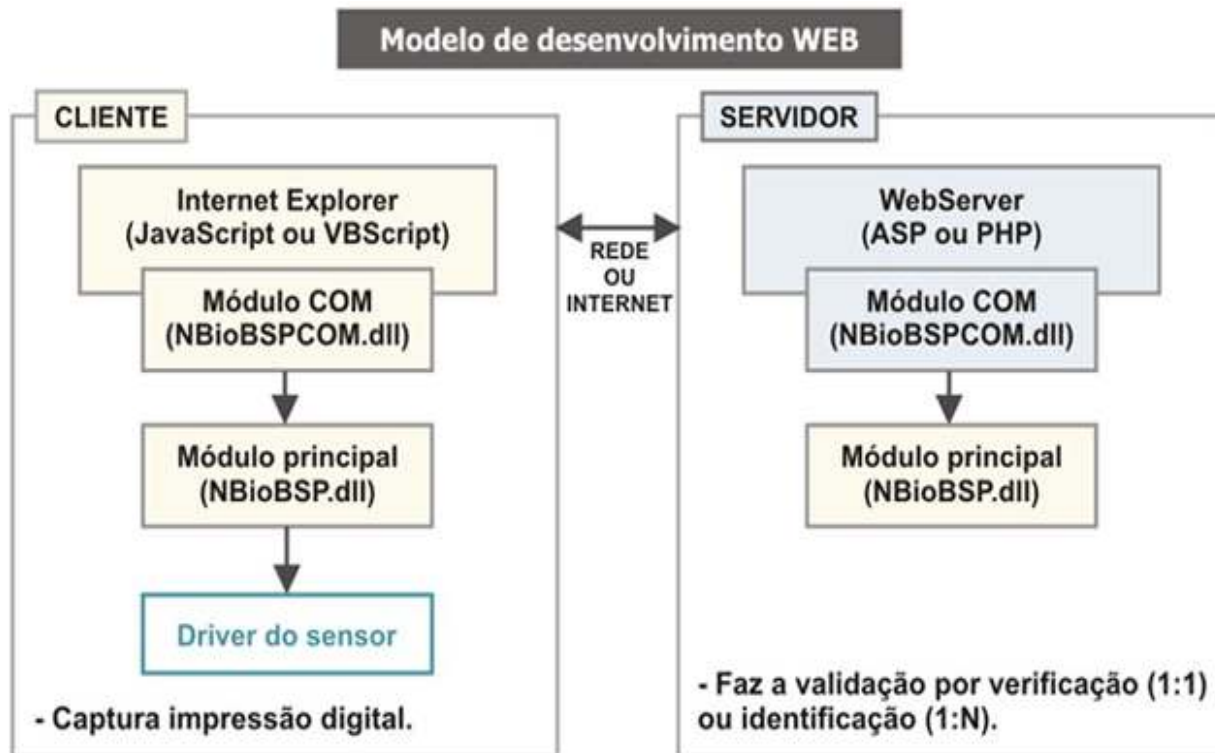


Índice:

CAPÍTULO 4. PROGRAMAÇÃO EM ASP	2
4.1 REGISTRAR.....	2
4.1.1 Códigos para configuração do objeto	2
4.1.2 Formulário para transferir dados da impressão digital	3
4.1.3 Código Javascript para registro da impressão digital.....	3
4.1.4 Armazenando a informação da impressão digital	5
4.2 VERIFICAÇÃO	5
4.2.1 Código para configuração do object	5
4.2.2 Formulário para transferir os dados da impressão digital	5
4.2.3 Código Javascript para capturar as impressões digitais	6
4.2.5 Comparando com uma impressão digital existente (1:1)	7
4.2.4 Comparando com uma impressão digital existente (1:N).....	8
4.2.5 Mudando a interface de usuário do NBioAPI	8

Capítulo 4. Programação em ASP

O módulo **NBioBSP COM** executa programas em ambientes Web sem nenhuma modificação adicional, devido ao fato de que é construído na arquitetura Microsoft COM. Pode também ser usado simultaneamente em Servers e Clients porque inclui componentes de Server e funções ActiveX para Client.



4.1 Registrar

A impressão digital do usuário deve ser registrada pelo Web Browser no lado cliente em Java Script ou VB Script. Os dados da impressão digital devem ser salvos em um documento ou banco de dados no lado servidor.

