

# Roteiro de Trabalho

Neste capítulo, se encontram as instruções referentes ao processo realizado durante o início de um caso até o seu encerramento e todas as etapas contidas neste intervalo.

- [Qual caso escolher?](#)
- [Localizando e configurando a base de dados](#)
- [Rotina de Desenvolvimento](#)
- [Liberando para testes](#)
- [Subindo um caso para o branch de versões](#)

# Qual caso escolher?

Antes de iniciar este tópico, certifique-se de já ter [configurado seu ambiente de trabalho](#).

## Passos Iniciais

### Trello

Para iniciar o desenvolvimento de um caso, você deve "escolher" um caso. Você não vai literalmente escolher qual caso trabalhar, mas sim, seguir uma ordem pré-definida pela qualidade (ou outro departamento) através do Trello.

Existem duas linhas principais de trabalho:

- Erros e Correções;
- Implementações e Solicitações;

Caso você não saiba em qual está alocado, verifique com o líder ou gerente da sua equipe.

Após definida a linha de trabalho, você selecionará o primeiro caso da coluna e clique em ingressar, conforme exemplificado a seguir:

 **0044381: Planilha contábil não apresenta o evento e706: Retirada Pró-Labore (V.Liq.).** ×

na lista [Aguardando Implementação Erros / Legislação](#)

ETIQUETAS

Erro / Erro Interno / Não Previsto

SSFolha

+

Descrição

Editar

<https://mantis.supersoft.com.br/desenvol/view.php?id=44381>

Atividade

Mostrar Detalhes



SUGERIDAS

Ingressar

ADICIONAR AO CARTÃO

Membros

Etiquetas

Checklist

Data Entrega


Anexo

Capa

Após ingressar no cartão, arraste-o para a coluna "Em desenvolvimento":

Em Desenvolvimento

0044381: Planilha contábil não apresenta o evento e706: Retirada Pró-Labore (V.Liq.).



## Mantis

Após a seleção do caso no Trello, e o deslocamento dele para a coluna "Em desenvolvimento", acesse o Mantis e no canto superior direito, busque seu caso pelo número indicado no cartão do Trello, ou caso tenha o link no cartão é só clicar.

Após localizar seu novo caso, será necessário atribuir a você mesmo, para que não ocorra o problema de outro desenvolvedor pegá-lo. No final do tópico "Ver Detalhes da Tarefa" você verá a opção de "Atribuir a:", se não estiver atribuído a ninguém, o próprio Mantis irá sugerir para atribuir para você (como mostrado a seguir), clique no botão para atribuir e pronto, o caso já estará sendo implementado por você.

Atribuir a:

[Eu mesmo]

▼

Concluída essa etapa, avançaremos para a [criação do seu Branch](#), para enfim começarmos a implementação do caso selecionado.

