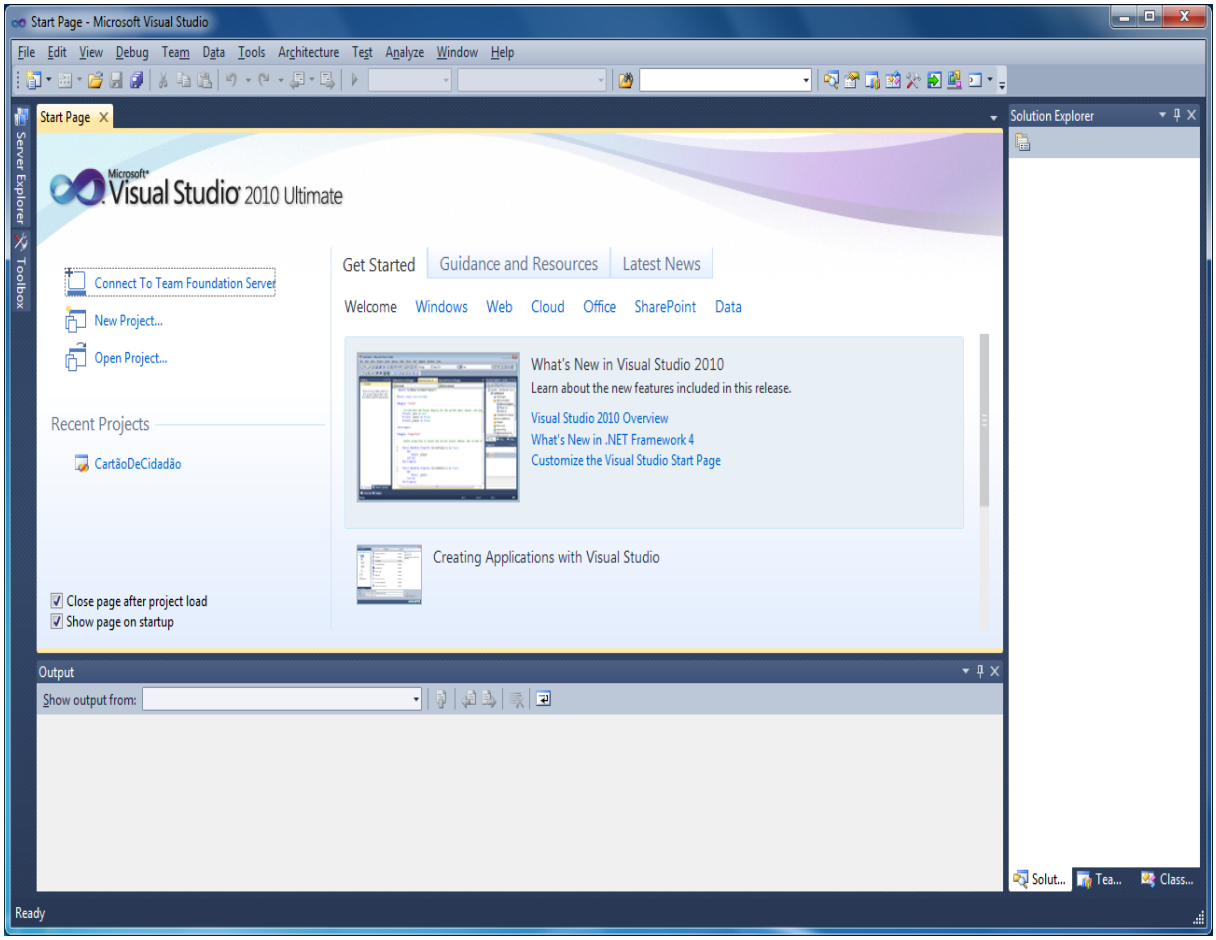
Como todas as novas aplicações são difíceis de entender mesmo para quem já sabe como programar, decidimos então ajudar nos primeiros passos para quem desejar utilizar o cartão do Cidadão de Cidadão para se autenticar nas suas aplicações.

