Documentação do Projeto: TaskService - Conexão com Banco de Dados

- **Objetivo do Projeto**

O objetivo deste projeto é desenvolver um sistema de gestão de tarefas que permita a criação, remoção, atualização e listagem de tarefas com integração a um banco de dados MySQL. O sistema foi projetado para ser utilizado por indivíduos ou pequenas equipes que desejam organizar suas tarefas, acompanhar prazos e modificar automaticamente o status de tarefas expiradas.

- **Linguagem Utilizada e Fundamentos**

A linguagem utilizada para o desenvolvimento é **C#**, uma linguagem moderna e orientada a objetos. C# é fortemente tipada e amplamente usada no desenvolvimento de aplicações desktop, web e móveis devido à sua robustez e flexibilidade. O projeto também faz uso do **ADO.NET** para conectar-se ao banco de dados MySQL e realizar operações de CRUD (Create, Read, Update, Delete).

Bibliotecas Utilizadas

As seguintes bibliotecas são utilizadas neste projeto:

- **System**: Biblioteca principal que oferece classes básicas essenciais para a aplicação.
- **MySql.Data.MySqlClient**: Permite a conexão e execução de comandos no banco de dados MySQL.

Funcionalidades e Métodos

‘InsertData_s’ | ‘InsertData_exe’ - (Inserção de Tarefas)

Adiciona uma nova tarefa ao banco de dados.

- **Parâmetros**: Nome da tarefa, Descrição, Data de vencimento, Prioridade (ALTA, MÉDIA, BAIXA), Status (Pendente por padrão).
- **Lógica**: O usuário fornece os detalhes da tarefa. O sistema valida os dados e insere a tarefa no banco de dados.

‘DeleteData_s’ - ‘DeleteData_exe’ - (Remoção indivídual de Tarefas)

Remove uma tarefa especificada pelo nome fornecido pelo usuário.

- **Parâmetros**: Nome da tarefa
- **Lógica**: O sistema solicita o nome da tarefa que deseja remover e, em seguida, executa o comando de exclusão no banco de dados.

‘DeleteAll_s’ - ‘DeleteAll_exe’ - (Remoção total das tarefas)

Realiza a remoção de todas as tarefas presentes no banco de dados.

- **Parâmetros**: Nenhum
- **Lógica**: O sistema solicita a senha cadastrada para confirmar a ação e realizar a remoção de todas as tarefas vinculadas ao banco.

‘UpdateData_s’ | ‘UpdateData_exe’ - (Atualização de Tarefas)

Atualiza os atributos de uma tarefa existente no banco de dados.

- **Parâmetros**: Nome da tarefa, novo valor para Nome, Descrição, Data de vencimento, Prioridade ou Status.
- **Lógica**: O sistema solicita o nome da tarefa e qual atributo será alterado. Em seguida, o banco de dados é atualizado com o novo valor.

‘QueryData_s’ | ‘QueryData_exe’ - (Consulta de Tarefas)

Lista todas as tarefas armazenadas no banco de dados, mostrando seus detalhes.

- **Parâmetros**: Nome da tarefa.
- **Lógica**: O sistema recupera e exibe todas as tarefas, independentemente de seu status, incluindo informações como nome, descrição, data de vencimento, prioridade e status atual.

‘CheckTask’- (Verificação automática do status da tarefa)

Verifica a data vinculada a tarefas com status “Pendente”.

- **Parâmetros:** Object da classe atributos ‘aux_date’ para instanciação da chamada da variável ‘Data_Atual’.
- **Lógica:** Na inicialização do programa, ocorre a verificação automática de tarefas que possuem status do tipo “Pendente”. Ocorre uma comparação direta entre a data

dessas tarefas e a data atual, se essas tarefas não possuírem data igual, ou superior a atual, é feita a mudança de status da tarefa para “Expirada”.

Estrutura do Código

Interface

- **ITask_Storage:** Interface responsável pela parte visual dos métodos/interação com o usuário e coleta de dados relativas as tarefas. Tendo por fim o apontamento para os métodos da interface ‘ITask_Execution’.
- **ITask_Execution**: Define os métodos básicos para gerenciar tarefas, como adicionar, remover, atualizar e consultar.gnm

Classes

- **Task_Attributes**: Define os atributos de cada tarefa (Nome, Descrição, Data de vencimento, Prioridade, Status) e utiliza modificadores de acesso protected/public.
- **Connect_db**: Gerencia a conexão com o banco de dados MySQL e implementa a string de conexão.
- **TaskService - Armazenamento de dados (pasta de classes)**: Incremento dos métodos da interface ‘ITask_Storage’, onde é executado uma interação com o usuário, coleta de dados e redirecionamento para os métodos da interface ‘ITask_Execution’.
- **TasKService - Execução dos comandos DB (pasta de classes):** Conforme apontamento dos métodos da interface ‘ITask_Storage’, é feito a execução dos comandos (CRUD), onde ocorre a lógica de gerenciamento de tarefas e execução de operações no banco de dados com base nos métodos definidos.
- **Static_Count_Task**: Função responsável em realizar a contagem de tarefas presentes no banco de dados, por meio de uma operação ADO.NET e retorno de um número inteiro com o resultado.
- **CheckTask:** Incremento de uma lógica concisa que realiza uma operação de consulta de tarefas com status pendente, convertendo o status para expirado caso a data vinculada a tarefa seja inferior a data atual.
- **Console_Main**: Contém o método Main que controla o fluxo da aplicação e gerencia as interações com o usuário.

Conclusão

Esta documentação apresenta uma visão geral do sistema **TaskService - Conexão com Banco de Dados**, explicando seu objetivo, linguagem utilizada, bibliotecas necessárias e funcionalidades principais. A estrutura modular do código facilita o gerenciamento de tarefas e a interação com um banco de dados relacional, tornando o sistema eficiente e escalável para usuários e equipes.