# Localizando e configurando a base de dados

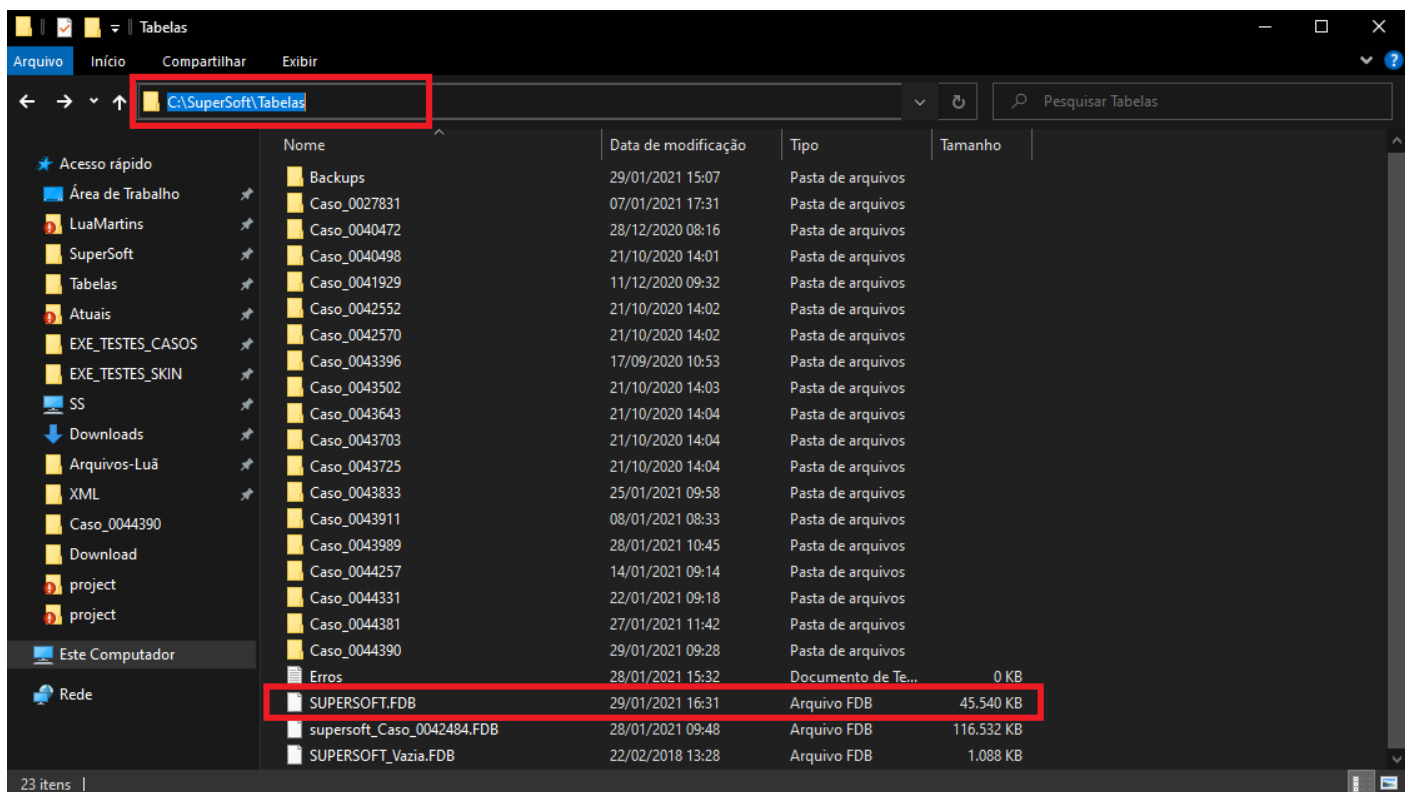
## Configurando bases de dados FireBird

A extensão do arquivo da base de dados precisar ser .FDB para a IDE conseguir compilar sem erros.

Na maioria das vezes quando iniciamos um novo caso, será necessário configurar a base de dados, que nos possibilita reproduzir o problema relatado e que geralmente está anexada no Mantis ou no Transfer (\\SS\\Transfer\\SUPERSOFT\_CLIENTES).

O processo é muito simples, basta fazer o Download ou uma cópia caso esteja no transfer para dentro da pasta SuperSoft localizada no seu diretório, possui uma pasta chamada Tabelas, é dentro dela que nossa IDE está configurada para buscar o arquivo com extensão .FDB.

Dentro da pasta Tabelas, crie uma pasta com o nome do seu caso e cole a base dentro dela. Depois extraia a base para obter o arquivo .FDB, conforme demonstrado a seguir:

