Script de Desenvolvimento

Documento utilizado para detalhamento das informações referentes ao tratamento realizado no caso.

Definição

Definição

Instruções para preenchimento do Script de Desenvolvimento

O que é o Script de Desenvolvimento?

O Script de desenvolvimento é um documento que deve ser preenchido pela equipe de desenvolvimento e encaminhado à equipe de Qualidade quando o status do caso passar para “Aguardando testes”.

Sendo que todo o caso, deverá conter necessariamente um script de desenvolvimento.

No script deverão conter detalhadamente todas as informações referentes ao tratamento realizado no caso. Essas informações deverão ser preenchidas de uma maneira clara e objetiva, facilitando o entendimento da equipe de qualidade e de possíveis outros setores de qual era o problema relacionado ao caso, o que foi feito para corrigi-lo e qual o resultado esperado.

O documento será disponibilizado nas pasta do caso através do SVN, no formato .ODT. O mesmo deve ser preenchido após o termino do tratamento e ter as alterações adicionadas ao SVN.
Para gravar o arquivo depois do preenchimento, usar o padrão "Script_Desenvolvimento_ + Numero Caso". Exemplo: "*Script_Desenvolvimento_0043310*"

Na conclusão do caso, a Qualidade esta responsável por gerar um arquivo .PDF e disponibilizar no caso do Mantis, para acesso dos setores de Suporte, Projetos e demais interessados. O objetivo é fornecer a informação das alterações realizadas para demais setores envolvidos.

Estrutura do Script

Para auxiliar no preenchimento do script de desenvolvimento corretamente, as seguintes informações deverão ser fornecidas:

Caso 00XXXXX - https://wiki.supersoft.com.br/desenvol/view.php?id=XXXXX

Programador: Nome

#	RESUMO	SIM	NÃO
1	Criação de tabela(s)		
2	Criação de campo(s) em tabela(s) já existente		
3	Criação de menu(s) / tela(s)		
4	Alteração em tela(s) já existente		
5	Alteração de lógica / validação / processo		
6	Correções de erros		
7	Adição de pastas / arquivos na instalação do sistema		

Preenchimento do script de desenvolvimento

Neste tópico, serão passadas algumas orientações à respeito das informações que devem ser fornecidas em cada tópico do script, para exemplos práticos verificar o capítulo de exemplos de scripts de desenvolvimento.

Resumo

Nesse quadro é indicado quais os tipos de alterações realizadas no caso através da marcação das opções com "X".

Sistemas alterados

Neste tópico é preenchido quais módulos foram alterados com o tratamento e devem ser liberados.

Detalhamento das alterações

Nesse tópico esperamos que seja apontado de forma clara e objetiva onde de fato estava o problema e o que foi corrigido ou alterado para solucioná-lo. Observe que, quando bem respondida esta questão, é fácil identificar o que exatamente foi corrigido, implementado ou alterado, facilitando e simplificando o processo de testes por parte da equipe da qualidade.

Neste tópico, pode-se indicar qual caminho é feito para chegar até a tela alterada, utilizando a seguinte notação

```
Menu > Sub-menu > Sub-Menu > Sub-menu > Sub-menu
```

Caso seja necessário realizar alguma ação na tela alterada, como preenchimento de algum campo, ativação de algum evento (saída, entrada, mudança de aba, etc), isso também deve ser respondido neste tópico.

Além disso, qualquer melhoria ou correção feita que esteja fora do escopo inicial do caso, também deve ser informada neste espaço.

É opcional a adição de imagens ilustrativas (print screen), para destacar algum menu/tela/campo novo, alteração de descrição de componentes, modificações em telas, etc. Em caso de novas implementações, é recomendado o uso de imagens, para ficar claro para os demais setores (Suporte/Projetos), quais foram as alterações visuais.

Resultados esperado

Indicar qual o comportamento esperado para o sistema nos locais onde os testes devem ser validados de maneira descritiva.

Validações

Este tópico é de suma importância para garantir que os testes sejam feitos de maneira completa e efetiva. Para isso, o desenvolvedor deve especificar quais telas devem ser testadas e sob quais circunstâncias, podendo informar alguns exemplos iniciais para que a equipe de qualidade, ao elaborar/seguir o roteiro de testes considere tais situações e elabore situações semelhantes que considerem relevantes para a validação do teste.

> Nota: Não cabe ao desenvolvedor informar COMO os testes devem ser realizados e sim o que deve ser levado em consideração ao validar as implementações feitas.

Caso está no branch de versão?

Informar apenas com SIM ou NÃO.

> Nota: Idealmente, os casos nunca deverão ser vinculados à algum branch de versão sem antes serem testados e validados nos branches dos casos. Então, espera-se que a resposta

> para essa pergunta seja "não" na grande maioria das vezes.

