

# TUsersCS Para Delphi 3/4

## Manual do Usuário por Tools&Comps

Tools&Comps Home Page: [www.gtc.com.br/~jgkrs/tandco.htm](http://www.gtc.com.br/~jgkrs/tandco.htm)

Tools&Comps e-mail: [tandco@gtc.com.br](mailto:tandco@gtc.com.br)

**VISÃO GERAL.....3**

**INSTALANDO O TUSERSCS.....4**

**COMO FUNCIONA O COMPONENTE TUSERSCS.....4**

1. CRIANDO AS TABELAS DE USUÁRIOS.....4

2. USANDO O TUSERSCS DENTRO DO IDE DO DELPHI.....4

3. OUTRAS CSPRIEDADES DO TUSERSCS.....5

4. CSPRIEDADES DA CLASSE TUSER.....5

5. MÉTODOS DO TUSERSCS.....5

**SUPORTE A OUTROS IDIOMAS.....6**

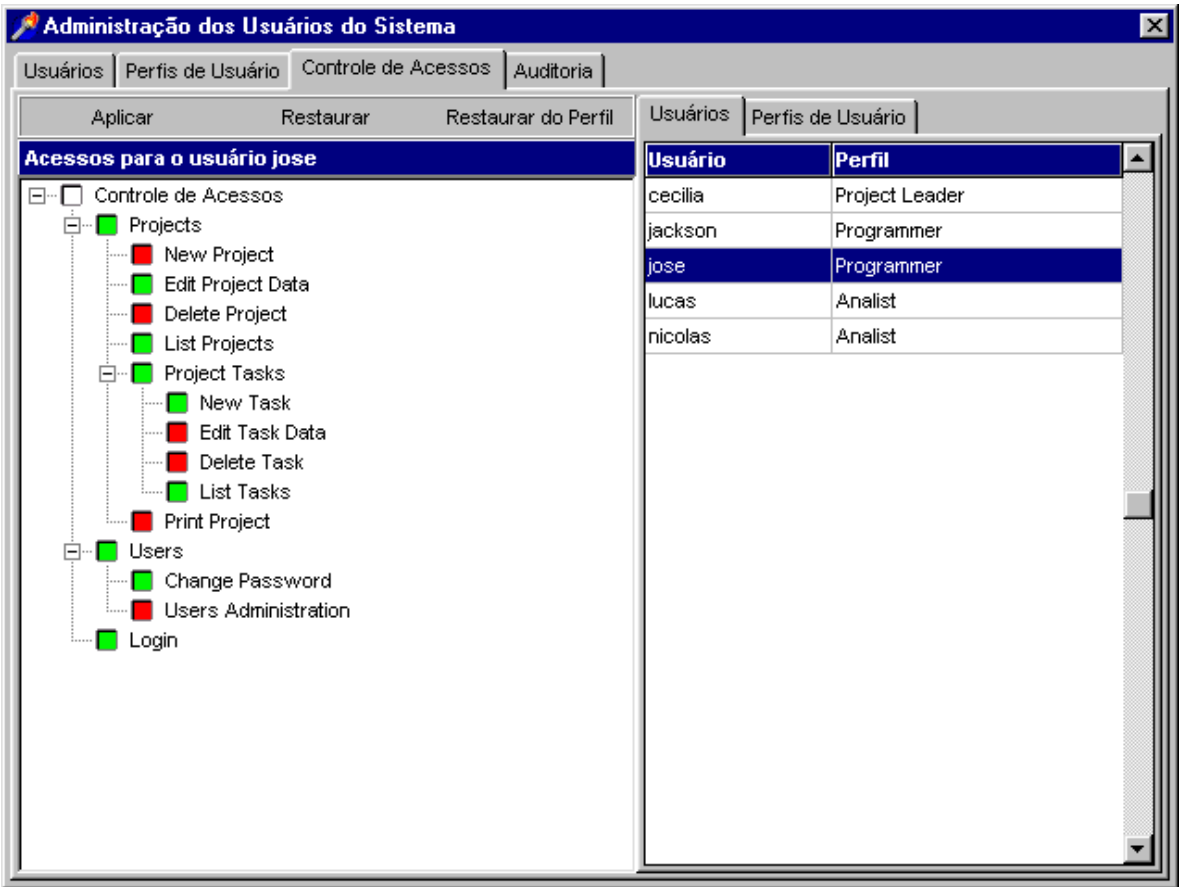
**REGISTRANDO O TUSERSCS.....6**

# VISÃO GERAL

**TUsersCS** (versão **Client/Server**) é um componente que torna simples implementar controle de usuários em programas escritos em Delphi. Com uma pequena configuração a nível de componente em tempo de design, sua aplicação estará pronta para controlar o acesso de seus usuários. Com o **TUsersCS**, você não precisa escrever nenhuma linha de código para implementar o controle de usuários. Basta você conectá-lo ao componente **TMenu** de seu formulário principal e deixar o resto por conta do **TUsersCS**.