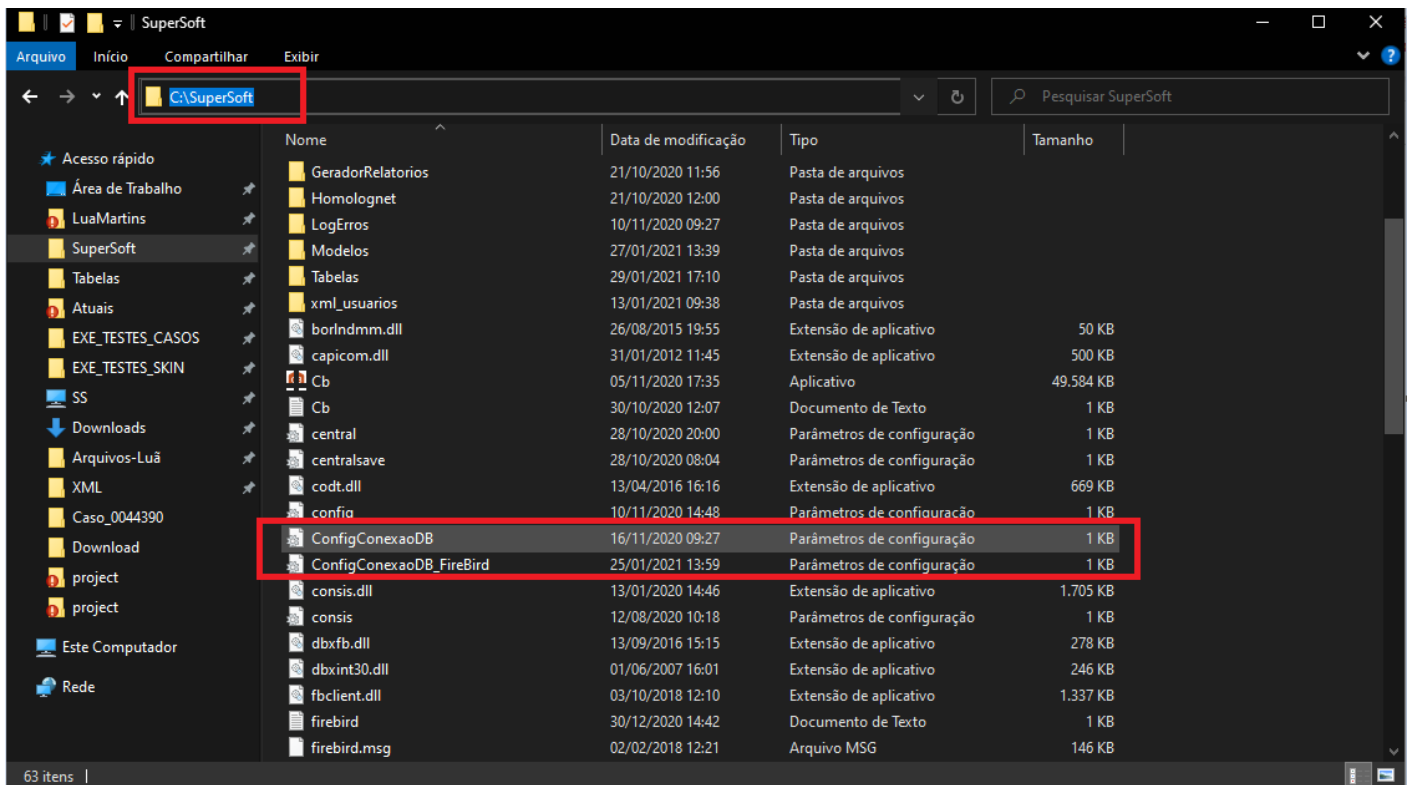


Pronto, sua base já está configurada para ser utilizada, dentro do Delphi compile seu código que a base já estará sendo utilizada no processo de compilação.

## Configurando bases de dados SQL

Em alguns casos será necessário reproduzir o erro com bases SQL, ou seja, você precisará ter instalado na sua máquina o software "sqlIncli" (de acordo com a arquitetura do Windows x32 ou x64), localizado em \\SS\transfer\FelipeB\App.

Você também precisará do arquivo da configuração do Banco de Dados, chamado "ConfigConexaoDB", que geralmente estará localizada anexada no Mantis ou no Transfer com seu respectivo caminho relatado no Mantis. Basta copiar este arquivo e colar dentro da pasta SuperSoft, mas antes renomeie o arquivo de configuração já existente dentro da pasta para qualquer outro nome que você possa se lembrar depois para não perder as configurações usadas para uma base FireBird, conforme exemplo a seguir:

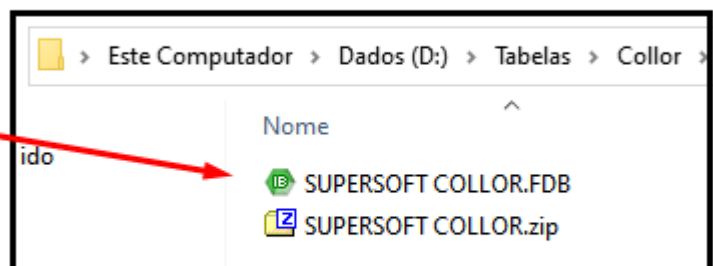
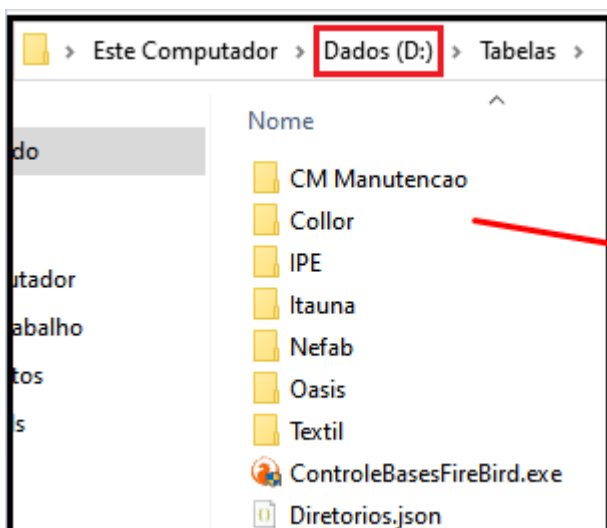


Pronto, depois basta compilar seu caso dentro do Delphi que a base estará pronta para ser utilizada.

## Outra forma:

### Requisito: Utilizar o Controle de Bases Firebird

O fato de usar o Controlador de Bases gera uma flexibilidade de onde ficará as bases de dados e até o nome. Assim poderá guardar por nome de cliente por exemplo, e também pode deixar na segunda partição/segundo disco caso tenha. Exemplo:







# Rotina de Desenvolvimento

## Pré-Requisitos

Antes de começar a implementação referente ao caso propriamente, é necessário que as seguintes etapas já estejam concluídas por parte do desenvolvedor:

1. Que já tenha ingressado no cartão do caso em questão, no [Trello](#);
  2. Que já tenha lido e interpretado o caso no [Mantis](#), além de estar atribuído ao caso;
  3. Que já tenha criado a [respectiva pasta](#) no seu diretório SVN;
  4. Que já tenha criado a [respectiva pasta](#) no seu diretório local;
  5. Que já tenha [localizado e configurado](#) a base de dados do caso.
- 