Units alteradas/adicionadas/eliminadas

Neste espaço do script, recomenda-se que o desenvolvedor informe brevemente as alterações realizadas (de maneira semelhante as mensagens de versionamento), assim, caso ocorra de outro desenvolvedor trabalhar em uma situação relacionada à essa, ele consiga ter uma noção do contexto em que foi realizado tais alterações.

Após isso, deve-se relatar quais units foram adicionadas ou alteradas. Por exemplo:

Units adicionadas
Pasta_pai/pasta_filha/Unit1.pas (Unit criada para implementação da classe X)

Units alteradas
Pasta_pai/pasta_filha/Unit2.pas (Tela de cadastro/alteração da entidade Y)
Pasta_pai/pasta_filha/Unit3.pas (Tela de cadastro/alteração da entidade W)
Pasta_pai/pasta_filha/Unit4.pas (Tela de geração dos recibos A, B, C)

Ajustes e padronização de código
Pasta_pai/pasta_filha/Unit.pas (Pequenos ajustes e padronização de código. Validar tela Z)

> Nota: A alteração/criação de novos campos, tabelas, etc., também deverão ser relatadas nesse tópico.

Exemplos

Capítulo destinado à exemplos de scripts de desenvolvimento preenchidos e situações que podem ser semelhantes em outros casos

Exemplo de Script de Desenvolvimento - Casos de erro

Neste documento iremos citar exemplos práticos de como devem e como não devem ser preenchidas cada uma das perguntas referentes ao Script de Desenvolvimento. As respostas devem ser sempre claras e objetivas, facilitando o entendimento da equipe de Qualidade e dos demais outros setores de qual era o problema do caso, o que foi feito para corrigi-lo e qual o resultado esperado.

Para saber mais sobre o que é o Script de Desenvolvimento e sua estrutura, acessar o capítulo Definição.

Utilizaremos o caso 0043310 como exemplo para o preenchimento das etapas do Script de Desenvolvimento.

Resumo geral

Forma correta de preenchimento:

> | # | RESUMO | SIM | NÃO |
> |---|---|---|---|
> | **1** | Criação de tabela(s) | | X |
> | **2** | Criação de campo(s) em tabela(s) já existente | X | |
> | **3** | Criação de menu(s) / tela(s) | | X |
> | **4** | Alteração em tela(s) já existente | X | |
> | **5** | Alteração de lógica / validação / processo | X | |

6	Correções de erros		X
7	Adição de pastas / arquivos na instalação do sistema		X

Neste exemplo podemos verificar que foi criado campo(s) em tabela(s) já existente, alteração em telas já existente e alteração de lógica ou funcionamento de algum processo. Com essas informações é possível ter uma noção inicial de quais tipos de validação deveram ser realizadas no caso.

Forma incorreta de preenchimento:

#	RESUMO	SIM	NÃO
1	Criação de tabela(s)		
2	Criação de campo(s) em tabela(s) já existente	X	
3	Criação de menu(s) / tela(s)		
4	Alteração em tela(s) já existente	X	
5	Alteração de lógica / validação / processo	X	
6	Correções de erros		
7	Adição de pastas / arquivos na instalação do sistema		

Neste exemplo podemos que não foram preenchidos os campos que não tiveram alterações, mas é importante a indicação do mesmo em todos os casos.