Através da conexão com o menu do formulário principal de sua aplicação, o **TUsersCS** permite que o administrador de segurança de seu sistema conceda permissões de acesso ao usuário a cada item de menu. Todos os acessos do usuário são mostrados em uma árvore que imita a hierarquia de menus de seu programa. Basta o administrador clicar no item desejado para conceder ou retirar o acesso do usuário para esse item. Quando o usuário logar no sistema, o **TUsersCS** se encarrega de desabilitar ou tornar invisível (de acordo com o perfil ao qual o usuário pertença) os itens de menu aos quais o usuário não tem acesso.

O componente **TUsersCS** conta ainda com um módulo de auditoria, que permite verificar a utilização dos programas pelos usuários.



O módulo de administração de usuários é muito simples de usar, principalmente para o usuário final de suas aplicações. Neste módulo, o administrador pode:

- criar usuários;
- criar perfis de usuário;
- associar usuários a um perfil;
- conceder permissões de acesso aos usuários (individualmente) e aos perfis, visualmente;
- atualizar as permissões de acesso de todos os usuários que pertençam a um perfil;
- mudar dados e senha de usuário;
- forçar a mudança de senha de usuários de um determinado perfil a cada intervalo de dias especificado;
- colocar um usuário ou um perfil de usuário em modo de auditoria e verificar a utilização dos programas, entre outras ações.

O componente **TUsersCS** pode ser usado com qualquer SGBD do mercado que suporte o padrão ANSI-SQL e que esteja acessível pelo Delphi usando **BDE** ou **ODBC**.

Outra característica importante do **TUsersCS** é que as tabelas de controle de usuários podem ser compartilhadas por diversas aplicações ao mesmo tempo, não sendo necessário recadastrar os usuários em todas as aplicações.

**TUsersCS** é implementado em Object Pascal puro, inclusive o módulo de administração de usuários, que é implementado juntamente com o componente. Não são necessárias bibliotecas externas nem programas adicionais a serem enviados juntamente com seu programa em Delphi para que o componente funcione corretamente. Você deve apenas enviar as tabelas de usuários juntamente com seu sistema.

## INSTALANDO O TUSERSCS

---

Para instalar o **TUsersCS**:

- abra o arquivo ToolsAndComps\UsersCS\Delphi3Trial\uCSD3.dpk, através do menu File|Open do Delphi e clique no botão Install.
- No IDE do Delphi, abra Tools|Enviroment Options|Library|Library Path, e adicione c:\ToolsAndComps\UsersCS\Delphi3Trial, clicando no botão **OK**.

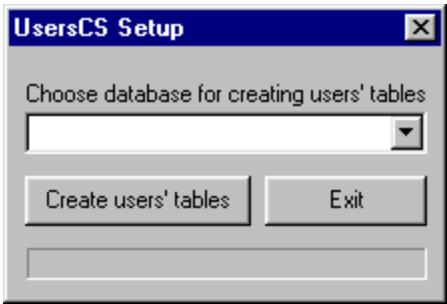
OBS: Os procedimentos são os mesmos para Delphi 4.

## COMO FUNCIONA O COMPONENTE TUSERSCS

---

### 1. Criando as tabelas de usuários

O primeiro passo para usar o **TUsersCS** é criar as tabelas que irão armazenar as informações sobre os usuários de seu sistema. O **TUsersCS** vem acompanhado de um utilitário que auxilia na criação dessas tabelas, o **UCSSetup.EXE**.