## Processos comuns no desenvolvimento

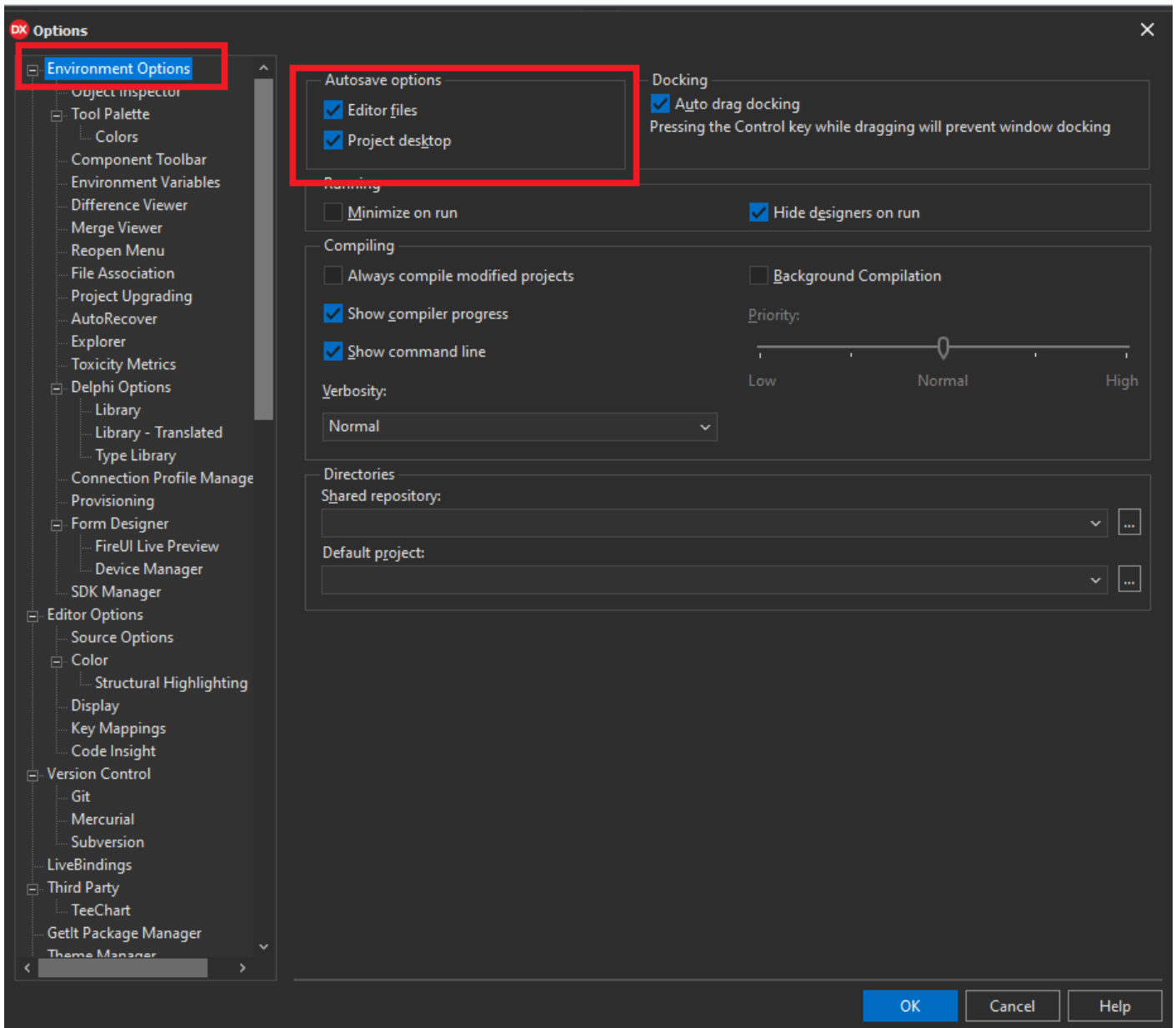
### Dicas de IDE

Quando estamos no desenvolvimento do caso, algumas dicas são fundamentais para facilitar nossa vida, a seguir será demonstrado algumas delas:

#### Salvando o estado do projeto

Após fecharmos o a IDE, as *units*, *breaking points* e os *bookmarks* adicionados, serão todos perdidos. Entretanto existe uma opção que salva todas esses estados no qual você deixou ao longo da implementação.

Dentro da IDE o menu *Tools -> Options*, aba *Environment Options*, marcar as duas opção do retângulo *Autosave Options*, conforme a imagem a seguir:



## Debug

Para reproduzir um erro ou testar alguma implementação, você precisa debugar para verificar se tudo ocorreu como esperava. Neste processo temos 3 maneiras.

### Debugando com a tecla "F9"

Você estará pulando de um Breaking point a outro, de forma que os métodos entre eles serão lidos pela IDE, mas não mostrados como a tecla "F8".

```
1530 begin
1531     Result := '';
1532     QVALANULBENEFFA := CriarFDQuery(nil);
1533     try
1534         CarregaValAnulBeneficiarioPA;
1535         if not TemValoresDePensao then
1536             Exit;
1537         Result := 'RTPA|';
1538         for I := 1 to 13 do
1539             begin
1540                 NomeCampo := 'PENSÃO'+PreencheComZeros(2, I.ToString);
1541                 ConcatenaCampo(Result, FormataNumMonLayout(QVALANULBENEFFA.FieldName(NomeCampo).AsFloat));
1542             end;
1543         finally
1544             FreeAndNil(QVALANULBENEFFA);
1545         end;
1546     end;
1547 end;
```

## Debugando com a tecla "F8"

A IDE irá ler linha a linha do código, tornando mais fácil analisar o que está sendo feito dentro de um método ou até mesmo alguma verificação.

```
180 end;
181 begin
182     QPesq := CriarFDQuery(nil);
183     try
184         CarregaBeneficiariosComValPensaoAlimenticia();
185         GeraValoresPASocAutNaValAnualBenefPA;
186     finally
187         FreeAndNil(QPesq);
188     end;
189 end;
```

## Debugando com a tecla "F7"

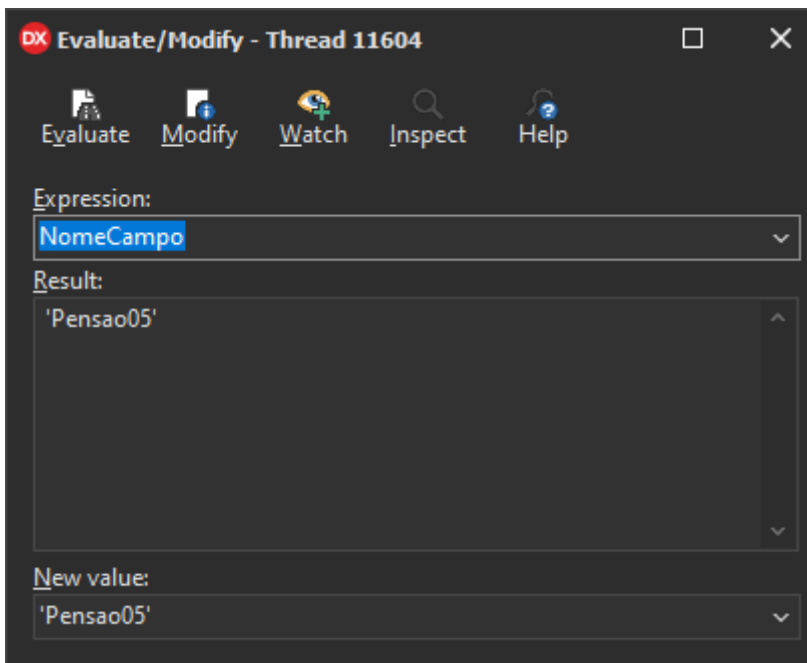
Ao se deparar com um método, você pode utilizar a tecla "F7" para adentrar nele e ver tudo o que está sendo feito.

Por exemplo, na rotina de exportação do arquivo da Sefip, chama o método GeraValoresPASocAutNaValAnualBenefPA(), apertando a tecla "F7" na linha que chama este método, você será redirecionado para dentro dele sem a necessidade de ter um breaking. Como mostrado a seguir,

```
151 procedure GeraValoresPASocAutNaValAnualBenefPA();
152 var
153     NomeCampo: String;
154 begin
155     NomeCampo := 'Pensao' + PreencheComZeros(2, LMesPagamento);
```

## Evaluate (Ctrl + F7)

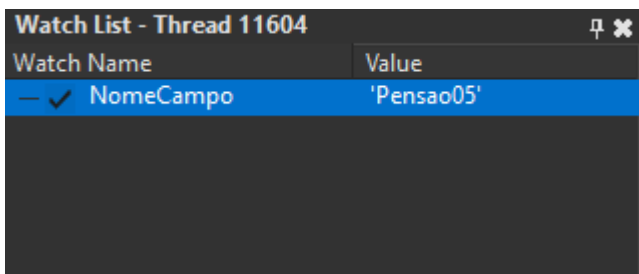
Dento da IDE a tela de Evalute é de grande importância na hora que estamos Debugando, pois com ela podemos obter diversas informações a partir dela.



Além de conseguir ver o conteúdo de uma variável passando o mouse por cima, pode ser visto com o Evaluate.

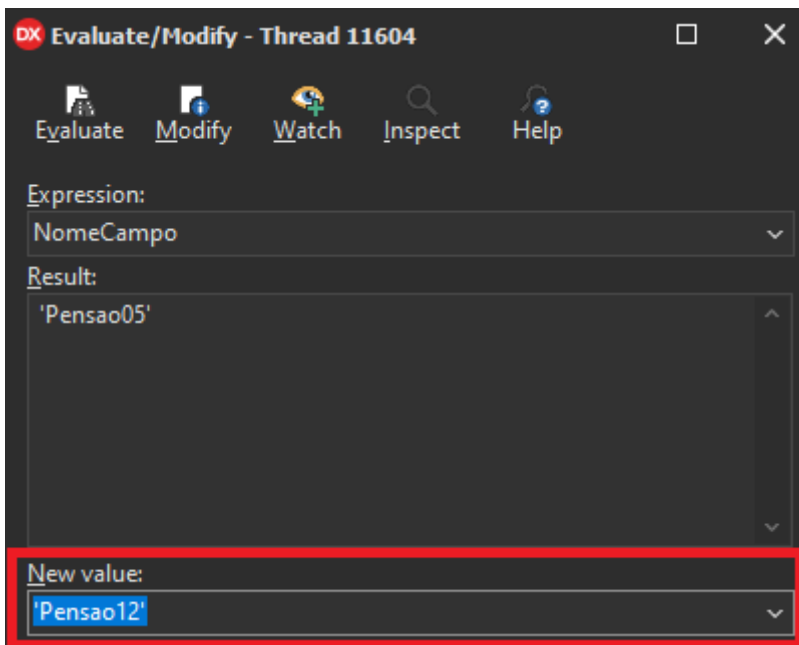
## Watch (CTRL + F5)

A ferramenta Watch disponível dentro do Evaluate, nos possibilita ver o que está acontecendo com uma determinada variável ou campo de uma Query em tempo real, a cada passo no processo de Debug, se o valor for alterado, imediatamente você consegue ver essa mudança.



## New Value

A ferramenta New Value, localizada na parte inferior do Evaluate é utilizada quando precisamos alterar um determinado valor no processo de execução, por exemplo, você deseja verificar o que está acontecendo dentro de uma rotina, mas por não entrar nesta rotina devido a alguma verificação não obedecida, é possível alterar o valor nesta verificação "forçando" com que entre na rotina sem precisar alterar o código, compilar e depois debugar até o trecho desejado.



Veja que a variável "NomeCampo" está com o valor de "Pensao05", adicionando no New value o valor desejado e apertando enter, a variável irá receber o conteúdo escrito no campo.

As próximas duas são muito importantes e muito úteis.

## MostraQuery

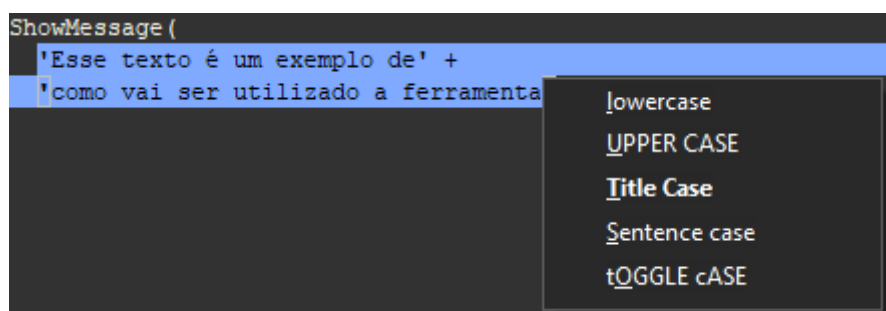
Constantemente trabalhamos com Querys, que são responsáveis por buscar informações do banco de dados. Podemos verificar o conteúdo dela com o comando MostraQuery("NomeDaQuery"), que irá mostrar todo os campos da query, mas deve ficar atendo pois em muitos casos ela estará em modo edição, então se alterar no MostraQuery o registro selecionado, estará mudando também na Query.

## MostraParams

O comando MostraParams é bem parecido com o MostraQuery, porém ele irá mostrar a clausula responsável por buscar as informações do banco de dados.

## Indentação de Texto (Alt + Shift + C)

A ferramenta serve para facilitar a **formatação** de textos .



O texto selecionado será o único a ser alterado. Sendo que pode-se escolher o tipo de formatação desejada. A formatação mais comum para **sentenças (strings)** será a "**Sentence case**".

Cuidado ao utilizar a "Title Case" em métodos, nesta formatação somente a primeira letra ficará em maiúsculo e as demais em minúsculo. Caso utilize, verificar se a formatação do método está correto.

# Dicas de Trabalho

## Commits periódicos das alterações

Após finalizar a implementação de algum trecho e que ainda será necessário alterar outras partes do código, é recomendado fazer o commit dessas mudanças, para assegurar que as alterações ficarão salvas caso aconteça algum problema em sua máquina ao decorrer da implementação.

## Merges periódicos com o Trunk

Ao decorrer de um caso, as implementações feitas podem não ter obtido os resultados esperados, que consequentemente será necessário novas implementações. Quando estiver tratando um caso grande, ou que já passou mais de uma semana desde a ultima liberação da versão, é importante o merge com o Trunk, fazendo com que o Branch do seu caso, receba todas as novas alterações que estejam no Trunk.

## Solicitação de ajuda

Apesar de nos depararmos com inúmeras dificuldades ao decorrer de uma determinada implementação, caso não consiga mais avançar no tratamento do seu caso, peça ajuda à algum desenvolvedor para lhe orientar e mostrar algum possível caminho para a solução do seu problema.

# Liberando para testes

Ao finalizar a implementação do caso e possíveis correções referentes à revisão de código colaborativa, o caso avança para os testes. Onde o tratamento será validado pela equipe da qualidade.

## Equipe ERP:

Ao liberar para testes montamos um informativo com a estrutura parecida com:

1. Marcar a(s) pessoa(s) da Qualidade ou que irão testar o caso (como Projetos em alguns casos).
2. Motivo do não funcionamento correto (caso haja).
3. Resumo das alterações feitas.
4. Informar Units Alteradas/Incluídas/Deletadas

Exemplos:

“

@AlineC, segue alteração realizada para testes iniciais.

Alteração

- Alteração realizada de forma que o sistema calcule pis e cofins ao importar xml
- Com a customização as alíquotas utilizadas devem ser buscadas dos parâmetros da empresa.(Dados Principais).
- Apenas os cálculos referente a pis e cofins devem ser reprocessados.
- Caso necessário o recálculo "automático", utilizar o "Recalcular valores da Notificação" nos itens.
- Verificar se os valores são calculados conforme necessidade do cliente ao importar o xml.
- Realizar testes sem a customização.

Unit Alterada

- FuncsCC.pas

\*Criada CustomCCNefab

Qualquer dúvida estou a disposição.  
Obrigado.

---

“ @AlineC Caso disponível para testes.

Alterações:

- Ao Gravar a nota, foi melhorado a rotina que limpava sujeira das tabelas filhas da NFModel1, agora também é feito a mesma rotina para tabelas filhas da ItensNF.
- Como mesmo aparecendo a mensagem "Já existe movimentação no SSEstoq..." ele gravava (pois a gravação da mesma na NFModel1 é feita antes destas validações) a nota gerando sujeira, agora apresenta mensagem que não foi gerado movimentação no SSEstoq mas continua com a rotina correta da gravação da Nota, assim não gera sujeira nessa situação.
- Adicionado barra de progresso ao fazer nota simultânea.

Units alteradas:

/SSVendas/NFMod12.pas

/SSvendas/FuncsNF.pas

Att.:

Vinicius Machado

---

“ @AlineC, segue alteração realizada para testes iniciais.

Alteração

1 - A regra para atualização da classificação de clientes foi alterada seguindo os seguintes critérios.

1.1 - Ao iniciar o sistema, o procedimento deve ser iniciado, rodando automaticamente apenas uma vez ao dia.



2 - Caso entre pelo menu Utilitários > Customização > Movimentação > Classificação de clientes, o mesmo deve sempre ser acionado.

Unit Alterada

- DllSource.pas

\*Alteração realizada diretamente na customização.

Qualquer dúvida estou a disposição.  
Obrigado.

Colocamos esse informativo como anotação no Mantis.

Alteramos o [Status](#) do caso para Aguardando Testes.

---




## Equipe FISCO:

Para a equipe FISCO, o processo é praticamente o mesmo da equipe ERP, a única diferença é que o desenvolvedor é responsável pela geração dos executáveis que serão testados. Caso você não saiba como fazer isso, acesse: [Geração de Executáveis](#)

# Subindo um caso para o branch de versões

Somente fazer esse procedimento após o pedido vindo da Qualidade.

Pelo lembrete recebido, terá o branch da versão que subirá seu caso, como exemplo abaixo é o Branch 043

 <b>Qua_Jefferson</b> 01/02/21 09:42 desenvolvedor -0296001	Realizado testes do caso e o sistema está funcionando como esperado. @GabrielF, por favor subir o caso para o Branch 043.  Obrigado  SUPERSOFT_RODZA.rar (91,222,238 bytes)  ROTEIRO_TESTES_caso_0044153-3.ods (21,983 bytes)
--	--

Verifique na sua pasta local de Branchs se há o Branch em questão já baixado, caso não tenha, fazer o [Checkout](#) da atual conforma imagem de exemplo, atente-se ao departamento se é **ERP** ou **Fisco**

URL: <https://svn.supersoft.com.br/svn/desenvolvimento/branches/Versoes/ERF> Revision: HEAD