Sistemas alterados

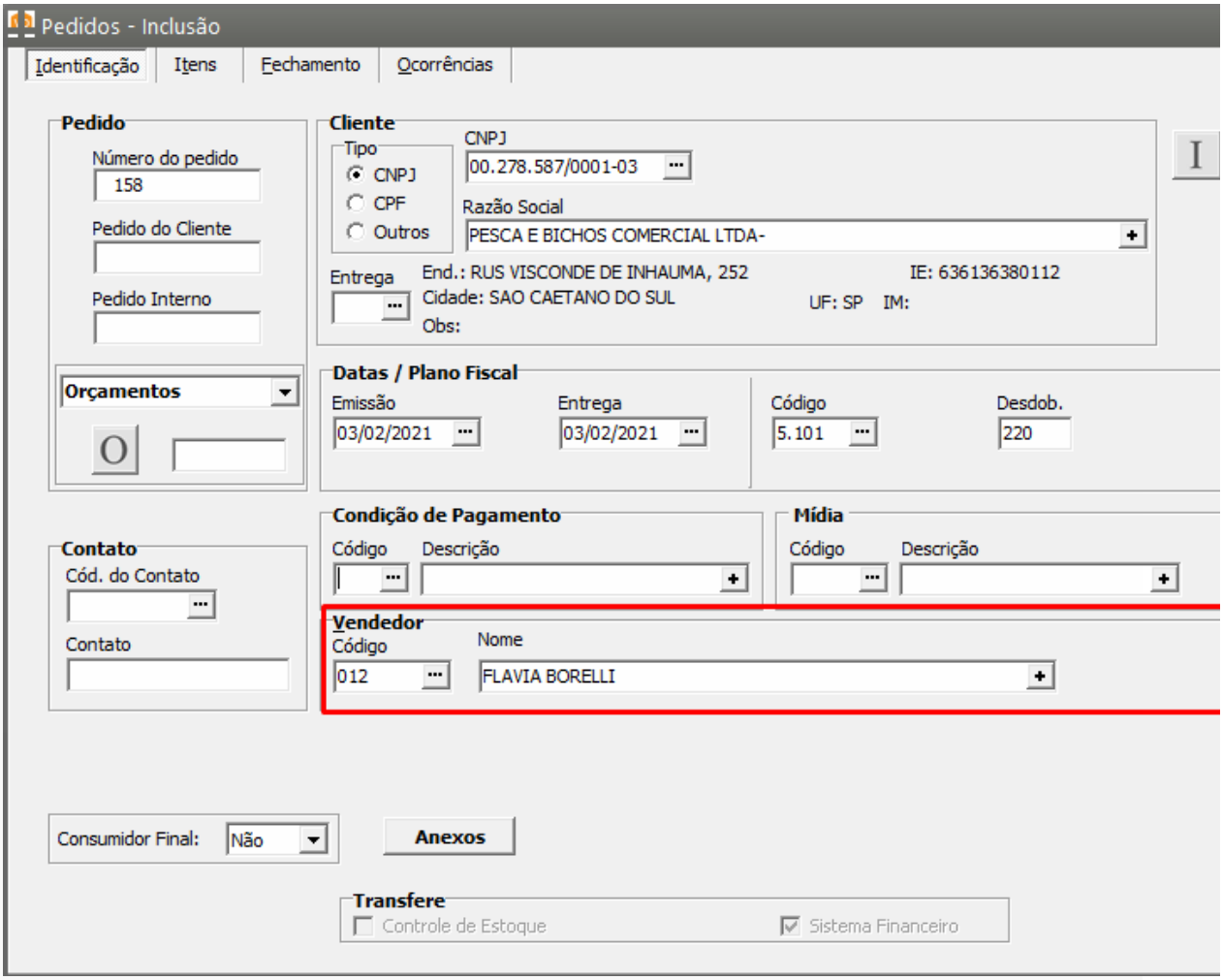
> VENDAS, COMPRAS, ESTOQUE

Nesse tópico deve se atentar em indicar quais módulos foram alterados, para que sejam realizados os testes corretamente e posteriormente sejam liberados para os clientes.