Para começar é importante referir que se pode desenvolver para todos os ambientes existentes desde o Windows, Linux, Mac OS e até para Java.

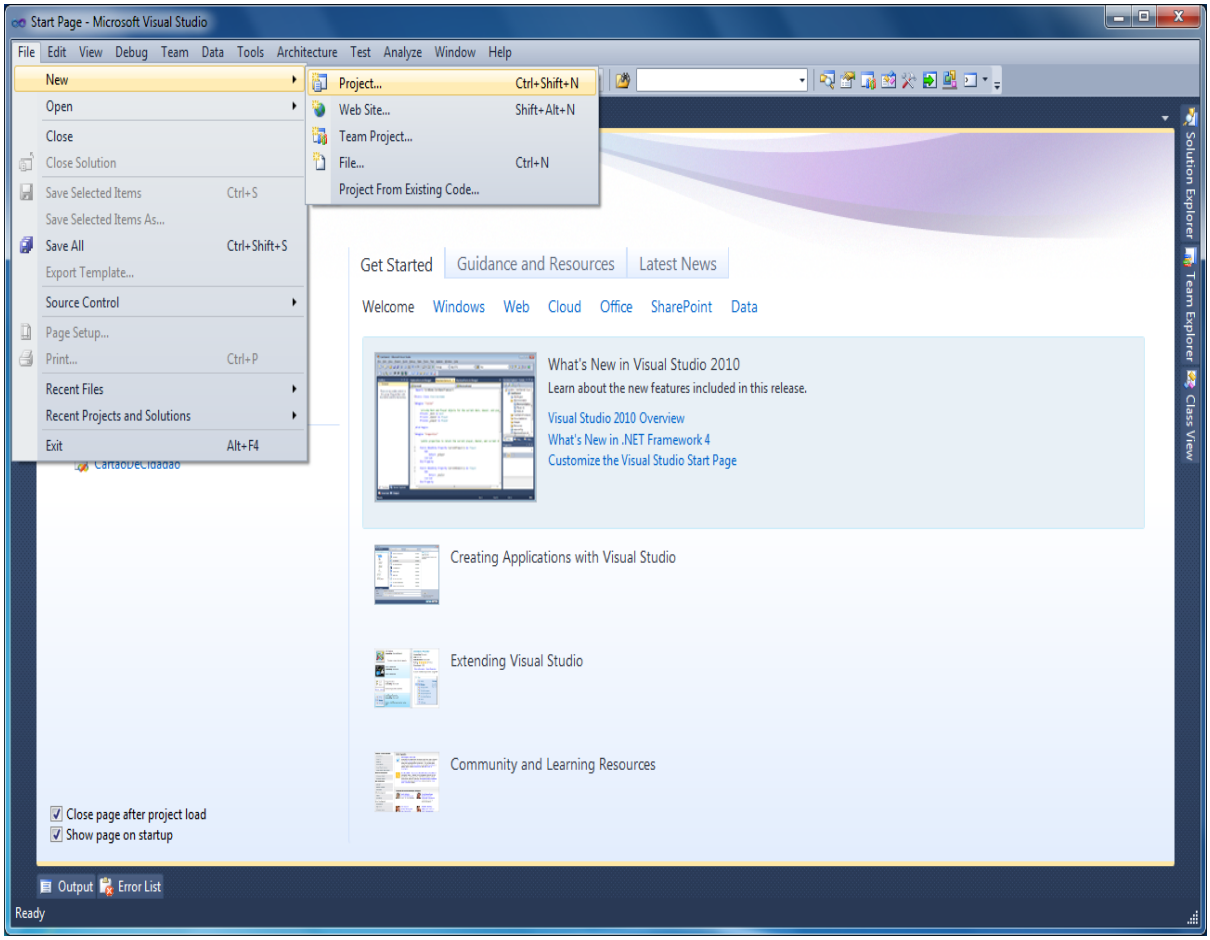
Optamos por demonstrar o uso da API eIDPT em .NET, utilizando o ambiente Windows e o IDE Visual Studio 2010 para fazer uma pequena demonstração de como se obter dados do cartão, e validar através do pedido do código PIN.