Para criar as tabelas de usuários, você deve informar o alias(banco de dados) no qual você deseja armazenar essas tabelas (você deve criar previamente um alias no BDE ou um DSN no administrador ODBC de seu computador). Depois de informar o alias(banco de dados), basta clicar no botão **"Create users's tables"** para que o utilitário crie as tabelas de acordo com as informações especificadas. As tabelas a serem criadas são:

1. USERSCS
2. USERINFO
3. PROFILES
4. PROGS
5. UACCESS
6. PACCESS
7. HISTORY
8. AUDIT

**Dica:** Você pode criar as tabelas de usuários no mesmo banco de dados que a sua aplicação vai usar ou você também pode criar um banco de dados específico para o controle de usuários.

**Nota:** Pelo fato de ser o desenvolvedor quem escolhe o SGBD aonde serão criadas as tabelas de usuários, a segurança dessas tabelas deve ser a mesma para as demais tabelas de seu sistema.

Quando o utilitário **UCSSetup.EXE** cria as tabelas de usuários, automaticamente insere um registro com o nome de usuário **'master'**, com senha também **'master'** e total acesso ao sistema. Esse é o usuário que irá gerenciar o sistema de segurança de seus programas, não sendo possível removê-lo ou mudar seus acessos dentro do sistema, sendo apenas possível mudar sua senha (o que é recomendado).

**Nota:** Para uma maior segurança do sistema, as senhas de todos os usuários cadastrados são criptografadas.

## 2. Usando o TUsersCS dentro do IDE do Delphi

Para usar o TUsersCS dentro de seus programas, na paleta de componentes, selecione a guia **ToolsAndComps**, depois adicione o componente ao formulário principal do seu programa.

Para que o componente funcione corretamente, você deve fornecer inicialmente o valor das seguintes propriedades no Object Inspector:

*AppKey: String[20]*

Chave para a aplicação. Essa propriedade é obrigatória e permite que as tabelas de usuários sejam compartilhadas por várias aplicações. Essa propriedade vai funcionar como a chave da aplicação atual e deve ser única entre as aplicações que vão compartilhar a mesma base de dados de usuários. Note que o componente não pode verificar a unicidade desta chave. O próprio desenvolvedor deve tomar as devidas precauções quanto a isso.

*DatabaseName: String*

Indica o alias ao qual o componente **TUsersCS** vai se conectar para abrir as tabelas de controle de usuários.

**IMPORTANTE:** Através do alias fornecido como parâmetro, o componente irá tentar abrir a base de dados de usuários(Criada através do utilitário **UCSSetup**) usando as propriedades do alias, incluindo o usuário e a senha usada pela sua aplicação para abrir sua base de dados. Note que o componente deve estar "autorizado" a acessar a base de dados de usuários. Essa autorização pode ser feita da seguinte forma:

1. Solicite ao seu DBA para que crie um usuário em seu banco de dados e que ele, e somente ele, tenha acesso as tabelas de controle de usuários.
2. Adicione um componente TDatabase ao seu formulário principal e conecte-o ao banco de dados aonde estão as tabelas de controle de usuários.
3. Na propriedade Params do TDatabase criado, adicione as seguintes linhas:  
USER NAME="Nome Do Usuario Criado Pelo DBA"  
PASSWORD="Password Do Usuario Criado"
4. Mude o valor da propriedade LoginPrompt para FALSE
5. Conecte o componente TUsersCS de seu formulário principal ao componente TDatabase criado

Veja os exemplos que acompanham o componente

*Menu: TMenu*

Indica o componente de menu que o **TUsersCS** irá controlar. Se seu formulário já tem um componente TMenu, o **TUsersCS** se conecta automaticamente a ele.

**Importante:** Note que o alias (propriedade DatabaseName) deve ser o mesmo que você forneceu quando criou as tabelas de usuários usando o utilitário **UCSSetup.EXE**. Sem os valores corretos, o componente não poderá funcionar corretamente.

## 3. Outras Propriedades do TUsersCS

*AutoLogin: Boolean*

Especifica se o componente vai chamar a função de Login no sistema automaticamente. Se AutoLogin é **True**, o componente chama a função de Login quando a aplicação é inicializada. Caso o usuário não consiga logar na aplicação após 3 tentativas, o componente termina a aplicação. Se AutoLogin é **False** a função de Login deve ser chamada manualmente.

*ActualUser: TUser*

Essa propriedade contém dados sobre o usuário atualmente logado no sistema.

