



favorito (11)

imprimir

anotar

marcar como lido

dúvidas?

Utilizando Biometria em .NET

Veja neste artigo como utilizar a Biometria, Usaremos neste artigo o leitor Hamster I (HFDU-01 – USB – 500 dpi) da empresa Nitgen e sua API de desenvolvimento, que possui suporte a diversas linguagens, inclusive .NET.

(11) (0)

Biometria são métodos de reconhecimento de pessoas, baseados em características físicas e comportamentais. Essas características são classificadas como: face, impressão digital, íris, mão, assinatura e voz. Na **Tabela 1** temos a descrição de cada característica com relação a facilidade de uso, eficiência e



Publique | Assine | Fale conosco

Curtir Página

Hospedagem web por **Porta 80 Web Hosting**

acesso físico e lógico, operadoras de planos de saúde etc.

Utilizada independente ou integrada com outras tecnologias como SmartCards, chaves de criptografias e assinaturas digitais, a biometria está se fixando como uma tecnologia de economia e segurança para nossas vidas.

Utilizada na autenticação pessoal, está ficando cada vez mais conveniente e segura que os tradicionais métodos de autenticação, como a utilização de senhas. Isso porque une o evento a um único indivíduo particular (diferentemente das senhas, as digitais são únicas para cada pessoa), é conveniente (nada para carregar ou lembrar), preciso (utiliza-se de alta tecnologia para autenticação), e o melhor de tudo, está cada vez mais acessível aos usuários, em virtude das constantes reduções de custos.

Estaremos abordando neste artigo a biometria de reconhecimento da digital (fingerprint), que por sua vez é a biometria mais utilizada no Brasil e no mundo. Nas tecnologias existentes de reconhecimento da digital, existem várias formas de leituras, e a mais utilizada no mercado é a óptica, que gera uma imagem de cinzas com uma escala representada por 8 bits, com cerca de 500x500 pixels.

Usaremos neste artigo para teste, o leitor Hamster I (HFDU-01 - USB - 500 dpi) da empresa Nitgen e sua API de desenvolvimento, que possui suporte a diversas linguagens, inclusive [.NET](#).

Tendo em vista a apresentação prática da biometria, mostraremos como utilizar as funções mais importantes da **API**, de modo que os exemplos não contenham nenhuma restrição para utilizar em outras aplicações.

Reconhecimento da digital

Em um sistema biométrico, uma característica física precisa ser registrada

Finalmente quando se faz a autenticação de uma pessoa (Verification) completa-se o ciclo (veja a descrição dos processos do ciclo na **Tabela 2**).

Processo	Descrição
Enrollment	Ato de gerar um registro dentro do sistema biométrico.
Template	Uma representação matemática da característica física capturada.
Bometric algorithm	Conjunto de técnicas lógicas embutidas em um processo para obtenção de algum resultado.
Verification	Processo de comparação entre duas templates, para verificar a autenticidade do indivíduo.

Tabela 2: Processos de reconhecimento da digital

FAR, FRR e FTE

O desempenho de uma medida de biometria normalmente se refere para termos de FAR (taxa de falso aceite), FRR (taxa de falsa rejeição) e a FTE (taxa de falso fracasso). A medida FAR é a porcentagem de usuários inválidos que são aceitos incorretamente com se fossem usuários verdadeiros.

Enquanto que o FRR mede a porcentagem de usuários válidos que são rejeitados como sendo usuários impostores. Por último quando você é novo em um sistema de biometria e precisa se registrar e não consegue, o FTE mede essa porcentagem de usuários.

Trabalhando com a API Nitgen

Descreveremos como fazer uma programação em .NET com a classe NBioBSP que

NBioBSP.dll e provê alto nível de interfaces.

Enquanto que um grande número de linguagens de programação é suportada pelo .NET, essa seção do artigo vai fazer uma introdução usando a linguagem C#. Para tanto devemos conhecer a utilização de alguns dos principais métodos da API para utilização do leitor biométrico. Veja na **Listagem 1** o trecho de código para instanciar o classe NBioBSP.

Listagem 1: Inicializar a classe NBioBSP

```
using NITGEN.SDK.NBIOBSP;

...

public void Inicializar()
{
    NBioAPI m_NBioAPI;
    m_NBbioAPI = new NBioAPI();
}
```

A inicialização do dispositivo, dever ser usada com o método OpenDevice, antes da utilização da funções de cadastramento, verificação e captura. O dispositivo pode ser selecionado automaticamente usando o NBioAPI.Type.DEVICE_ID.AUTO. Veja na **Listagem 2** a utilização do OpenDevice.