Existe muita informação que se pode obter no site oficial do Cartão do Cidadão sobre as API desde manuais ao código fonte da mesma, esses links são fornecidos no fim do relatório.

Comecemos por abrir o Visual Studio 2010, daqui por diante denominado por “VS2010”.

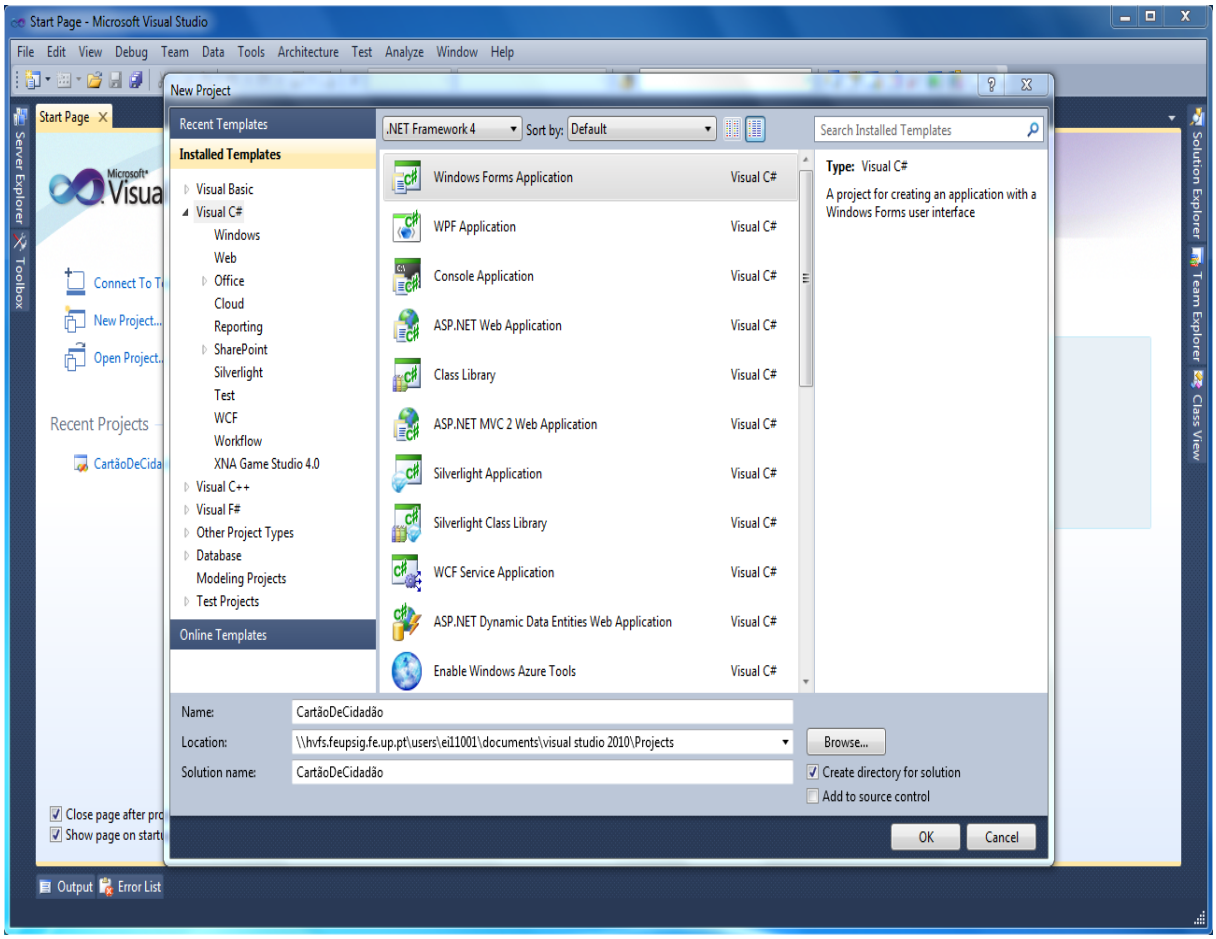


De seguida clica-se no menu File e selecciona-se a opção “Project...”