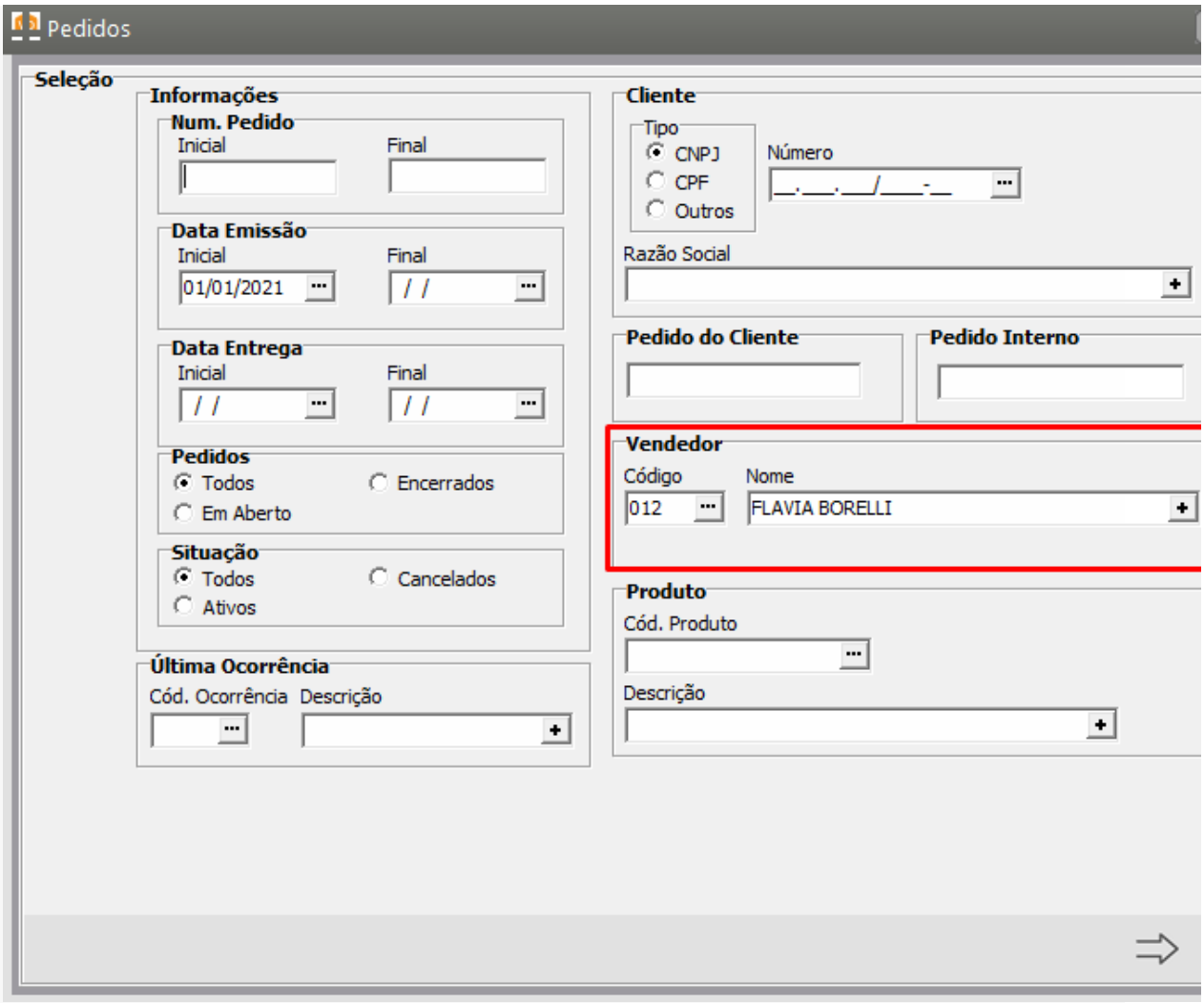
Detalhamento das alterações

Forma correta de preenchimento:

> “Corrigido o tratamento que verifica se o USUÁRIO logado possuí cadastro no menu
> 'Manutenção >Vendedores > Login de Vendedor > Cadastra/Manutenção', e faz a atribuição do código nos documentos de Orçamento, Pedido e Nota Fiscal.
>
> No cadastro de documentos:
> - No cadastro de um segundo documento sem fechar a tela, não estava atribuindo corretamente o Código do Vendedor do Usuário logado. Realizado correção;



Na tela de manutenção de documentos:
- Padronizada a função para que seja feita da mesma forma que dentro dos documentos;
- Criada uma validação que trava os campos de 'Vendedor', quando existir registro em 'Login de
Vendedor'.



OBSERVAÇÃO: Não foram tratadas nesse caso as telas de 'Clientes' e 'Interessados'.”

Neste exemplo podemos identificar facilmente onde é realizado o vínculo entre usuário/vendedor (menu "Manutenção > Vendedores > Login de Vendedor > Cadastra/Manutenção") e que o problema é que a atribuição do código do vendedor não estava sendo feita corretamente quando cadastrado um segundo documento, podendo ser Orçamentos, Pedidos ou Notas Fiscais.
Além de ter corrigido o erro principal relatado no caso, o desenvolvedor realizou melhorias nos menus de manutenção de documentos, especificando o que foi feito.
Foi utilizado imagens ilustrativas para destacar os campos que tiveram impacto no tratamento realizado (opcional).
Observar também que o desenvolvedor deixa claro que neste caso não foram tratados os menus de "Clientes" e "Interessados".

Forma incorreta de preenchimento:

> “Corrigido o tratamento que verifica se o USUÁRIO logado possuí vínculo de usuário/vendedor e não estava fazendo a atribuição do código do vendedor corretamente nos documentos.
>
> Realizado ajustes para padronização na tela de manutenção dos documentos."

Verificar neste exemplo que o desenvolvedor nos diz que o erro do caso é referente à não atribuição correta do código do vendedor nos documentos quando o mesmo possui vínculo de usuário/vendedor, porém não deixa claro qual menu do sistema pode ser consultado o cadastro de usuário/vendedor, quais documentos apresentam o problema e nem que o problema acontecia somente no cadastro de um segundo documento.
O desenvolvedor cita que foram realizados ajustes para padronização do menu de manutenção de documentos, porém também não especifica o que foi feito. Também não foi citado que este caso não tratou os menus de "Clientes" e "Interessados", deixando brecha para que outros departamentos entendam que os mesmos também foram corrigidos.

Resultados esperado

Forma correta de preenchimento:

> “Em todas as telas de cadastro e edição, o resultado deve ser a atribuição correta do Código e Nome do Vendedor quando o usuário logado no sistema possuir cadastro de "Login de Vendedor".
> Para usuários que não possuírem registro de "Login de Vendedor" não deve ser atribuído automaticamente nenhum código de vendedor no cadastro, edição ou pesquisa (menu manutenção) dos documentos."