4.1.1 Códigos para configuração do objeto

O módulo **NBioBSP COM** deve ser configurado para cada página HTML.

```
<OBJECT classid="CLSID: F66B9251-67CA-4d78-90A3-28C2BFAE89BF"  
        height=0 width=0  
        id="objNBioBSP"  
        name="objNBioBSP">  
</OBJECT>
```

O nome demonstrado acima será usado como nome do objeto no Javascript.

4.1.2 Formulário para transferir dados da impressão digital

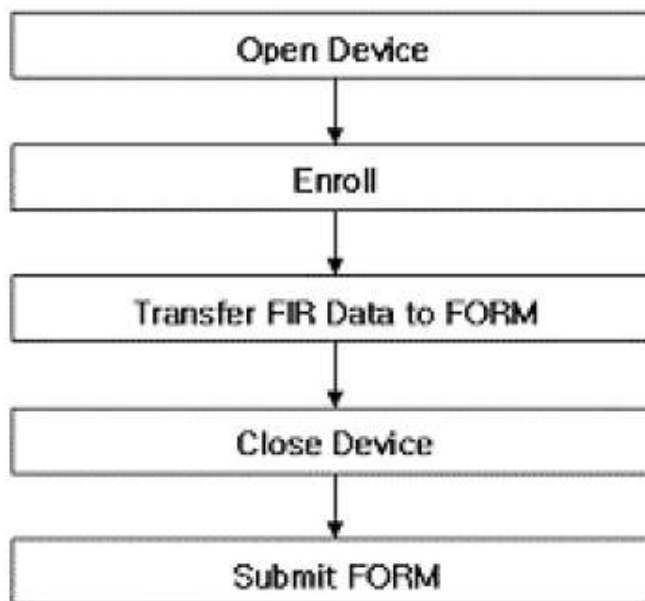
Os dados da impressão digital, registrados no Javascript (lado cliente), podem transferidos para o servidor através do formulário exemplo existente no eNBSP SDK em ASP.

```
<form action='regist.asp' name='MainForm' method='post' OnSubmit='return regist();'>  
<input type='hidden' name='FIRTextData'>  
User ID : <input type='text' name='UserID' size=20><p>  
<input type='submit' value=' Click here to register your fingerprint '>  
</form>
```

Ao selecionar este formulário, o Javascript chama a função `regist()` e executa o registro. A impressão digital registrada será transferida para o texto do **FIRTextData**.

4.1.3 Código Javascript para registro da impressão digital

O código Javascript será usado para fazer a comunicação entre o Web Browser e o módulo **NBioBSP COM**, ambos no lado cliente. O método **Enroll** também pode ser usado.



[Process flow in the Regist function]

O exemplo mostra como as funções podem ser compostas. Note que o Javascript é *case sensitive*.

```
<script lang='javascript'>  
  
function regist()  
{  
    var err, payload;  
    // Check ID is not NULL
```

```

if ( document.MainForm.UserID.value == " )
{
    alert("Please enter user id !");
    return(false);
}
try // Exception handling
{
    // Open device. [AUTO_DETECT]
    // You must open device before enroll.
    DEVICE_FDP02 = 1;
    DEVICE_FDU01 = 2;
    DEVICE_AUTO_DETECT = 255;

    Var objDevice = document.objNBioBSP.Device;
    Var objExtraction = document.objNBioBSP.Extraction;

    objDevice.Open(DEVICE_AUTO_DETECT);

    err = document.objNBioBSP.ErrorCode; // Get error code

    if ( err != 0 ) // Device open failed
    {
        alert("Device open failed !");
        return(false);
    }
        // Enroll user's fingerprint.
    document.objNBioBSP.Enroll(payload);
    err = document.objNBioBSP.ErrorCode; // Get error code

    if ( err != 0 ) // Enroll failed
    {
        alert("Registration failed ! Error Number : [' + err + ']");
        objDevice.Close(DEVICE_AUTO_DETECT);
        return(false);
    }
    else // Enroll success
    {
        // Get text encoded FIR data from NBioBSP module.
        document.MainForm.FIRTextData.value =
        objExtraction.TextEncodedFIR;
        alert("Registration success !");
    }
        // Close device. [AUTO_DETECT]
    objDevice.Close(DEVICE_AUTO_DETECT);
    objExtraction = 0;
    objDevice = 0;
    }
    catch(e)
    {
        alert(e.message);
        return(false);
    }
        // Submit main form
    document.MainForm.submit();
    return(true);
}