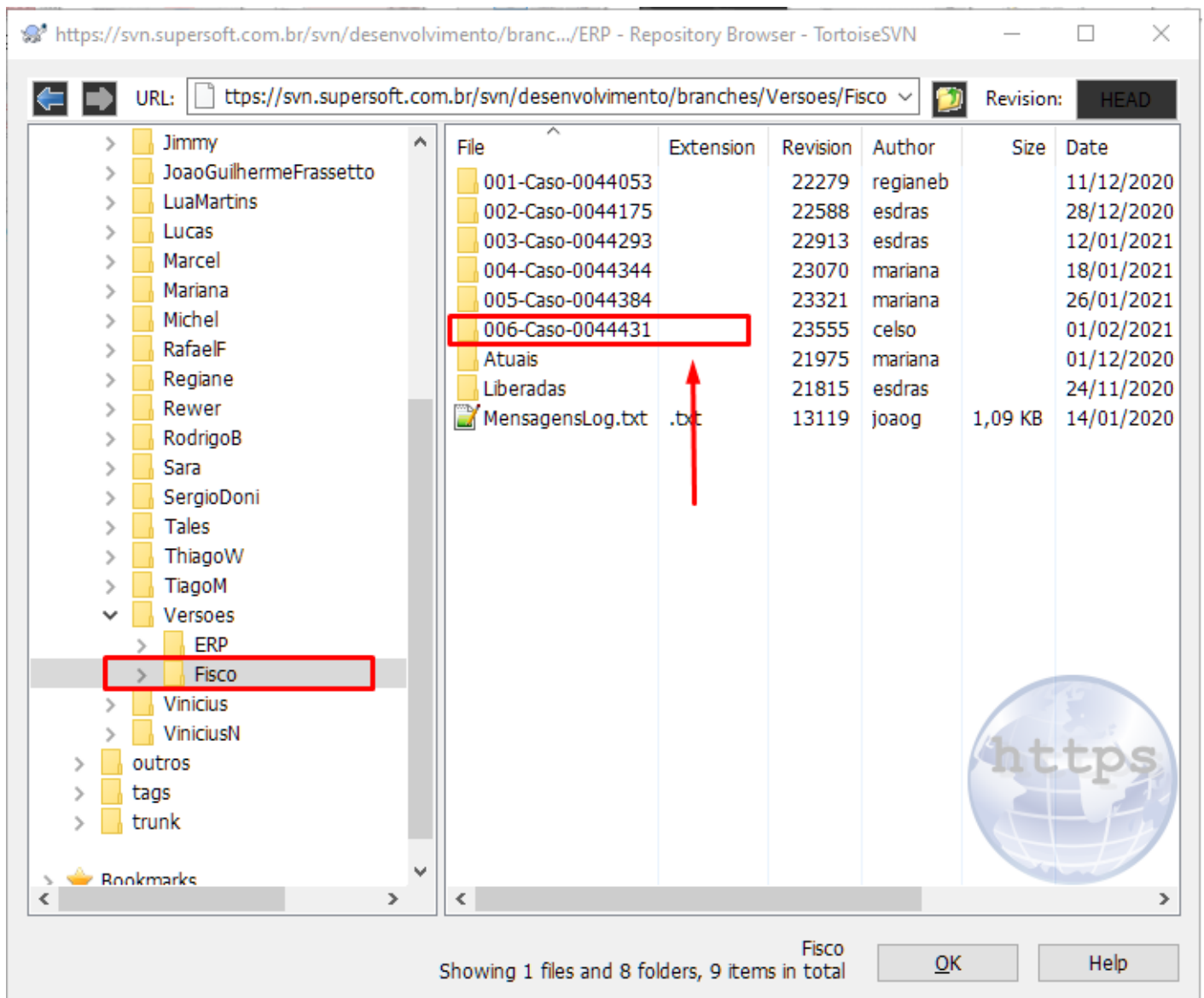
File	Extension	Revision	Author	Size	Date
020-Caso-0042932		18611	rewerp		05/08/2
021-Caso-0042970		18610	rewerp		05/08/2
022-Caso-0043005		18609	rewerp		05/08/2
023-Caso-0043089		18608	rewerp		05/08/2
024-Caso-0043133		18607	rewerp		05/08/2
025-Caso-0043278		18606	rewerp		05/08/2
026-Caso-0043317		18391	rewerp		30/07/2
027-Caso-0043372		18551	compiladorerp		03/08/2
028-Caso-0043393		18812	compiladorerp		11/08/2
029-Caso-0043454		19240	vinicius		26/08/2
030-Caso-0043511		19556	compiladorerp		08/09/2
031-Caso-0043598		19872	compiladorerp		17/09/2
032-Caso-0043659		20355	compiladorerp		02/10/2
033-Caso-0043741		20541	compiladorerp		09/10/2
034-Caso-0043791		20898	mateus		23/10/2
035-Caso-0043844		21168	sergiotd		03/11/2
036-Caso-0043889		21568	vinicius		16/11/2
037-Caso-0043970		21721	mateus		19/11/2
038-Caso-0044027		21959	rewerp		01/12/2
039-Caso-0044090		22124	rewerp		07/12/2
040-Caso-0044129		22412	mateus		17/12/2
041-Caso-0044219		22822	mateus		08/01/2
042-Caso-0044323		23336	mateus		27/01/2
043-Caso-0044433		23558	gabrielF		01/02/2

Showing 0 files and 43 folders, 43 items in total

ERP

OK Help

ERP



### Fisco

Após o Checkout, verificar se tanto o branch da versão quanto a pasta do seu caso estão atualizadas com o Trunk, caso não esteja poderá fazer [Merge](#) diminuindo as diferenças entre as pastas.

Você terá algumas formas de passar o seu tratamento para o branch da versão, as duas apresentadas a seguir usarão a ferramenta Beyond Compare, caso não saiba utilizar poderá ver [Clicando Aqui](#).

1. Comparar as duas pastas inteiras uma com a outra, e passar suas alterações para branch da versão localizando as Units alteradas.
2. Comparar Unit por Unit que saiba que alterou, e passar suas alterações para branch da versão.

Cuidado para não retirar o tratamento de outro Dev ao passar o seu para o Branch da versão.

Após passar suas alterações para a pasta referente ao Branch da versão, irá fazer um [Commit](#) seguindo as mensagens de [Versionamento](#).

Ir no caso no Mantis e enviar um **lembrete** para Qualidade avisando que o caso está disponível para reteste no branch da versão indicado por eles,

Alterar o status para **Aguardando revisão de testes**

The screenshot shows the Mantis issue form. At the top, there are buttons for 'Atualizar', 'Atribuir a:', and 'Alterar Status:'. The 'Alterar Status:' dropdown menu is open, showing a list of status options. The option 'aguardando revisão de teste' is highlighted with a red rectangle. Below the status dropdown, there are sections for 'Relações' and 'Usuários monitorando esta tarefa'. The 'Relações' section has a dropdown for 'Esta tarefa' and an 'Adicionar' button. The 'Usuários monitorando esta tarefa' section has a list of users: Des\_Mariana, Pro\_Wesley, and Qua\_Jefferson, and a text input for 'Nome de usuário'.

No Trello estará na coluna **Testado (por caso)**, mova para a coluna **Vinculados/Em testes (Branch versão)**.