Listagem 2: Inicializando o dispositivo biométrico

```
public void InicializarDispositivo()
{
    uint ret m_NBioAPI.OpenDevice(NBioAPI.Type.DEVICE_ID.AUTO);
    if (ret == NBioAPI.Error.NONE)
        //leitor inicializado com sucesso
    else
        //leitor não inicializado
}
```

Para fechar o dispositivo, o método CloseDevice de ser utilizado. Como parâmetro usaremos novamente o BioAPI.Type.DEVICE_ID.AUTO. Na **Listagem 3** é mostrado a

Listagem 3: Fechando o dispositivo biométrico

```
public void FecharDispositivo()
{
    uint ret m_NBioAPI.CloseDevice(NBioAPI.Type.DEVICE_ID.AUTO);
    if (ret == NBioAPI.Error.NONE)
        //leitor fechado com sucesso
    else
        //leitor não fechado
}
```

Antes de usar o `OpenDevice`, você pode usar uma função chamada `UnenumerateDevice` que determina o número e o tipo de leitor conectado ao computador. Uma vez chamada, o número de dispositivos e o ID de cada dispositivo serão retornados pela função. Dessa forma poderia abrir e fechar o dispositivo com esse ID retornado, ao invés de usar o `BioAPI.Type.DEVICE_ID.AUTO`, como parâmetro de `OpenDevice` e `CloseDevice`.

O `Enroll` é usado para registrar digitais. Os tipos de dados usado pela `NBioBSP` são do tipo `handle`, binário e texto codificado. Os dados da digital será colocado dentro de um `handle` de `FIR` (Fingerprint Identification Record), propriamente depois de um registro completo das digitais pelo `Enroll`, e depois pode retornar um tipo binário ou texto codificado.

A `NBioBSP` possui vários `Enroll Overloading` que pode ser usado para diversos propósitos. Veja na **Listagem 4** um exemplo dessa utilização.

Listagem 4: Utilizando um dos metodos `Enroll`

```
public void Cadastrando_Enroll()
{
    NBioAPI.Type.HFIR hNewFIR;
    NBioAPI.Type.FIR_TEXTENCODING m_textFIR;
    InicializarDispositivo();
    uint retEnroll = m_NBioAPI.Enroll(out hNewFIR, null);
}
```

```
return,  
}  
  
FecharDispositivo();  
m_NBioAPI.GetTextFIRFromHandle(m_hNewFIR, out m_textFIR, true);  
//gravar no BD  
}
```

Veja na **Tabela 3** a estrutura de um registro FIR.

4 bytes		20 bytes			Tamanho variável	
Formato		Cabeçalho			Dados da impressão digital	
Estrutura do Cabeçalho						
Tamanho do cabeçalho	Tamanho dos dados	Versão	Tipos de dados	Propósito	Qualidade	Reservado
4 bytes	4 bytes	2 bytes	2 bytes	2 bytes	2 bytes	4 bytes

Tabela 3: Estrutura do registro FIR

Verificação de impressão digital

Para executar a verificação das impressões digitais existe um método chamado `Verify` que compara uma impressão digital existente com uma impressão digital capturada e retorna o resultado da verificação.

Depois da verificação, é armazenada dentro da FIR uma chave de criptografia que é chamada `Payload`. Também o `Payload` pode armazenar outras informações como senhas ou nomes de usuários. Veja na **Listagem 5**, o exemplo de uma verificação.

Listagem 5: Executando uma verificação entre impressões digitais

```
bool result,
NBioAPI.Type.FIR_TEXTENCODING m_textFIR;

InicializarDispositivo();

NBioAPI.Type.FIR_PAYLOAD myPayload = new NBioAPI.Type.FIR_PAYLOAD();

uint retVerifica = m_NBioAPI.Verify(m_textFIR, out result, myPayload);

if (retVerifica != NBioAPI.Error.NONE)
{
    MessageBox.Show("Erro na verificação!");
    return;
}

if (result)
{
    if (myPayload.Data != null)
    {
        MessageBox.Show("Payload Encontrado");
    }
    else
    {
        MessageBox.Show("Payload não encontrado");
    }

    MessageBox.Show("Digitais conferem");
}
else
{
    MessageBox.Show("Digitais não conferem");
}
}
```

Conclusão

Neste artigo fizemos uma pequena demonstração de como utilizar a biometria por reconhecimento da digital. Como não foi abordado todos os recursos da classe NBioAPI, a intenção é mostrar aos desenvolvedores os comandos mais importantes a facilidade de desenvolvimento com a tecnologia.



da biometria continuará se expandindo como a indagação humana para conhecimento e estudo contínuo.