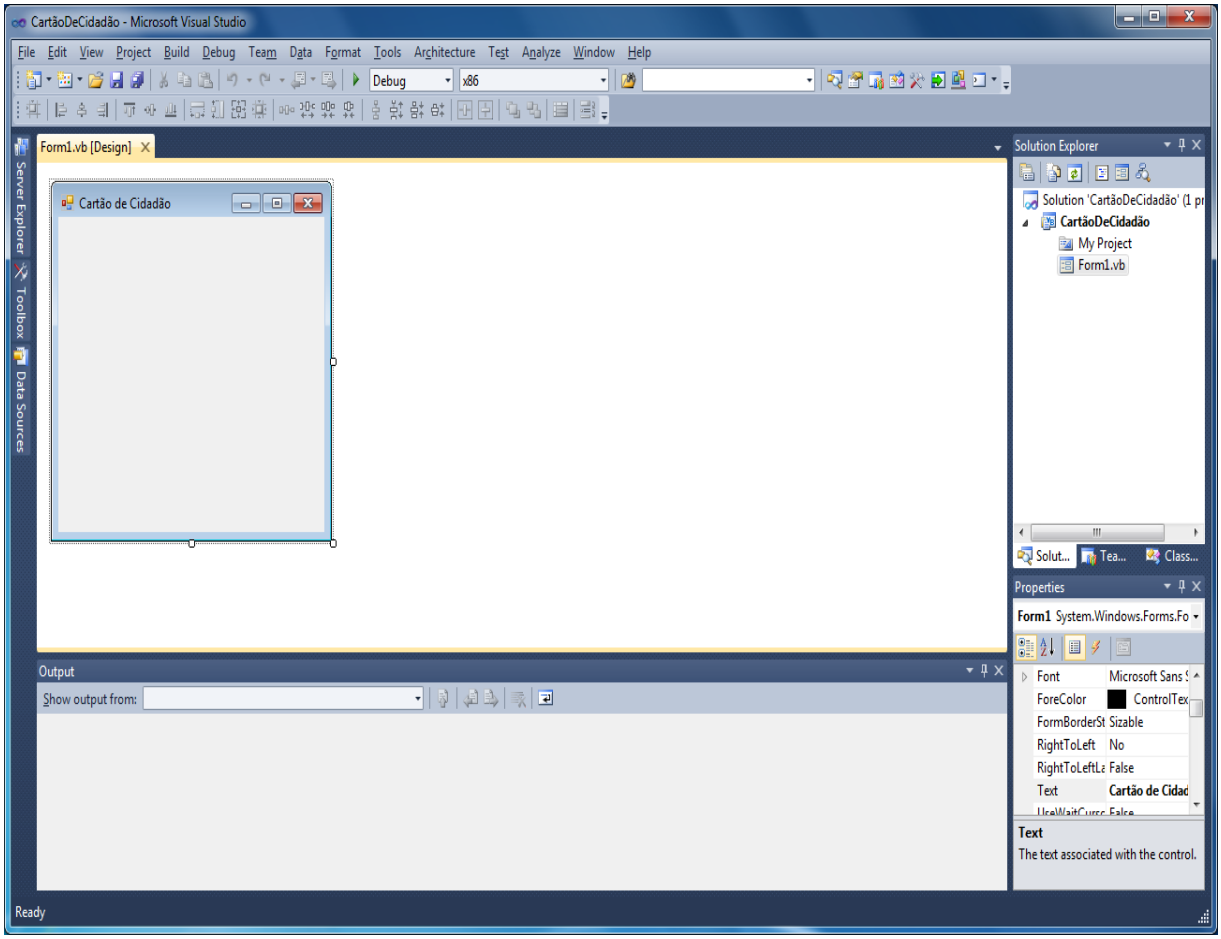

Start Page - Microsoft Visual Studio
File Edit View Debug Team Data Tools Architecture Test Analyze Window Help
New
Open
Close
Close Solution
Save Selected Items Ctrl+S
Save Selected Items As...
Save All Ctrl+Shift+S
Export Template...
Source Control
Page Setup...
Print... Ctrl+P
Recent Files
Recent Projects and Solutions
Exit Alt+F4
Project... Ctrl+Shift+N
Web Site... Shift+Alt+N
Team Project...
File... Ctrl+N
Project From Existing Code...
Get Started
Guidance and Resources
Latest News
Welcome
Windows
Web
Cloud
Office
SharePoint
Data
What's New in Visual Studio 2010
Learn about the new features included in this release.
Visual Studio 2010 Overview
What's New in .NET Framework 4
Customize the Visual Studio Start Page
Creating Applications with Visual Studio
Extending Visual Studio
Community and Learning Resources
Close page after project load
Show page on startup
Output
Error List
Ready
Solution Explorer
Team Explorer
Class View

Escolhe-se um novo projecto em Visual C# para permitir desenhar uma interface mais amigável, e selecciona-se a plataforma .NET Framework 4.

Insiram o nome da aplicação que vão desenvolver e cliquem no botão OK.

