Em andamento

Livro para capítulos e páginas que ainda não foram finalizadas

To-Do

- Adicionar nessa lista, tópicos que precisam ser adicionados ou tópicos já existentes que precisam de revisões ou definições à respeito do processo.
- Conforme os tópicos forem concluídos, "tachar" (~~Texto exemplo~~) o texto e adicionar descrição de conclusão ("Revisão feita", "processo definido", "tópico atualizado", etc).

Para melhor organização e identificação da página, definiremos cores para os editores:
Esdras - Estarei digitando em azul
Mateus - Estarei digitando em verde
Luã - Estarei digitando em Laranja
V1 - Estarei digitando em Rosa

A fazer:

Fazendo:

- Revisar "Atualização dos arquivos AltVer.rtf no SVN e módulos", verificar necessidade de abrir ALTVER1 antes do Altver propriamente. - Revisar "Commit das alterações dos arquivos Config.ini e Altver.rtf", abstrair processo do commit e especificar apenas as informações referentes à esse processo. Há a necessidade de criar mensagens de versionamento padrões para branch da versão? (Semelhante à mensagens de versionamento para casos: Versionamento). - Adicionar SVN - Avançado (Concluir documentacao merge, e analisar "como resolver conflitos, para busca de informação, principalmente Merge: Documentação TortoiseSVN"). - Adicionar "perguntas frequentes": "Quando adicionar aspas duplas, quotedstr, #39 ou parâmetros?" "Como utilizar "break", "exit" ou "abort"? - Discutir a respeito do arquivo Alt|Ver para incluir tratamentos importantes lançados. - Padronizar os documentos referentes às etapas e sub processos relacionados aos processos de rotina de testes e gerenciamento do branch de versão, conforme o padrão SuperSoft de documentação. Obs: utilizados os comandos [padronizar] e [linkar] para facilitar na pesquisa dos documentos que precisarão ser padronizados e/ou serem criados e posteriormente linkados com os processos que os utilizarão. - Montar fluxograma para os processos de rotina de testes e gerenciamento do branch da versão.	* Criar "Manual de documentação" (Conforme mais pessoas adicionam na wiki, precisamos definir padrões para os nossos documentos terem uma identidade)

Feito:

~~- Revisar "Realizando o update, merge e commit para liberação das versões", visto que teremos um capítulo para o uso do SVN, podemos abstrair dessa página o processo propriamente e apenas especificar as informações referentes ao branch da versão.~~
(Tópico atualizado)

~~- Revisar "Geração dos arquivos AltVer.rtf", processo de armazenamento dos AltVer. Será definido um diretório apenas para os altver? Será utilizado o mesmo diretorio dos exes? Definir nomenclatura dos arquivos armazenados.~~
~~**Atualização**: Conversei com a Mari, ela vai verificar essa situação certinho e logo teremos uma definição.~~

(Tópico atualizado)

~~- Elaborar roteiro de dinâmica dos casos~~

~~- Documentar "Abertura do Branch", "Manutenção do Branch" e criar índices para o livro Gerenciamento do Branch de Versões~~ (Tópico atualizado)

~~- Abertura de branch: Criar caso no mantis. Copy to do trunk.~~
~~- Manutenção do branch: Merge com o trunk diariamente (definir frequência). Relacionar casos/Versoes dos módulos. Atualizar config ini (faz parte da manutenção ou da abertura?)~~
(Tópico atualizado)

~~- Adicionar na parte de configuração do ambiente, casos que serão usada base SQL é necessário a instalação do software "sqlncli" (de acordo com a arquitetura do Windows x32 ou x64), localizado em \\SS\transfer\FelipeB\App~~

~~- Adicionar padrão para TMenuItem na lista de componentes delphi. (Verificar necessidade de outros componentes tb).~~
(Item adicionado, não identifiquei outros componentes para inserir na lista no momento.)

~~Adicionar guia para utilização de bases anexadas nos casos, na utilização individual das implementações.~~

~~- Adicionar processos referentes à Library Path padrão e processo para recuperar o estado do delphi ao reabrir o programa.~~

[BETA] Processo: Análise de complexidade

Neste capítulo, são feitas considerações à respeito da análise para definição da complexidade de um caso.

