

AppManutencao

Projeto destinado à avaliação de contratação para equipe de Sustentação - Softplan. Siga as instruções deste README para realizar as correções e implementações necessárias.

Versão utilizada do Delphi: Tokyo 10.2. Pode ser utilizada outras versões (inclusive Community), desde que o código seja compatível.

Dica: antes de iniciar as alterações, leia todo o README.

Instruções

Correções a serem realizadas

- Implemente as correções e implementações descritas nas sessões abaixo.
- **Todos** os `hints` e `warnings` do projeto devem ser resolvidos.
Dica: sempre rode o build (Shift + F9), ao invés do compile (Ctrl + F9), para ver todos os hints e warnings.
- **Todos** os `memory leaks` do projeto devem ser resolvidos.

Como submeter uma correção

- Realize as implementações e correções descritas abaixo e nos envie o projeto de alguma forma: compactado por e-mail, link de algum repo (github, gitlab, etc.), ou faça upload para nuvem e nos envie o link (com acesso público).
- Envie o projeto limpo, apenas com os **mesmos arquivos enviados originalmente** (sem dcu, binário, etc.).

Implementações

Implementação 1: implemente um gerenciador de exceções no projeto. A cada exceção gerada, independente do ponto da aplicação que ocorra, deve passar pelo gerenciador, e a classe e mensagem da exceção devem ser salvas em um arquivo de log. Depois de salva a mensagem no log, a exceção deve ser levantada normalmente. Fica a critério do candidato se quiser criar uma tela específica para mostrar a exceção, ou se quiser tratar algumas exceções. Isso é opcional.

Implementação 2: na tela ClienteServidor, implemente o comportamento da barra de progresso para mostrar o progresso da operação de envio de arquivos.

Implementação 3 (opcional): na tela ClienteServidor, implemente o comportamento do botão "Enviar paralelo". Esse botão deve fazer a mesma coisa que o botão "Enviar sem erros", porém de forma paralelizada, com objetivo de ganhar performance. Pense em uma solução que irá funcionar mesmo que a constante QTD_ARQUIVOS_ENVIAR tivesse valores mais altos, como mil ou dez mil

Implementação 4: para demonstrar seu domínio em programação com Threads no Delphi, crie um formulário com o nome da unit "Threads.pas" e nome do form "fThreads", adicione ao formulário 2 edits, um botão, uma barra de progresso e um memo (TMemo). O primeiro edit será utilizado para especificar o número de threads a serem criadas, o segundo edit para informar um valor de tempo em milissegundos máximo de espera entre cada iteração dos threads. Estes threads irão realizar um laço de 0 até 100, onde a cada iteração do laço elas deverão aguardar um tempo randômico em milissegundos, sendo o valor máximo determinado pelo usuário considerando o dado inserido no formulário. Ao iniciar o processamento um thread deve inserir no memo (TMemo) seu thread-id e o texto "Iniciando processamento" (Ex. 1543 - Iniciando processamento) e ao término do mesmo seu thread-id e o texto "Processamento finalizado" (Ex. 1543 - Processamento finalizado), os textos respectivos devem ser disparados dentro do thread em si. A cada iteração do laço, a thread deverá incrementar o valor do contador de iterações do loop na barra de progresso disponível no formulário. Todos os threads compartilharão a mesma barra de progresso, sendo então o valor mínimo da barra de progresso 0 e seu valor máximo o número de

threads * vezes número de iterações (Em nosso exemplo de 0 a 100 ocorrem 101 iterações). Importante: Preocupe-se com o fechamento do formulário, o mesmo deve solicitar que as threads finalizem "gentilmente" ou esperar o término do processamento. Dica: utilize o procedimento Sleep(Milisegundos: Integer) para fazer a espera entre cada iteração do loop das threads e a função Random para garantir um valor aleatório de espera.

Correções

Corrija cada defeito descrito abaixo. Na descrição do defeito terá o problema e o objetivo da correção. Para cada defeito, preencher o campo "Solução" detalhando tecnicamente a causa do problema e a solução dada.

Defeito 1: na tela DatasetLoop, ao clicar no botão "Deletar pares" não deleta todos os pares do dataset. Objetivo: que todos os números pares sejam deletados

Solução:

Defeito 2: na tela ClienteServidor, ocorre erro "Out of Memory" ao clicar no botão "Enviar sem erros". Objetivo: que não ocorra erro por falta de memória, e que todos os arquivos sejam enviados para a pasta Servidor normalmente.

Solução:

Defeito 3: na tela ClienteServidor, ao clicar no botão "Enviar com erros", os arquivos enviados anteriormente não são apagados da pasta Servidor. Objetivo: quando ocorrer erro na operação, que é o caso que esse botão simula, os arquivos copiados anteriormente devem ser apagados, simulando um "rollback". Ou seja, no fim da operação, os arquivos devem continuar na pasta apenas se não ocorreu erro na operação. obs: não é para ser corrigido o erro que ocorre ao clicar nesse botão, visto que ele serve justamente para simular um erro.

Solução: