

## Componente de Autenticação do Launcher para o Delphi

<u>Revisão</u>	<u>Versão</u>	<u>Data</u>	<u>Autor</u>
1	2018.2.3	01/02/2018	Daniel Júnior ( <a href="mailto:Daniel.junior@engegraph.com.br">Daniel.junior@engegraph.com.br</a> )
2	2018.2.4	09/02/2018	Daniel Júnior ( <a href="mailto:Daniel.junior@engegraph.com.br">Daniel.junior@engegraph.com.br</a> )

O Objetivo do componente é o de encapsular a autenticação dos sistemas da Engengraph executados a partir do **launcher**.

Pré requisito mínimo para instalação:

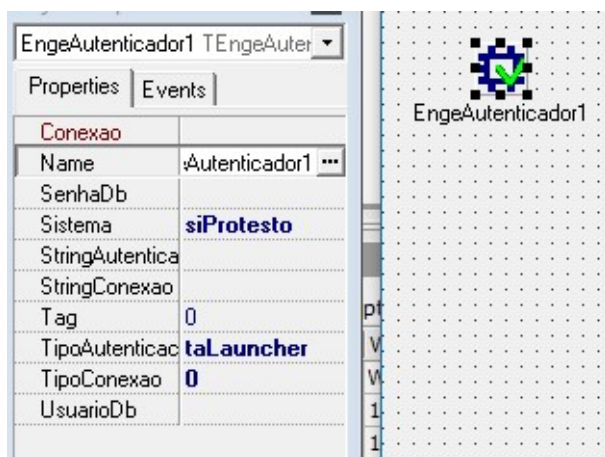
**Delphi 7**

**Indy Library**, versão 10.5432 ou superior:

Download do Indy e instruções de instalação no link abaixo:

<https://www.evernote.com/shard/s275/sh/7536c345-8023-4648-9b22-1f7060416df3/f057f75aff2c2251>

Após instalado, o componente, pode ser localizado na paleta EngLib:



### Como usar

- Carregue as informações do driver e configurações da Conexão no Componente de conexão (**SqlConnection**, da paleta **DbExpress**);
- Informe o **SqlConnection**, na propriedade Conexão, no **object inspector**;
- Informe um tipo de autenticação na propriedade **TipoAutenticacao**;

## Métodos, Propriedades e eventos

Propriedade	Descrição
<b>SenhaDb *</b>	Senha do usuário da conexão com o banco de dados
<b>Sistema *</b>	Selecionar o sistema (RI, Notas...etc)
<b>StringAutenticacao</b>	String de autenticação Base64 criptografada que é enviada pelo launcher com os dados da conexão/ usuário
<b>StrinConexao *</b>	String de conexão com o banco de dados. Observação: para o Sql Server: Server=[SERVER];Database=[NOME-BANCO]
<b>TipoAutenticacao</b>	Tipo de autenticação: taLauncher = Autenticação pela string informada na propriedade StringAutenticacao; taDebug = Autenticação manual, usada pelo desenvolvedor em tempo de projeto para executar o sistema sem o launcher (testes de desenvolvimento)
<b>TipoConexao</b>	0 = firebird, 1 = sql server, 2 = Oracle
<b>UsuarioDb *</b>	Usuário de conexão com o banco de dados

\* Propriedades para o modo Debug

```
//Para definir o tipo de autenticação
```

```
EngeAutenticador1.TipoAutenticador := taLauncher;
```

```
(* Receber a string de autenticação enviada pelo Launcher (Considerando que a mesma é o primeiro parâmetro *)
```

```
EngeAutenticador.StringAutenticacao := StrParams();
```

```
//Autenticar
```

```
EngeAutenticador1.Autenticar();
```

```
//testa se o autenticador está válido
```

```
EngeAutenticador1.Autenticado;
```

```
(* Exiba mensagem de acesso ao sistema restrito (Parte integrante dos sistemas da engeraph). Este tem por objetivo exibir mensagem ao usuário, mostrando q o sistema só deverá ser executado pelo Launcher *)
```

```
EngeAutenticador1.ExibeAcessoRestrito();
```