To-do:
Definir processo.
Concluir/Revisar considerações gerais.
Concluir revisões conjuntas.

Considerações Gerais

Tópicos considerados

Durante o processo de análise para definir a complexidade de um caso, alguns pontos devem ser levados em consideração:

Tipo de alteração

1. **Alteração apenas "adicional"**
   A alteração feita no caso é considerada uma implementação adicional quando não é alterado nenhum processo já existente, apenas acrescentada uma nova condição para um processo que já existe ser executado.
2. **Alteração adicional com ressalvas**
   Uma alteração é considerada adicional com ressalvas quando, é adicionada uma condição para a execução de um processo já existente, entretanto, esse processo necessita de adaptações para a condição adicionada.
3. **Implementação de um processo**
   Uma alteração onde existe a implementação de um processo antes não existente, é uma alteração de implementação.

Categorias de Complexidade:

Baixa (1)

- Alteração de relatórios;
- Alterações meramente visuais;
- Categorias de erros de baixa complexidade:
  - "Field not found";
  - "Index out of bound";
  - "Focar em campo invisível";
  - "Erros de estado da query";
- Modificações nos eventos mais comuns dos componentes:
  - OnExit;
  - OnClick;
  - OnEnter;
  - OnDlgClick;
  - OnChange;

Média (2)

- Cláusulas extensas com inúmeros JOIN, funções de agregações, etc;
- Criação de campos e tabelas;
- Métodos/Funções de Rotinas Comuns;
- Criação/Implementação de relatórios;
- Modificações nos eventos dos componentes:
  - OnKeyDown;
  - OnKeyPress;
  - OnKeyUp;

Alta (3)

- Eventos de componentes do tipo TFDQuery;
- Eventos de componentes não citados na complexidade anterior;
- Rotinas de transmissão (específica por módulo: SPED, eSocial, notas em geral);
- Rotinas de Integração entre sistemas;

Estabelecer prazo "máximo" antes de marcar pessoas do nível de complexidade "pedindo" revisão

Revisões Conjuntas

Definição

Para maior assertividade nas revisões feitas, as revisões feitas dessa maneira, são realizadas por dois desenvolvedores conjuntamente. Onde contará com um desenvolvedor "especialista" do sistema (**Dev 1**), ou seja, que trabalha ou trabalhou na manutenção do sistema em questão e que possui conhecimento intermediário/avançado das rotinas alteradas que serão revisadas. A revisão também será composta por outro desenvolvedor (**Dev 2**), que não conhece propriamente os processos alterados do sistema e que será auxiliado/guiado pelo Dev 1.

Como funciona

A revisão conjunta visa incentivar e proporcionar o intercâmbio de conhecimento entre o Dev 1 e o Dev 2 do caso, para que isso seja efetivo, é necessário que o Dev 2 vivencie toda a rotina do processo em questão no caso e que consiga reproduzir a situação relatada no caso com o auxílio do Dev 1.

Normas e Instruções de uso do svn

Branch da versão

O branch da versão é a cópia de trabalho do trunk utilizada para testes e liberação. Isso garante a estabilidade do trunk, onde o teste é realizado em uma versão paralela e sempre atualizada, enquanto o trunk não é alterado. Este fato contribui para a liberação mais rápida e eficiente de versões do sistema, além de facilitar a liberação de casos emergenciais.

Quando o setor de Qualidade solicitar a atualização e geração do caso para testes, o caso deve ser seguido os passos de "Realizando um merge da pasta do caso no Branch".

Após os testes na versão do caso, o caso estará disponível para entrar na versão do sistema, porém isso só será permitido quando a Qualidade solicitar. Para realização dessa etapa deverão ser seguidos os passos de Incluindo um tratamento do branch da versão.

Neste documento estão incluídas instruções e normas de trabalho de uso do svn para seguir a nova forma de trabalho.

Controle de Versões - Supersoft-ERP.png
Image not found or type unknown

Iniciando um novo caso

1. Clicar com o botão direito na pasta https://svn.supersoft.com.br/svn/desenvolvimento/trunk/Codigo Base Skin
2. Selecionar a opção "Copy To".
3. https://svn.supersoft.com.br/svn/desenvolvimento/branches/Programador/Caso_0000000
   - (substituir "Programador" pelo nome da pasta referente ao programador)
   - (substituir "0000000" pelo número do caso)

4. Baixar a pasta referente ao caso em C:.
5. Renomear a pasta para o nome "Código fonte".

Baixando um caso do branch para o computador

1. Clicar com o botão direito em C: e selecionar a opção SVN Checkout...
2. Em “Url”, navegar até a pasta do caso e clicar em Ok.
   - Exemplo: “https://svn.supersoft.com.br/svn/desenvolvimento/branches/Tales/Caso_0038751”
3. Em “Checkout directory” utilizar a “C:\Caso 0038751” ou diretamente C:\Codigo Fonte (caso não tenha outra pasta já criada com este nome).

Ou

1. Clicar com o botão direito em C: e selecionar a opção Tortoise SVN -> Repo Browser
2. Navegar até a pasta do caso.
   - Exemplo: “https://svn.supersoft.com.br/svn/desenvolvimento/branches/Tales/Caso_0038751”
3. Clicar com o botão direito e selecionar a opção CheckOut.
4. Em “Checkout directory” utilizar a “C:\Caso 0038751” ou diretamente C:\Codigo Fonte (caso não tenha outra pasta já criada com este nome).

Realizando um merge da pasta do caso no Branch

1. Commite todas as alterações necessárias no branch.
2. Caso tenha alguma alteração não necessária realize o reverte ou exclua a unit e baixe novamente.

3. Clique com o botão direito na pasta -> TortoiseSVN -> Merge
4. Utilize o diretório abaixo para realizar o merge
   - https://svn.supersoft.com.br/svn/desenvolvimento/trunk/CodigoBaseSkin
5. Em *merge depth* (profundidade da fusão), selecionar a opção *fully recursive* (totalmente recursivo).
6. Realizar um Test merge para verificar possíveis conflitos.
7. Em caso de conflito em alguma unit, verificar e tratar o conflito.
8. Commitar o *merge* (o merge não deve ser commitado junto com alterações relativas ao caso).

Incluindo um tratamento no Branch da Versão

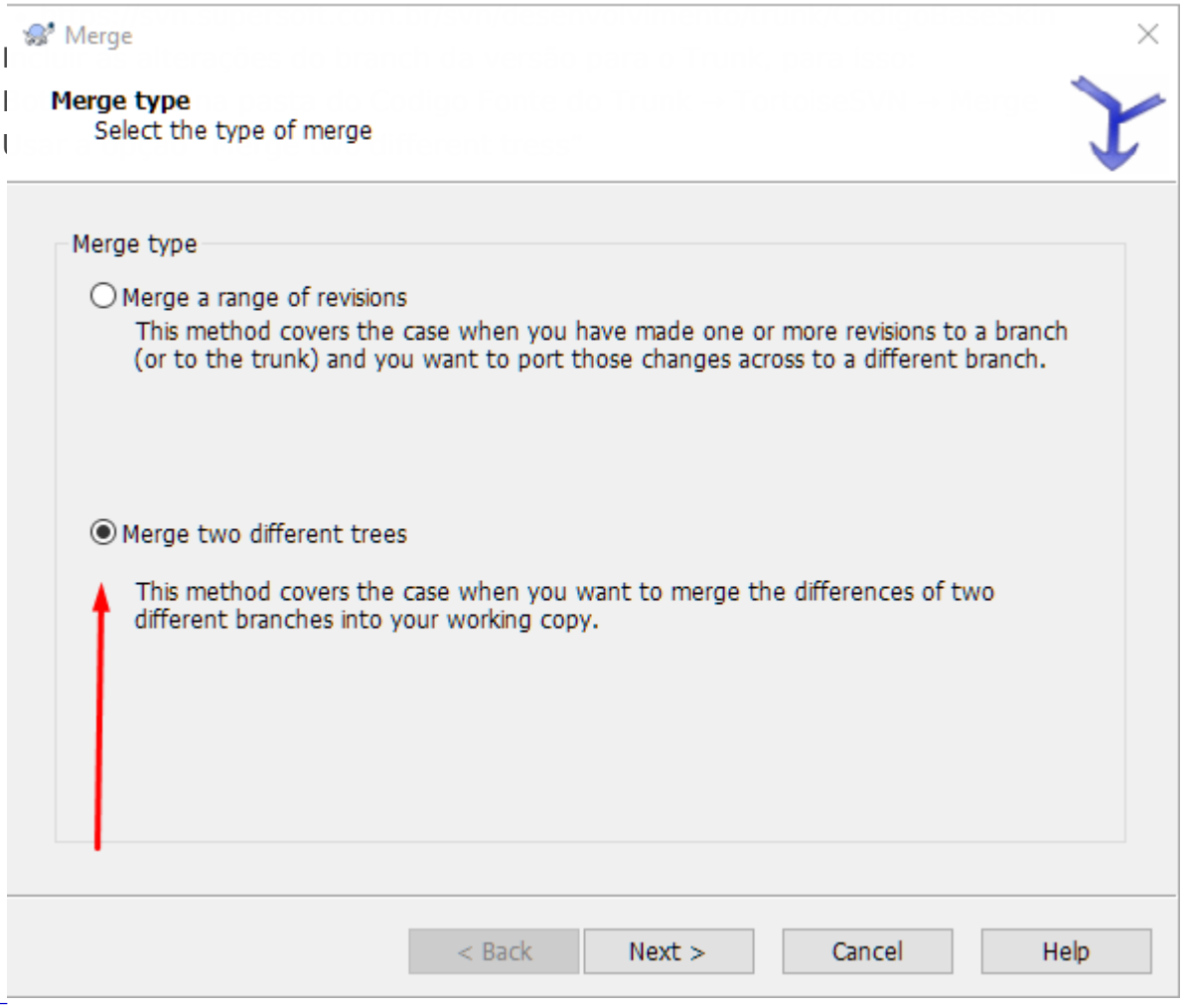
1. Baixar a pasta a seguir para o computador.
   - https://svn.supersoft.com.br/svn/desenvolvimento/branches/Versoes/ERP/001-Caso-0041704
2. (substituir o nome da pasta para o nome da pasta da versão atual de trabalho).
3. Incluir as alterações do caso.
4. Realizar o commit na pasta da versão.

Atualizando o Branch da versão

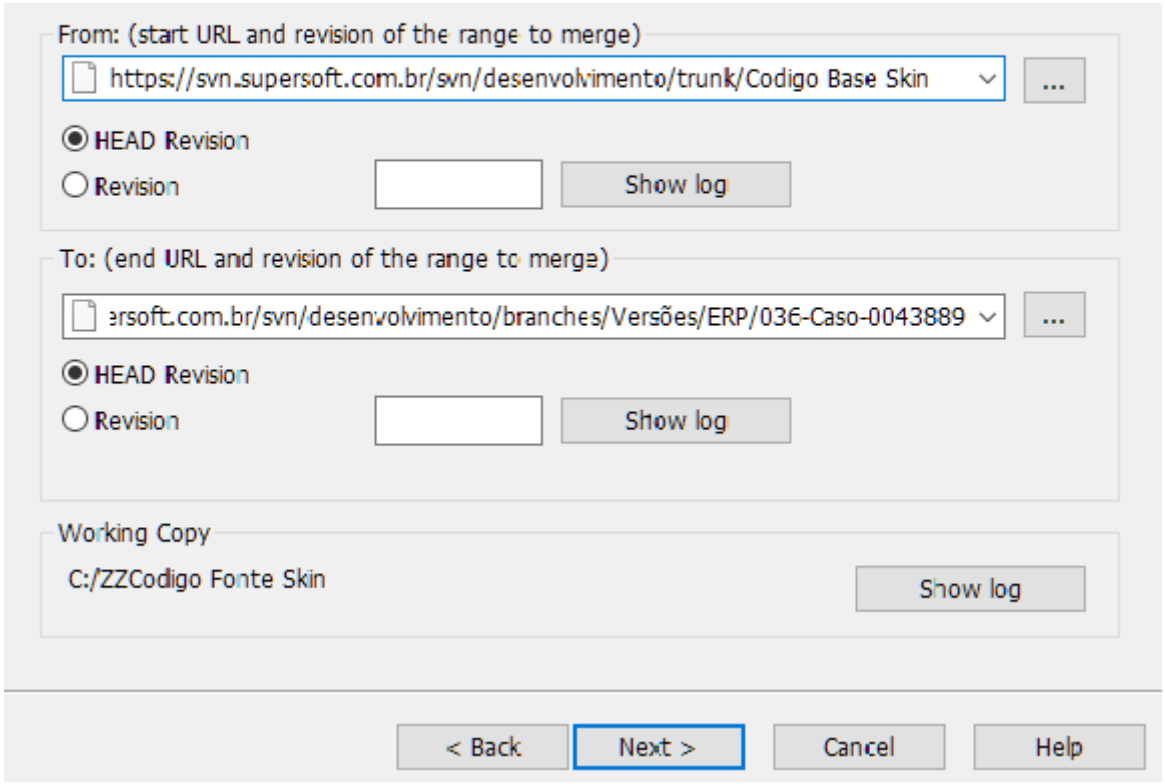
1. Clique com o botão direito na pasta -> TortoiseSVN -> SVN Update
2. Clique com o botão direito na pasta -> TortoiseSVN -> Merge
3. Utilize o diretório abaixo para realizar o merge
   - https://svn.supersoft.com.br/svn/desenvolvimento/trunk/CodigoBaseSkin
4. Em *merge depth* (profundidade da fusão), selecionar a opção *fully recursive* (totalmente recursivo).
5. Em caso de conflito em alguma unit, verificar e tratar o conflito.
6. Commitar o *merge*.

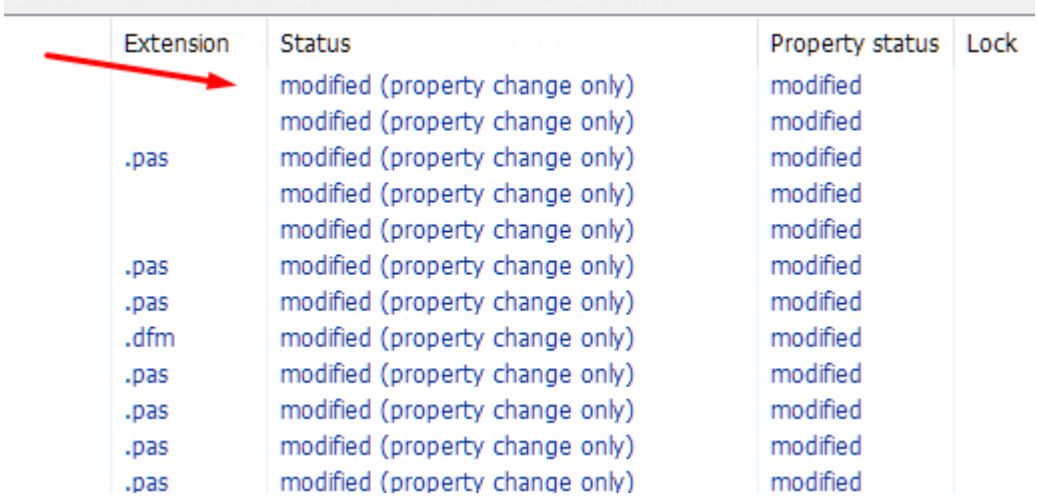
Incluindo Branch da Versão no Trunk

1. Realizar o merge do branch da versão.
2. Baixar a pasta a seguir para o computador.
3. I
4. I
5. I



1. Em “From” colocar o caminho do Código Fonte Trunk.
2. Em “To” colocar o caminho do Código Fonte do Branch da versão.



3. Next → Fully Recursive → (Teste Merge/Merge)
4. Clique com o botão direito na pasta -> Commit
5. 

Extension	Status	Property status	Lock
	modified (property change only)	modified	
	modified (property change only)	modified	
.pas	modified (property change only)	modified	
	modified (property change only)	modified	
	modified (property change only)	modified	
.pas	modified (property change only)	modified	
.pas	modified (property change only)	modified	
.dfm	modified (property change only)	modified	
.pas	modified (property change only)	modified	
.pas	modified (property change only)	modified	
.pas	modified (property change only)	modified	
.pas	modified (property change only)	modified	

6. As Units que restaram são tratamentos do Branch em questão, mas vale fazer algum tipo de validação.

7. Realizar o commit no Trunk.

Notações: Menus e Tabela/Imagens

Menus

As notações serão muito utilizadas no preenchimento da plataforma ao definir um processo dos sistemas existente.

