AVALIAÇÃO PROGRAMADOR .NET + SQL Server

Autor: Pedro Canadas Atualizado em: 13/10/2022

.NET

Você poderá escolher a linguagem C# ou VB.NET, a aplicação, Windows Forms, Web Forms ou MVC.

Você deverá criar uma única aplicação, para responder todas as questões. Você deverá zipar esse ÚNICO projeto com todos as questões feitas e enviar como resultado.

Você pode consultar na internet as coisas que você não conhece.

Veja como esperamos o resultado do teste:
https://github.com/phenriquec/publicFiles/blob/main/RH_Teste/.NET.rar

1. Faça a aplicação permitir a digitação de números e mostre esses números em tela de forma ordenada.
2. Agora grave os números visualizados cada 1 em uma linha em um arquivo texto na pasta raiz da aplicação de nome numeros_ordenar.txt.
3. Crie uma lista contendo 100 itens de uma classe de nome clsTeste com as propriedades codigo como número e descricao como texto, os objetos deverão ser criados com a propriedade codigo com números sequenciais (ex: 1,2,3,4,5) e a descricao como a data e hora atual (ex: 2022/10/13 08:50:22.123)
4. Grave a lista do item 3, em um arquivo de nome data.json na pasta raiz da aplicação.
5. Crie um Grid, leia o arquivo data.json que foi gravado, e mostre os dados no Grid criado.
6. Consuma o webservice dos correios passando um CEP qualquer e mostre em tela o endereço que o mesmo retornar.

   Endereço: https://apps.correios.com.br/SigepMasterJPA/AtendeClienteService/AtendeCliente?wsdl

   Método: consultaCEP
7. Consuma a API para buscar a lista de bancos brasileiros

   Documentação: https://brasilapi.com.br/docs#tag/BANKS/paths/~1banks~1v1/get

   URL: https://brasilapi.com.br/api/banks/v1

   Mostre os dados de retorno da API em um Grid.
8. Pela aplicação faça o download da imagem https://redeservice.com.br/wp-content/uploads/2020/07/redeservice-logo.png, colocar na pasta do sistema, e criar alguma função para ler essa imagem e mostrar em tela no formato base64.

AVALIAÇÃO PROGRAMADOR .NET + SQL Server

SQL SERVER

Você deverá criar um único arquivo para responder todas as questões, no padrão que esperamos.

Nesse arquivo você vai encontrar o script para criação das tabelas e também como esperamos receber a resposta:
https://github.com/phenriquec/publicFiles/blob/main/RH_Teste/SQL%20Server.rar

1. Criar um select para listar todas as descrições (campo [descrição]) da tabela "Tabela" que estejam duplicadas, mostrando no select a descrição e a quantidade de registros duplicados

2. Criar um select para listar os campos da tabela "Tabela" de todos os registros que estão na tabela "Tabela", porém não estão na tabela "Tabela_esp" utilizando o campo [código] como chave

3. Criar um update para atualizar o campo OBS da tabela "Tabela" com o conteúdo "S" se o registro existir na tabela "Tabela_esp"

4. Criar uma procedure de nome tst_sp_descricoes_duplicadas como dbo, que deverá receber um parâmetro chamado @descricao. Essa procedure deverá incluir em uma tabela temporária de nome Registros_duplicados todos os registros duplicados de uma determinada descrição e logo em seguida visualizá-los

5. Criar uma trigger na tabela "Tabela" que inclua a situação anterior na tabela "Tabela_hist" toda vez que a tabela "Tabela" sofrer uma alteração ou exclusão

6. Criar uma function de nome tst_fn_existecodigo que receba um parâmetro chamado @codigo. Essa função deverá ter um retorno do tipo bit e retornar 1 quando existir o código na tabela "Tabela" e 0 quando não existir

7. Criar um script que crie uma tabela idêntica a tabela "Tabela" de nome "Tabela_espelho" caso ela não exista. Inclua nessa tabela o conteúdo da tabela "Tabela" de 1 em 1 registro (utilizando cursor) e executando um commit a cada 100 registros.

8. Criar uma procedure de nome tst_sp_importa_arquivo como dbo, que deverá receber um parâmetro chamado @arquivo. Essa procedure deverá incluir na tabela ARQUIVO_TESTE o conteúdo do arquivo "arquivo_teste.txt", localizado na sua pasta de teste, pelo comando bulk insert e logo depois visualizá-lo