Neste exemplo o desenvolvedor deixa claro qual comportamento o sistema deve ter quando o usuário logado no sistema possuir registro de "Login de Vendedor" e quando o usuário não possuir.

Forma incorreta de preenchimento:

> "Atribuir corretamente o Código e Nome do Vendedor quando o usuário logado no sistema possuir cadastro de "Login de Vendedor".

Não deixa claro em quais documentos deve ser atribuído corretamente o Código e Nome do Vendedor, se deve ser incluído somente no cadastro, se somente na edição, na consulta (manutenção) de documentos, etc.
Também não cita o que deve acontecer quando o usuário logado no sistema NÃO possuir registro de "Login de Vendedor".

Validações

Forma correta de preenchimento:

> Telas de manutenção de Orçamento, Pedido e Nota Fiscal (todas as Notas e Recibo):
>
> - Usuário que possuí "*Login* de Vendedor";
> - Usuário que NÃO possuí "*Login* de Vendedor";
> - Usuário "SUPERVISOR":
>   - Abertura das telas e atribuição das informações nos campos de 'Código de Vendedor' e 'Nome';
>   - Avançar para a próxima aba (Botão 'Próximo') e voltar e validar se as informações ficaram
>   corretas.
>
> Telas de cadastro de Orçamento, Pedido e Nota Fiscal (todas as Notas e Recibo):
>
> - Usuário que possuí "*Login* de Vendedor";
> - Usuário que NÃO possuí "*Login* de Vendedor";
> - Usuário SUPERVISOR:
>   - Abertura das telas e atribuição das informações nos campos de 'Código de Vendedor' e 'Nome'
>   (Modo: 'Novo', 'Alteração' e 'Visualização');
>
> Validar quando a empresa utiliza Comissão com vários vendedores ('Manutenção > Parâmetros
> Adicionais da Empresa > Comissão de Vendedores > "Tem mais de um vendedor por nota"').
> Validar os itens citados nos ajustes e padronização do código.”

Neste exemplo o desenvolvedor deixa claro que deverão ser realizados testes nas telas de manutenção (edição) e cadastro de Orçamentos, Pedidos e Notas (todos modelos da NFmodel1), citando alguns exemplos do que validar. É especificado também que os testes deverão ser realizados com usuários que possuam "Login de vendedor", usuários que NÃO possuam "Login de vendedor" e com o usuário Supervisor.

É citado que deverão ser realizados testes com empresas que utilizam comissão com vários vendedores e especificado qual o caminho para configurar tal parâmetro.
Lembrando: No Script de Desenvolvimento o desenvolvedor deverá descrever o que e onde validar, porém caberá ao departamento de Qualidade analisar o caso, o que foi feito e então elaborar um Roteiro de Testes para fazer a validação do tratamento.

Forma incorreta de preenchimento:

> "Realizar o cadastro de documentos e verificar se o código de vendedor será preenchido corretamente quando o usuário logado no sistema possuir vínculo de "Login de vendedor"."

Neste exemplo o desenvolvedor orienta para realizar testes no cadastro de documentos e verificar se o código do vendedor será atribuído corretamente quando o usuário logado no sistema possuir vínculo de "Login de vendedor", porém não orienta em quais documentos houveram alteração e precisarão ser validados, sobre testes com usuários que não possuam "Login de vendedor" e com usuário Supervisor. Também não é orientado sobre testes com outras parametrizações, como por exemplo se estiver assinalado o parâmetro que utiliza mais de um vendedor por nota.

Caso está no Branch da Versão?

> Não.

Resposta apenas com "SIM" ou "NÃO", portanto não há preenchimento "certo" ou "errado".
Obs: Idealmente os casos nunca deverão ser vinculados diretamente ao Branch da Versão sem antes serem testados e validados no Branch do Caso.

Desenvolvimento - Units alteradas/adicionadas/eliminadas:

Forma correta de preenchimento:

> “DESENVOLVIMENTO
> Criado a classe 'Vendedor.Classes' para adicionar função relacionadas a Vendedores, como no caso, a função de atribuição de LOGIN DE VENDEDOR nos

> documentos.
> Classe e funções foram criadas para evitar replica desnecessária de código e para facilitar manutenção.
>
> **Units Adicionadas:**
> Sistemas/Rotinas Comuns/Classes/Vendedor.Classes.pas (Classe para funções de 'Vendedores')
>
> **Units Alteradas:**
> Sistemas/SSVendas/NFMod12.pas (Tela de Cadastro/Alteração de Notas Fiscais e Recibo)
> Sistemas/SSVendas/Orcamvd2.pas (Tela de Cadastro/Alteração de 'Orçamentos')
> Sistemas/SSVendas/Pedidos1.pas (Tela de Manutenção de 'Pedidos')
> Sistemas/SSVendas/Pedidos2.pas (Tela de Cadastro/Alteração de 'Pedidos')
>
> **Ajustes e padronização de código:**
> ClassesD10/ParâmetroDoSistema.pas (Apenas ajustes/padronização de código. Validar a
> gravação do menu "Parâmetros do Sistema");
> Sistemas/SSVendas/DetComis.pas (Apenas ajustes/padronização de código. Validar
> preenchimento de comissão);
> Sistemas/SSVendas/Pedidos4.pas (Apenas ajustes/padronização de código. Validar busca de
> Produtos no Pedido).”