Todo sistema contém seus menus e seus submenus adjacentes, então ao definir um caminho, por exemplo o menu: Movimentação > Rotinas Mensais > Alteração Salarial, que irá efetuar uma alteração salarial dos funcionários, deve ser definido da seguinte forma.

Utilize o ultimo nome do que o caminho descrito irá fazer e adicione um número sequencial, iniciando do 1 até o numero necessário e logo em seguida adicione no número a opção de sobrescrito e link para a parte de legenda dos caminhos do processo, conforme será exemplificado à seguir:

Alteração Salarial [1]

Após o usuário clicar no link, será redirecionado para a parte de legenda dos caminho, podendo visualizar o caminho especifico e os demais contidos naquela página.

Legendas dos caminhos

> Alteração Salarial [1]: Movimentação > Rotinas Mensais > Alteração Salarial

Tabela ou Imagem

Ao adicionar um conjunto de informações correlacionadas, você pode ficar em dúvida em adicionar uma imagem com as informações ou uma tabela. Então você deve levar a seguinte pergunta em questão, o conteúdo a ser adicionado, pode vir a ser utilizado, de maneira que a pessoa copie as informações para colar em outro local?

Então é mais vantajoso adicionar uma tabela, que possibilitará a pessoa de copiar os conteúdo e colar em outro local, diferentemente da imagem, que não é possível copiar o conteúdo contido nela.

Pautas (Maio)

Padrão de Código (Delphi) e Refatorações

Quebra de linha em operação aritmética

```
// Opção A
begin
  Objeto.Atributo :=
    LQuery.FieldByName('Campo').AsFloat +
      LVariavelDouble +
      LOutraVariavelDouble;
end;

// Opção B
begin
  Objeto.Atributo :=
    LQuery.FieldByName('Campo').AsFloat +
    LVariavelDouble +
    LOutraVariavelDouble;
end;

// Opção C
begin
  Objeto.Atributo :=
    LQuery.FieldByName('Campo').AsFloat +
      LVariavelDouble +
        LOutraVariavelDouble;
end;
```

Definir padrão para:

Operações com apenas um tipo de operador

Operações com vários tipos de operadores

Utilização de constantes

```
// Declaração "padrão"

// Opção A
// Unit separada por tópico (Entidade.Constantes.pas)

// Opção B
// Mesma unit, escopo global

// Opção C
// Declaração inicial no mínimo escopo viável, "evoluindo" conforme sua expansão

//***********************************************//

// Método de refatoração

// Opção A
// Pontual (Apenas onde será alterado)

// Opção B
// Por método (Em todas as ocorrências no método alterado)

// Opção C
// Por unit (Em todas as ocorrências na unit alterada)
```

Padrão SQL

Indentação INSERT INTO

```
// Opção A
INSERT INTO Tabela
            (CampoA, CampoB, CampoC, CampoD,
             CampoE, CampoF)
     VALUES (:PA, :PB, :PC, :PD, :PE, :PF)

// Opção B
INSERT
  INTO Tabela
       (CampoA, CampoB, CampoC, CampoD,
        CampoE, CampoF)
```

```
VALUES (:PA, :PB, :PC, :PD, :PE, :PF)
```

Indentação DISTINCT

```
/* Encontrar "média" comum */
/*
Tabela: Carros
Campos: (PK)Placa, Modelo, Ano
Condição de busca: Um modelo por ano no intervalo de 2000-2021
*/
```

Formalização de indentação com múltiplos operadores lógicos

```
SELECT *
  FROM Tabela
 WHERE CampoA = :ParametroA
   AND (   CampoB = :ParametroB
        OR CampoC = :ParametroC)
```