```

```
}  
</script>
```

4.1.4 Armazenando a informação da impressão digital

Salvar em arquivo ou no servidor em um banco de dados as informações de UserID e FIRTextData as quais representam o ID do usuário e o template capturado (em formato de string de caracteres).

```
<%  
UserID      = Request.Form("UserID")  
FIRTextData = Request.Form("FIRTextData")  
' Write UserID and FIRTextData to File or DB.  
%>
```

4.2 Verificação

A impressão digital capturada poder ser transferida a partir do Web Browser no lado cliente e comparada com uma impressão digital já cadastrada no bando de dados no lado servidor.

4.2.1 Código para configuração do object

O módulo NBioBSP COM deve ser configurado para cada página HTML para ser usado através Browser da Web. É esta referencia que garante a execução do código em Javascript.

```
<OBJECT classid="CLSID: F66B9251-67CA-4d78-90A3-28C2BFAE89BF"  
height=0 width=0  
id="objNBioBSP"  
name="objNBioBSP">  
</OBJECT>
```

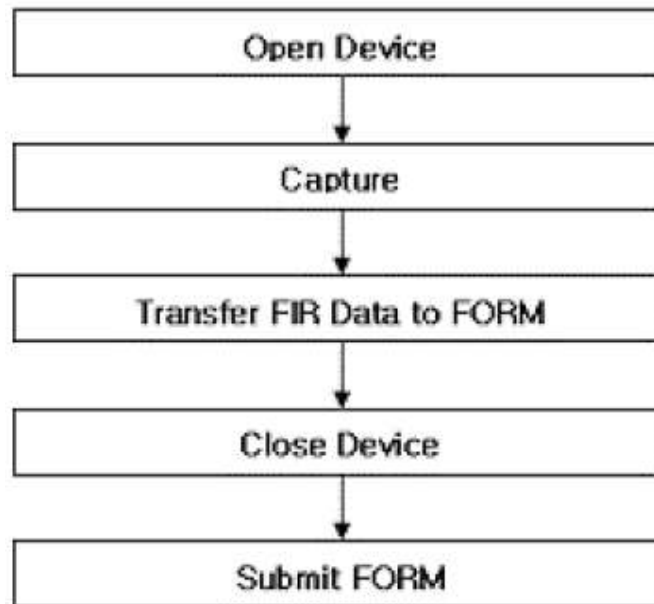
4.2.2 Formulário para transferir os dados da impressão digital

Os dados da impressão digital capturada pelo Javascript podem ser transferidos para o servidor da seguinte forma: a função em Javascript para captura de impressão digital é chamada quando o Form é submetido.

```
<form action='verify.asp' name='MainForm' method='post' OnSubmit='return capture();'>  
<input type='hidden' name='FIRTextData'>  
User ID : <input type='text' name='UserID' size=20>  
<p>  
<input type='submit' value=' Click here to verification with your fingerprint '>  
</form>
```

Ao clicar o botão, a impressão digital é capturada através da função **capture()** no Javascript e depois transferida como texto no **FIRTextData**.

4.2.3 Código Javascript para capturar as impressões digitais



[Process flow in the capture function]