Neste exemplo o desenvolvedor primeiramente faz um resumo para o próprio Desenvolvimento das alterações que foram realizadas, depois deixa claro todas as Units que foram adicionadas, alteradas e as que realizou algum tipo de ajuste e padronização de código.
Em frente de cada Unit é especificado qual o Menu/Tela do sistema está relacionado à Unit, dessa forma é fácil para a equipe da Qualidade ou para qualquer outro departamento saber exatamente em quais menus/telas do sistema houveram alterações.

Forma incorreta de preenchimento:

> “**"Units Adicionadas:**
> Sistemas/Rotinas Comuns/Classes/Vendedor.Classes.pas
>
> **Units Alteradas:**
> Sistemas/SSVendas/NFMod12.pas

> Sistemas/SSVendas/Orcamvd2.pas
> Sistemas/SSVendas/Pedidos1.pas
> Sistemas/SSVendas/Pedidos2.pas"

Neste exemplo não foi feito o resumo para o próprio Desenvolvimento sobre as alterações que foram realizadas.
O desenvolvedor apenas cita as Units que foram adicionadas e alteradas, porém não as relaciona com os menus/telas do sistema, dificultando para que a equipe da Qualidade e demais setores saibam exatamente quais menus/telas sofreram qualquer tipo de alteração no caso.

Processos do Script de Desenvolvimento

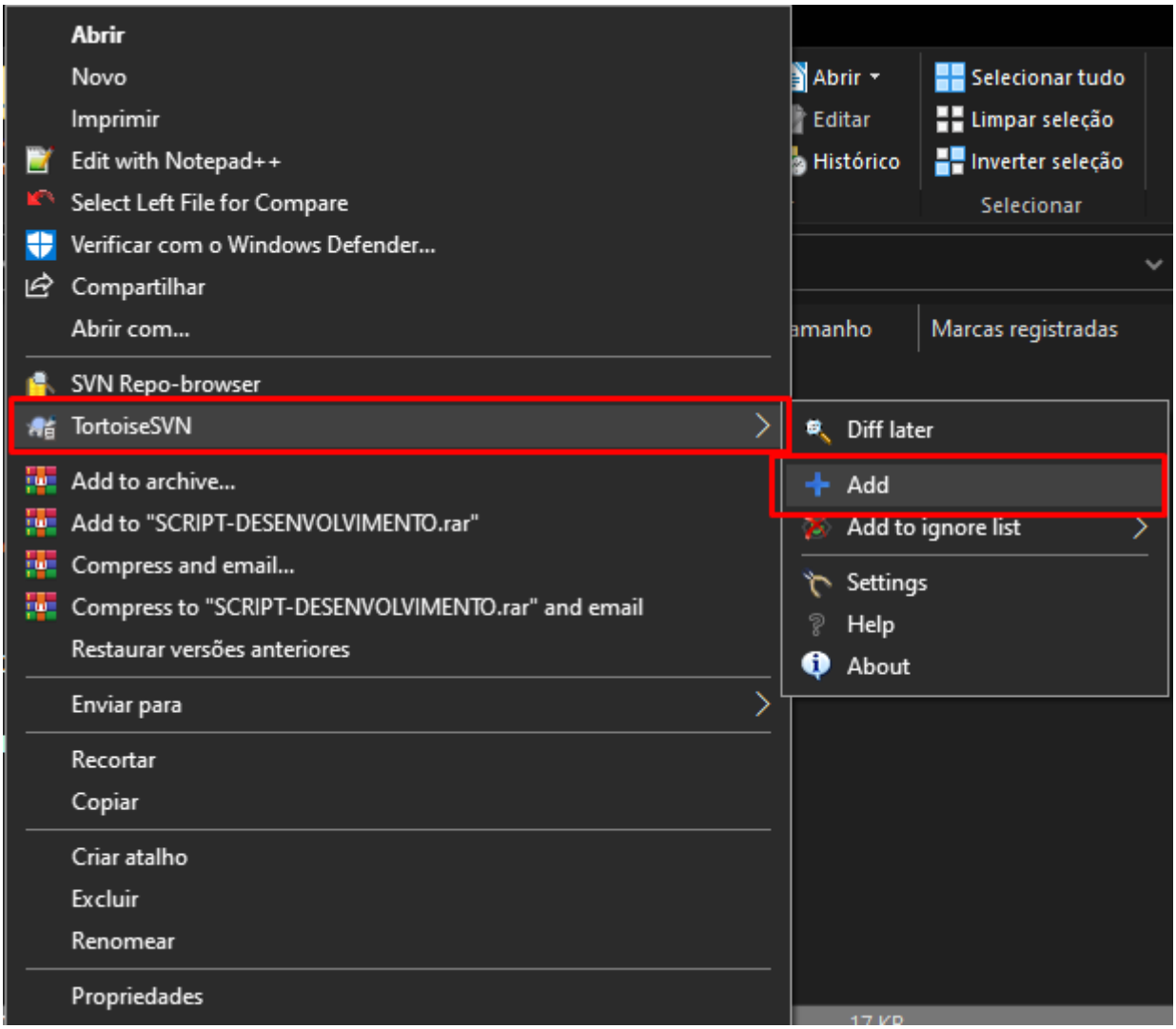
[BETA] Processo de criação, preenchimento e postagem - Desenvolvimento / Qualidade