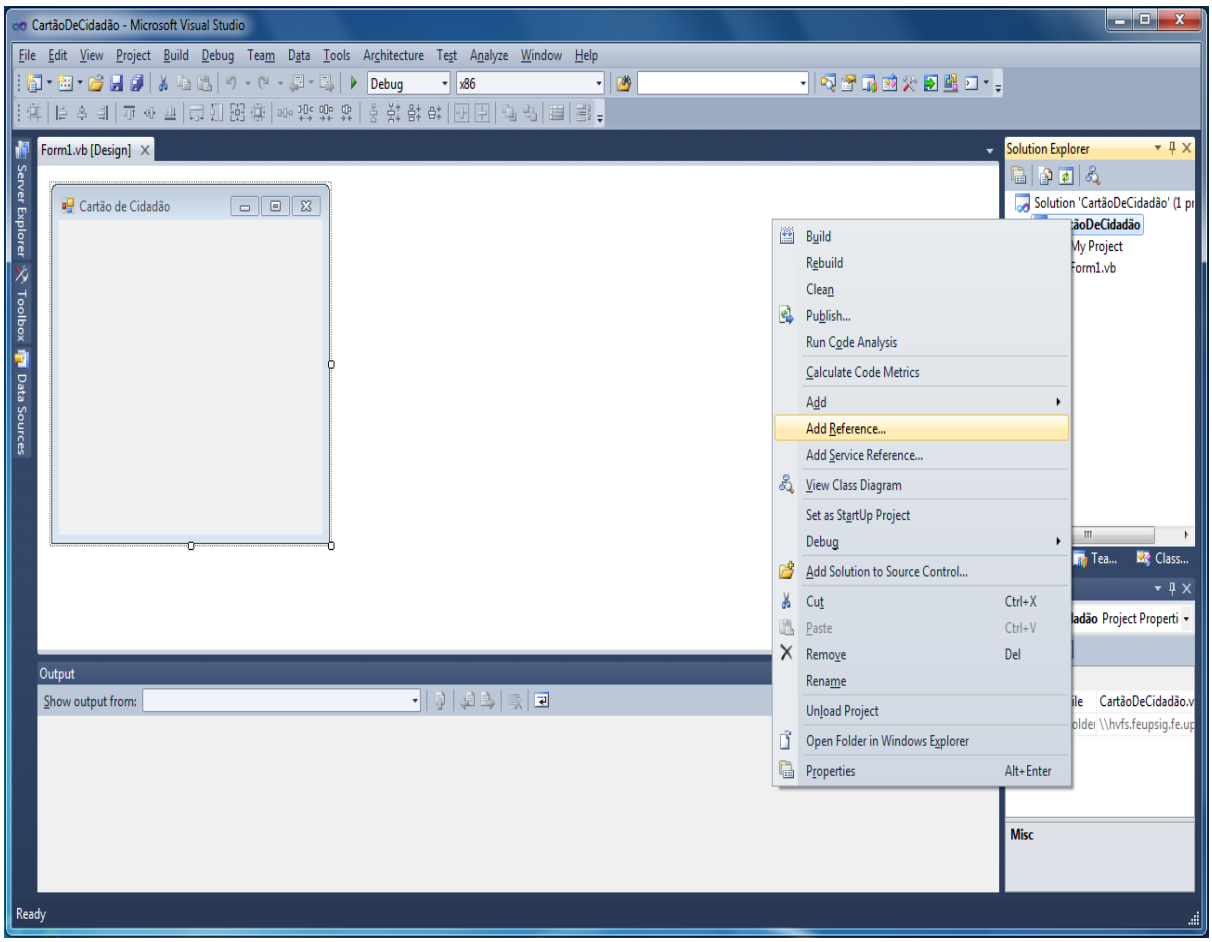


É apresentada a primeira Form gerada automaticamente, neste exemplo simples apenas uma Form é necessária para mostrar os dados do cartão.



Efectuados estes primeiros passos agora é necessário importar as Referências da API para o projecto, vão ser importados