## 4. Propriedades da Classe TUser

*UserName: String*

Nome do usuário logado

*RealName: String*

Nome real do usuário logado

*UserId: Integer*

Número de identificação do usuário logado

*Password: String*

Senha do usuário logado

**Nota:** O componente mantém um histórico de senhas de todos os usuários, de forma que este seja forçado a usar senhas diferentes, cada vez que muda sua senha. Para que um usuário possa reutilizar uma senha, o administrador deve remover a senha do histórico de senhas do usuário, no módulo de administração.

*LastPwdChange: TDateTime*

Data da última mudança de senha do usuário

*ExpirationDate: TDateTime*

Data da expiração do usuário dentro do sistema. A partir dessa data o usuário não poderá mais logar no sistema. Para voltar a usar o sistema, o administrador de segurança deve torná-lo ativo e mudar sua data de expiração. Essa propriedade só faz sentido se a propriedade *UserExpire* é TRUE.

**IMPORTANTE:** Observe que este valor está vinculado a aplicação, ou seja, um usuário tem a data de expiração diferente para cada aplicação.

*UserExpire: Boolean*

Indica se a validade do usuário expira dentro do sistema.

**IMPORTANTE:** Observe que este valor está vinculado a aplicação, ou seja, um usuário pode expirar ou não, de acordo com a aplicação.

*UserActive: Boolean*

Indica se o usuário está ativo, ou seja, se a validade do usuário dentro do sistema não expirou.

**IMPORTANTE:** Observe que este valor está vinculado a aplicação, ou seja, um usuário pode estar ativo ou não, de acordo com aplicação.

*Profile: String*

Retorna o nome do perfil ao qual o usuário pertence.

*ProfileId: Integer*

Retorna o identificador do perfil ao qual o usuário pertence.

*AuditoryMode: Boolean*

Indica se o usuário está em modo de auditoria ou não.

**Nota:** Todas as Propriedades da classe **TUser** são somente leitura.

## 5. Métodos do TUsersCS

*Login: boolean*

Esse método deve ser usado para chamar a tela de login do sistema. Se após 3 tentativas, o usuário não conseguir logar no sistema, essa função booleana retorna **False**. Para um login bem sucedido, essa função irá retornar **True**. A chamada a essa função se faz necessária quando o CSriedade *AutoLogin* é **False**. Neste caso, o desenvolvedor deve escrever o código necessário para que a aplicação não possa ser usada por um usuário não autorizado. Veja os exemplos.

*ChangeUserPassword*

Esse método permite ao usuário logado efetuar a mudança de sua senha. Deve ser chamado manualmente pelo Desenvolvedor.

*UsersAdm*

Chama o módulo de administração de usuários do sistema. Deve ser chamado manualmente pelo Desenvolvedor.

## SUPORTE A OUTROS IDIOMAS

O **TUsersCS** foi projetado para que o desenvolvedor possa modificar facilmente o idioma usado pelo componente. A versão atual conta com tradução de todas as suas strings de interface com o usuário em português, espanhol e inglês. Para cada um dos idiomas citados, o pacote do **TUsersCS** vem com o arquivo de recursos já compilado (UsersCSRes.RES), o arquivo fonte da STRING TABLE (UsersCSRes.RC) e o arquivo com as constantes (UsersCSRes.PAS). Para compilar seu programa para um dos idiomas citados, basta incluir em seu projeto, o arquivo UsersCSRes.Pas que está no diretório do idioma desejado.

Caso você queira usar o componente em alemão, por exemplo, siga os seguintes passos:

1. Dentro do diretório ToolsAndComps\UsersCS\Resources, crie um diretório para o idioma desejado.

2. Copie todos os arquivos do diretório de algum dos idiomas para o novo diretório.
3. Traduza o arquivo UsersCSRes.RC para o idioma desejado.
4. Compile o novo arquivo UsersCSRes.RC utilizando o utilitário BRCC32.EXE, que está no diretório BIN da sua instalação do Delphi. Consulte sua documentação do Delphi a respeito do uso do BRCC32.EXE.
5. Inclua em seu Projeto, o arquivo UsersCSRes.Pas que está no diretório do novo idioma. O novo arquivo UsersCSRes.RES gerado também deve estar nesse diretório.
6. Recompile seu Projeto.

## **REGISTRANDO O TUSERSCS**

---

Visite nossa home page para obter informações atualizadas sobre como registrar o componente TUsersCS v1.0