Criação do Branch do caso e adição do Script de Desenvolvimento (Desenvolvimento):

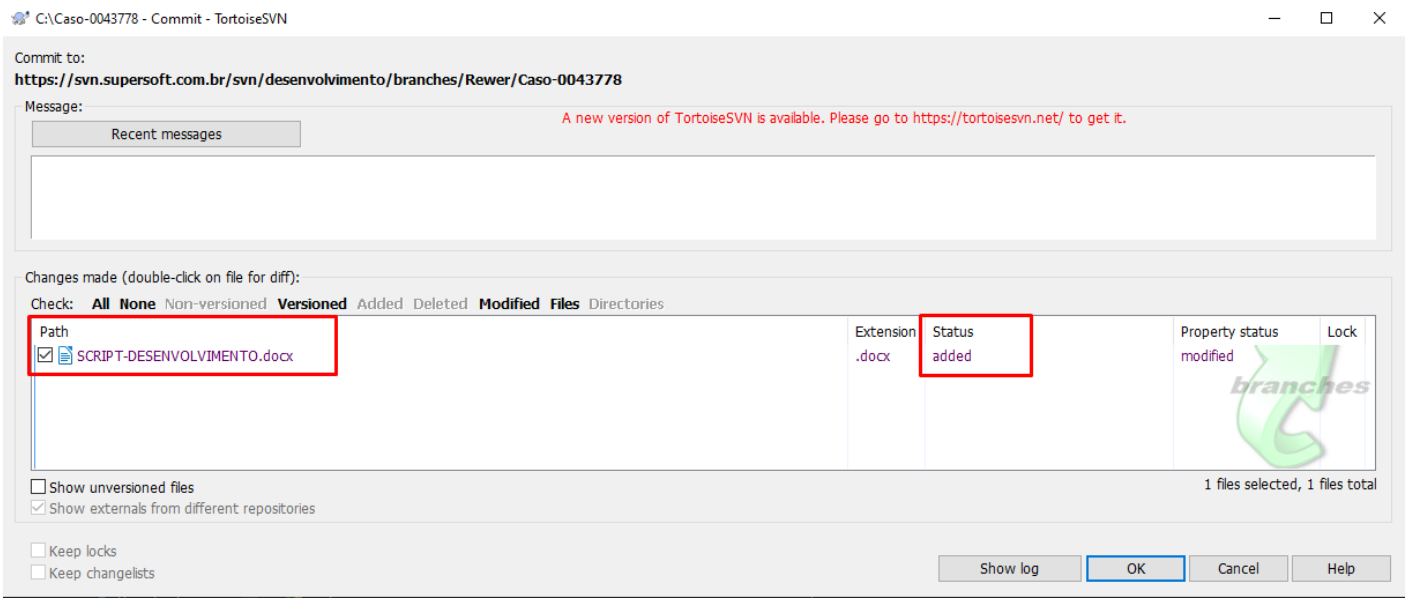
Inicialmente é necessário cria a pasta do Branch conforme a estrutura básica padrão (SVN - Estrutura básica).

Após a criação do Branch e *Checkout* do mesmo, adicionar o arquivo "Script_Desenvolvimento.odt" em anexo nesta página, dentro da pasta **"Documentos"**.