## Evento OnErro

Usado para capturar um erro ao autenticar. O parâmetro **e** encapsula a exceção, exemplo:

```
ShowMessage(e.message); //exibe a mensagem de erro
```

### Evento OnAntesAutenticar

Executa antes de autenticar

### Evento OnDepoisAutenticar

Executa código depois de autenticar

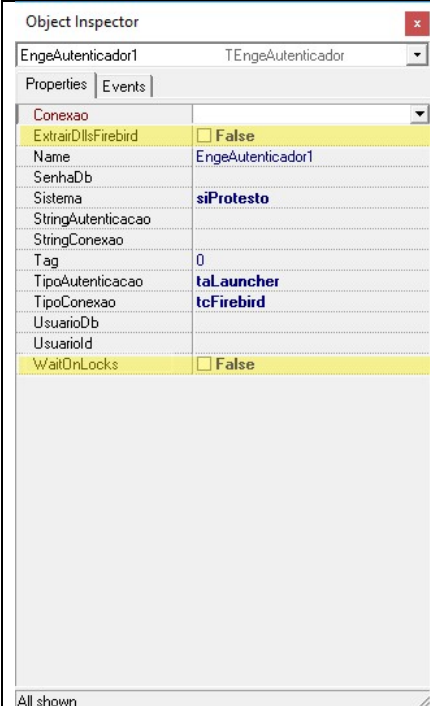
### Evento OnExibirAcessoRestrito

Executa código ao exibir a mensagem de acesso restrito

## Novas Propriedades e outras alterações do componente

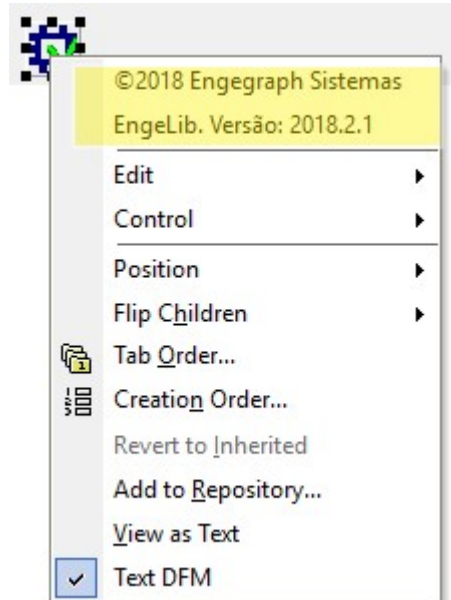
Daniel Júnior em 01/02/2018

### Novas propriedades incluídas no componente

	<p><b>ExtrairDllsFirebird:</b> Quando for necessário extrair dinamicamente as dlls do firebird de acordo com a sua versão (1.5 ou 2.5) . As dlls são extraídas para o diretório de execução do aplicativo.</p> <p><b>WaitOnLocks:</b> Parâmetro setado na conexão DbExpress, aparentemente nem todos os sistemas da empresa são configurados igualmente, por isso foi criada esta opção</p>
--	---

## Exibição da versão do Componente

Ao clicar com o botão direito sobre o componente de autenticação no menu de contexto do Delphi será exibida a informação da versão do componente, conforme abaixo:



## Informação sobre enumerados:

Para usar todos os enumerados do compoente, no seu código, referenciar na cláusula uses a unit **untEngeEnumerados**.