dois *.dll um para aceder a todas as funções do cartão e um segundo para auxiliar no desenho da fotografia inserida no Cartão de Cidadão.

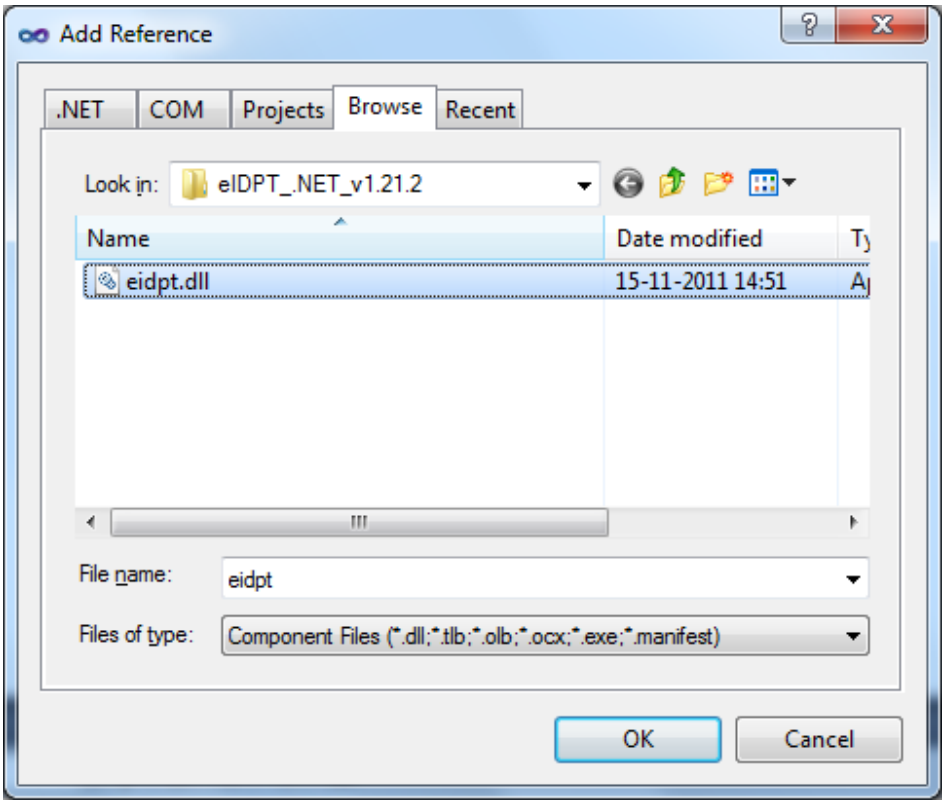
Para isso clica-se com o botão do lado direito do rato sobre o projecto e selecciona-se a opção “Add Reference...”.