Publicado no [Canal .net](#)



por DevMedia Bottcher

Guru .net e tecnologias MS

Ajude-nos a evoluir: você gostou do post? (11) (0)

Compartilhe:

Ficou com alguma dúvida?



Poste aqui sua dúvida ou comentário que nossa equipe responderá o mais rápido possível.



Marcelo Silva

Bom dia! Estive lendo o post, que é muito bom, e pretendo implementar esta solução em uma aplicação Asp.net. Se o código está implementado no lado servidor, como seria possível fazer a captura das digitais no lado cliente enviando as mesmas para o servidor para gravação em BD? abs!

há +1 mês



Henrique Gasparotto

CONSULTOR

Olá Marcelo, obrigado pelo seu comentário.

Você salvaria o valor do FIR na base de dados. O ideal é fazer uma chamada AJAX para essa captura e salvar esses dados no banco. Assim você consegue fazer essa conexão entre o cliente e o servidor de sua aplicação.

A forma depende se você está utilizando MVC ou WebForms.

**Raul Guilherme**

Boa Tarde, gostei muito do post.

Mas na hora de aplicar surgiu um problema, não sei se os colegas passaram por isso, instalei os drives do Nitgen tudo certo, coloquei a NITGEN.SDK.NBioBSP.dll no meu projeto, reconheceu as classes e etc. Porém quando compilo o projeto quando vou Instanciar o objeto NBioAPI da o seguinte erro : Unable to load DLL 'NBioBSP.dll': Não foi possível encontrar o módulo especificado. (Exception from HRESULT: 0x8007007E). Alguem pode me ajudar ?

Agradeço desde já,
Abs

há +1 ano

**Douglas Claudio**

Olá Raul, obrigado pelo seu comentário.

Enviamos sua solicitação a Equipe DevMedia e estamos no aguardo de um feedback do mesmo.

Um abraço.

há +1 ano

**Diogo Souza**

Olá Raul,

O seu erro pode ter algo a ver com um glitch da DLL, ou seja, o programa não a está encontrando por razões desconhecidas...

Tenta um dos passos:

1. Reinicie o computador, pode ser algo temporário;
2. Tenta recuperar essa DLL da pasta recycle do seu Windows e substituir pela que está usando;
3. Caso não consiga, baixa a ferramenta fix tool que te ajudará a reparar estes tipos de erro, em <http://www.fixerrorhelps.com/download.php>

Se nada funcionar, verifique se o seu Windows não tem nenhuma atualização pendente, isso pode te ajudar a atualizar o SO e suas DLLs.

há +1 ano



Você teria algum material legal para programadores Java EE com utilização de banco de dados?

há +1 ano



Wesley Yamazack

Olá Thiago,
entramos em contato com o autor para que ele possa te auxiliar, tudo bem ? Peço que aguarde um pouco até que o mesmo possa retornar o contato.

Obs: Este é um post antigo pode ser que o autor demore um bom tempo para responder.

Um abraço

há +1 ano



Pedro Filho

Thiago

Recomendo a utilização da documentação oficial java EE:

<http://docs.oracle.com/javaee/6/tutorial/doc/>

há +1 ano



Fabrício Damasceno

Boa tarde, Pedro.

Gostei do manual que você passou sobre Java EE 6 e gosto muito das suas aulas. Sei que na programação, tudo é possível para aquele que estuda, mas você dar um exemplo de como ficaria com Java?

O próprio manual do fabricante não tem nada falando sobre java.

Acho que seria interessante.

Obrigado Pedro.

há +1 ano



Diogo Souza

Opa Fabrício,

A biometria está relativa às dlls e drivers que o seu proprietário disponibiliza. Que tipo de hardware está usando? Já verificou se existe algum driver de comunicação pro mesmo?

Dá uma olhada também nessa thread aqui no fórum da DevMedia sobre o assunto:

<http://www.devmedia.com.br/forum/biometria-java/417388>

há +1 ano

DEVMEDIA



Compartilhado pela área.

há +1 ano

Adicionar um comentário...

Mais posts

Artigo

Cortana API: Implementando reconhecimento de voz para apps universais

Artigo

Desenvolva aplicações universais com o Windows App Studio

Artigo

Gerando boletos com a biblioteca Boletos.net

Artigo

Criando apps com UWP e SQLite para Windows 10

Revista

Revista .net Magazine 130

Listar mais conteúdo

DEVMEDIA