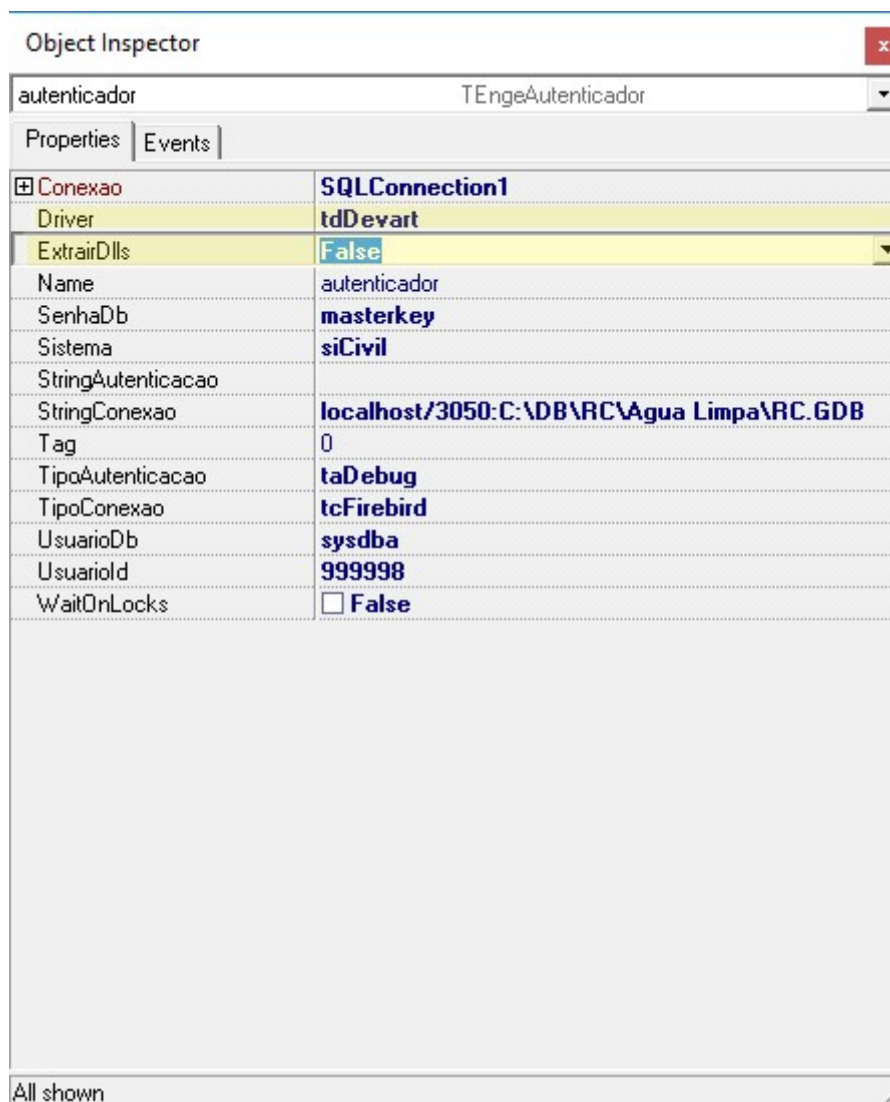
### Enumerados:

TTipoAutenticacao = (taLauncher, taDebug);

TTipoConexao = (tcFirebird = 0, tcMSSQL = 1, tcOracle = 2);

TSistema = (siProtesto, siRtd, siNtcs, siRics, siCivil);

## Integração do componente com o DevArt (09/02/2018)



Para a utilização do componente DevArt foi criada a propriedade “Driver”. Esta possui duas opções: tdNativo e tdDevart (unit **untEngEnumerados**).

tdNativo: Mantem o driver nativo dbexpress do Delphi

tdDevart: Usa as configurações do DevArt.

## Alteração ExtrairDlls (09/02/2018)

Anteriormente a propriedade “ExtrairDlls” era nomeado como “ExtrairDllsFirebird”, pois estava “tipado” diretamente ao banco de dados Firebird. Com a inclusão do Devart se fez necessária a utilização do método para extrair as Dll’s do Devart para Sql Server e futuramente Oracle (Se necessário).