Surge uma pop-up para indicar a API, agora pode escolher fazer o download de uma versão já compilada ou fazer download do código fonte para compilar.

Para ser mais rápido optou-se por uma versão já compilada a versão eIDPT_NET_v1.21.2, clica-se no botão OK e fica adicionada as referências do projecto.

Para a segunda API torna-se a repetir o processo de adicionar referencia e nesta optou-se por usar a indicada também no site oficial CSJ2K, para fazer o tratamento da imagem.



De seguida, podemos ver um excerto simples que mostra como se pode obter a informação do cartão e as respectivas verificações relacionadas com o cartão estar disponível, ser válido, etc. Este código pode constituir (tal como no exemplo que foi implementado) uma base genérica para a construção de diversas aplicações:

```
public frmMain()
{
    InitializeComponent();
}

private void LoadIdentityData()
{
    Id citizen = EIDPT.GetID();
    Picture picture = EIDPT.GetPicture();
    System.IO.MemoryStream ms = new System.IO.MemoryStream(picture.Bytes, 0, picture.BytesLength, false);
    Image tempImage = CSJ2K.J2kImage.FromStream(ms);
    ms.Close();
    pbPhoto.Image = tempImage;
    txtLastName.Text = citizen.Name;
    txtFirstName.Text = citizen.FirstName;
    txtDocumentNumber.Text = citizen.BI;
}

private void frmMain_Load(object sender, EventArgs e)
{
    this.StartPosition = FormStartPosition.Manual;
    this.Location = new Point(Screen.PrimaryScreen.WorkingArea.Width - this.Width, Screen.PrimaryScreen.WorkingArea.Height - this.Height);
    scWatcher = SCWatcher.GetInstance();
    this.currentReader = string.Empty;
    this.readerList = new List<string>(scWatcher.ListReaders());
    if (this.readerList.Count > 0)
        this.currentReader = this.readerList[0];
    scWatcher.CardInserted += new SCWatcher.CardInsertedHandler(scWatcher_CardInserted);
    scWatcher.CardRemoved += new SCWatcher.CardRemovedHandler(scWatcher_CardRemoved);
    scWatcher.ReaderInserted += new SCWatcher.ReaderInsertedHandler(scWatcher_ReaderInserted);
    scWatcher.ReaderRemoved += new SCWatcher.ReaderRemovedHandler(scWatcher_ReaderRemoved);
    scWatcher.OnError += new SCWatcher.ErrorHandler(scWatcher_OnError);
    EIDPT eId = EIDPT.GetInstance();
}
```

Aplicação .NET C#

De seguida, podemos ver a aplicação exemplo criada com base no modelo descrito anteriormente. As funcionalidades que possui são simples: autenticação com os diferentes códigos PIN e verificação de informação referente ao cartão.

Aplicação Java (*pteidlibj.jar*)

Para integrar o cartão do cidadão numa aplicação java usando a biblioteca *pteidlibj.jar* é necessário ter a aplicação oficial do cartão do cidadão instalada. Para além disto, em Linux, é necessário especificar a localização do ficheiro *libpteidlibj.so* da aplicação resultante da instalação da aplicação oficial.

O exemplo produzido permite a recolha de informação dos dados do utilizador e da morada do mesmo, sendo que para este último é necessário introduzir a palavra passe da morada.