O fluxograma mostra como a execução pode ser composta.

```
<script lang='javascript'>
function capture()
{
var err
    // Check ID is not NULL
if ( document.MainForm.UserID.value == " )
{
alert("Please enter user id !");
return(false);
}
Try    // Exception handling
{
    // Open device. [AUTO_DETECT]
    // You must open device before capture.
    DEVICE_FDP02 = 1;
    DEVICE_FDU01 = 2;
    DEVICE_AUTO_DETECT = 255;
    Var objDevice = document.objNBioBSP.Device;
    Var objExtraction = document.objNBioBSP.Extraction;
    objDevice.Open(DEVICE_AUTO_DETECT);
    err = objDevice.ErrorCode; // Get error code
    if ( err != 0 ) // Device open failed
    {
    alert("Device open failed !");
    return(false);
    }
    }
}
}
</script>
```

```

    }
    // Enroll user's fingerprint.
    objExtraction.Capture();
    err = objExtraction.ErrorCode; // Get error code
    if ( err != 0 ) // Enroll failed
    {
        alert('Capture failed ! Error Number : [' + err + ']');
        objDevice.Close(DEVICE_AUTO_DETECT);
        return(false);
    }
    else // Capture success
    {
        // Get text encoded FIR data from NBioBSP module.
        document.MainForm.FIRTextData.value=
        objExtraction.TextEncodedFIR;
        alert('Capture success !');
    }
    // Close device. [AUTO_DETECT]
    objDevice.Close(DEVICE_AUTO_DETECT);
    objExtraction = 0;
    objDevice = 0;
} // end try
catch(e)
{
    alert(e.message);
    return(false);
}
// Submit main form
document.MainForm.submit();
return(true);
}
</script>

```

4.2.5 Comparando com uma impressão digital existente (1:1)

Use o método **VerifyMatch** para ler uma impressão digital já existente e compare com uma impressão digital previamente capturada. O resultado pode ser encontrado nas propriedades do **MatchingResult**.

```

<%
' Read FIR data from file or DB.
Set objNBioBSP = Server.CreateObject("NBIOBSPCOM.NBioBSP")
Set objMatching = objNBioBSP.Matching
' Verify Match
' FIRTextData is Caputured FIR
' fFIRTextData is FIR from file
Call objMatching.VerifyMatch(CStr(FIRTextData), CStr(fFIRTextData))
if ( objMatching.MatchingResult = 0 ) then
' Verification failed !!!
else
' Verification success !!!
end if
' Release NBioBSP object

```

```
Set objMatching = nothing
Set objNBioBSP = nothing
%>
```

4.2.4 Comparando com uma impressão digital existente (1:N)

Use o método IndexSearch para armazenar e identificar os templates. No objeto IndexSearch criado, devem ser adicionados o template (impressão digital) e o id dos usuários. Se estes dados estiverem armazenados em um DB, então devem ser carregado no IndexSearch através de uma rotina de repetição, até que todos os templates e IDs do banco forem adicionados.

O processo de identificação ocorrerá diretamente com os dados armazenados neste objeto (na memória) e não no DB, é este processo que torna a busca instantânea.

O resultado da identificação será o ID do template identificado

Adicionando no IndexSearch :

```
<%
' Read FIR and ID from file or DB.
Set objNBioBSP = Server.CreateObject("NBIOBSPCOM.NBioBSP")
Set objIndexSearch= objNBioBSP. objIndexSearch
' nUserID is the User ID
' fFIRTextData is FIR from file
call objIndexSearch.AddFIR(fFIRTextData, nUserID)
%>
```

Identificando:

```
<%
Set objNBioBSP = Server.CreateObject("NBIOBSPCOM.NBioBSP")
Set objIndexSearch= objNBioBSP. objIndexSearch

call objDevice.Open(255)
call objExtraction.Capture(1)
call objDevice.Close(255)
// Faz a identificação do usuário. 1º Parametro: String capturada a identificar. 2º Parametro:
Nível de segurança (Varia de 1 à 9).
call objIndexSearch.IdentifyUser(objExtraction.TextEncodeFIR, 6)
// objIndexSearch.UserID irá retornar o ID do user identificado. Com este valor deve-se fazer a
busca no Database dos demais dados do usuário.
Response.Write(objIndexSearch.UserID);
%>
```

4.2.5 Mudando a interface de usuário do NBioAPI

O módulo **NBioBSP.COM** oferece recursos para customização da UI (user interface) básica. Use o método **SetSkinResource** para carregar os recursos da UI (Pop-UP) em diversas línguas.

Para que o Skin (Pop-UP) seja utilizado, inclua somente a linha abaixo no código

especificando o local da DLL no PC cliente. Salvar DLL em: C:\WINDOWS\system32.

```
// Set skin resource
<script lang='javascript'>
var objNBioBSP = new ActiveXObject('NBioBSPCOM.NBioBSP.1');
objNBioBSP.SetSkinResource('NBSP2Por.dll');
...
</script>
```