Em seguida é necessário realizar o processo de adição do arquivo ao projeto no SVN através do menu **"*TortoiseSVN > Add*"**:



O próximo passo é submeter (*commit*) a adição do arquivo, conforme exemplificado a seguir:

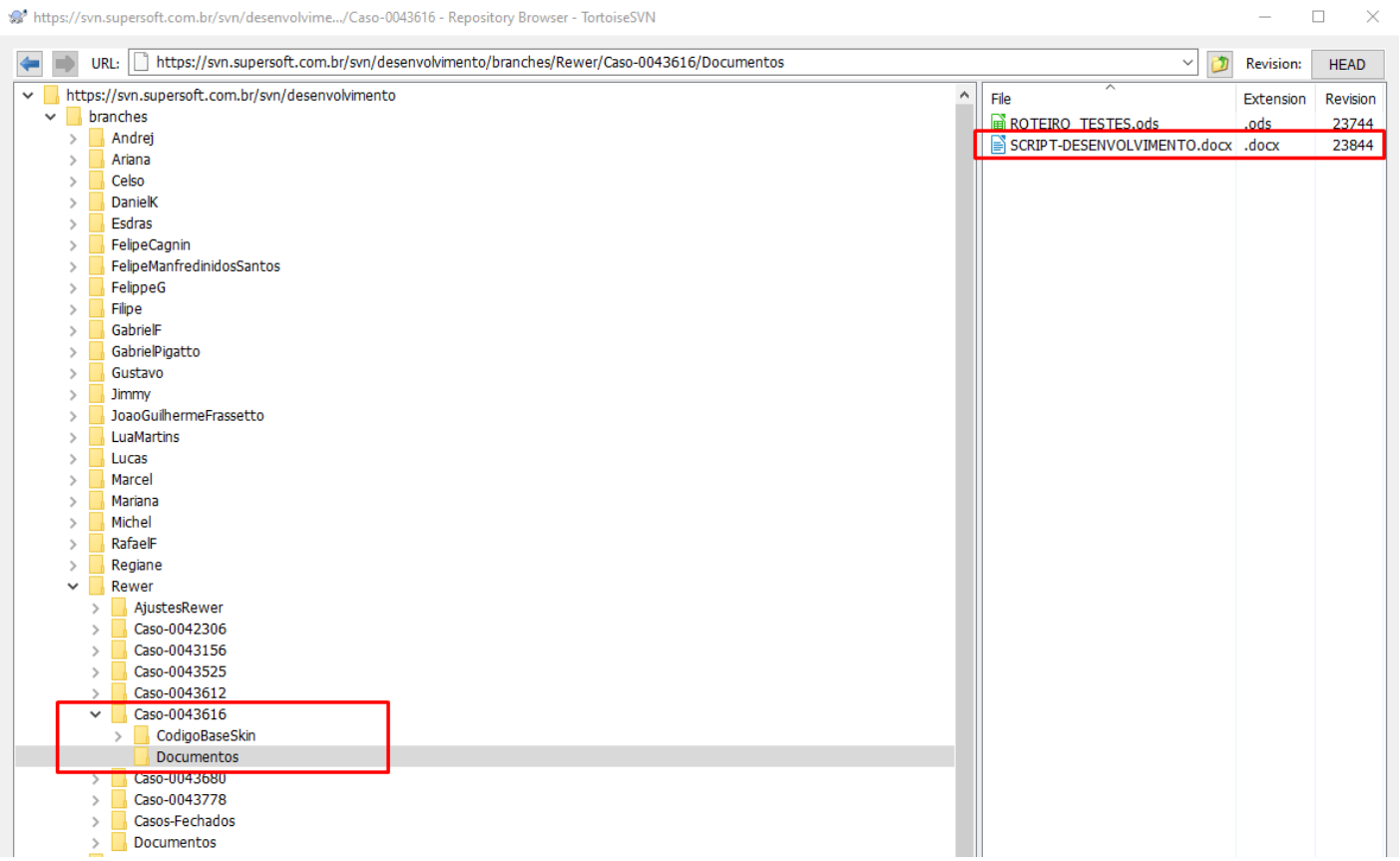


> **EXTRA:** Ao criar um novo caso, também será necessário realizar a adição do ROTEIRO_TESTES.ods, seguindo os mesmo passos dessa pagina.

Os documentos devem ficar organizados na seguinte disposição:

> *Pasta do Programador / Pasta do Caso / Pasta do Codigo Fonte + Documentos / Script de Desenvolvimento*

Exemplo: .../branches/Programador/Caso-43616/Documentos/SCRIPT-DESENVOLVIMENTO.docx (figura abaixo)



Alterações do Script de Desenvolvimento e *commit* no *SVN* (Desenvolvimento):

Após preenchimento do Script de Desenvolvimento, as alterações devem ser submetidas (*commit*) ao *SVN*, da mesma forma que é realizado com o código fonte.

Publicação do Script de Desenvolvimento no caso do Mantis (Qualidade):

Quando o caso for finalizado, realizado todas as etapas de testes e o mesmo for liberado em uma versão, a Qualidade deve gerar um arquivo .PDF do *Script* de Desenvolvimento e disponibilizar o mesmo no caso do Mantis.

O objetivo é que outros setores interessados, como Suporte, Projetos, etc, tenham acesso ao documentos para ficar informado sobre as alterações, implementações e correções realizadas no caso.