**Códigos do dia 22/03/2024 – Solução Chamados Técnicos**

Atenção : Este documento será atualizado durante as aulas, o documento é composto por um objeto de texto, onde você pode abrir e ter acesso ao código completo e a explicação do código abaixo.

**Código do arquivo Chamados.cs**

****

**Descrição do Código:**

Este código define uma classe chamada **Chamado** dentro de um namespace chamado **Data**. Uma classe é como um molde que descreve as propriedades e comportamentos de um objeto. Neste caso, a classe **Chamado** é usada para representar as informações e ações relacionadas a um chamado de suporte técnico ou serviço.

Propriedades da Classe **Chamado**:

1. **CodigoChamado** (tipo **int**):
   * Descrição: Esta propriedade armazena o código único para cada chamado. O tipo **int** indica que o valor deve ser um número inteiro.
   * Exemplo: Se um chamado tem o código 123, isso significa que ele pode ser identificado unicamente por este número dentro do sistema.
2. **DataSolicitacao** (tipo **DateTime**):
   * Descrição: Armazena a data e hora em que o chamado foi feito. **DateTime** é um tipo que combina data e hora.
   * Exemplo: Se um chamado foi feito em 22 de março de 2024, às 14:00, essa informação seria registrada aqui.
3. **Ocorrencia** (tipo **string**):
   * Descrição: Uma descrição textual do que aconteceu para que o chamado fosse aberto. O tipo **string** significa que essa propriedade pode conter uma sequência de caracteres, ou seja, texto.
   * Exemplo: "A impressora do setor de vendas está sem funcionar."
4. **Problema** (tipo **string**):
   * Descrição: Detalhes específicos do problema reportado no chamado.
   * Exemplo: "A impressora não liga mesmo após verificar a conexão com a energia."
5. **Concluido** (tipo **bool**):
   * Descrição: Indica se o chamado foi concluído. O tipo **bool** é um valor booleano, o que significa que só pode ser **true** (verdadeiro) ou **false** (falso).
   * Exemplo: Se o chamado foi resolvido, esta propriedade estaria definida como **true**.
6. **fk\_Cliente\_CodigoCliente** (tipo **int**):
   * Descrição: Uma chave estrangeira que relaciona o chamado a um cliente específico através de um código único de cliente.
   * Exemplo: Se o cliente com código 456 fez um chamado, este número estaria associado ao chamado para indicar quem é o cliente.
7. **fk\_Tecnicos\_CodigoTecnico** (tipo **int**):
   * Descrição: Semelhante à propriedade anterior, mas neste caso, relaciona o chamado a um técnico específico que irá ou já atendeu o chamado.
   * Exemplo: Se o técnico responsável pelo chamado tem o código 789, este número estaria aqui para identificar o técnico.

**Como a Classe Funciona:**

* Cada instância (ou objeto) criado a partir da classe **Chamado** pode armazenar informações sobre um chamado específico, como a data em que foi solicitado, o problema reportado, e se já foi resolvido.
* As propriedades **fk\_Cliente\_CodigoCliente** e **fk\_Tecnicos\_CodigoTecnico** são usadas para associar o chamado a um cliente e a um técnico específicos, respectivamente, usando seus códigos únicos. Isso ajuda a manter o registro de quem fez o chamado e quem é responsável por atendê-lo.

**Código do arquivo Cliente.cs**



**Explicação Detalhada:**

* **CodigoCliente:** Serve como identificador único para cada cliente, facilitando operações como busca, atualização e exclusão de registros de clientes no sistema.
* **Nome:** O nome é fundamental para a identificação pessoal do cliente, usado em comunicações e registros.
* **Profissao:** Conhecer a profissão do cliente pode auxiliar na personalização dos serviços oferecidos, adaptando-os às necessidades específicas de sua área de atuação.
* **Setor:** Esta informação ajuda a contextualizar o cliente dentro de sua organização, podendo influenciar no tipo de serviço ou produto que será mais adequado para ele.
* **Obs:** O campo de observações é flexível e permite o armazenamento de qualquer informação adicional que possa ser relevante para a gestão do relacionamento com o cliente, como preferências particulares, datas importantes (aniversários, por exemplo), entre outros.

**Código do arquivo Cliente.cs**



**Pontos Importantes do Código:**

* **ADO.NET e SqlConnection:** O código utiliza ADO.NET, um modelo de acesso a dados do .NET Framework, e a classe **SqlConnection** para conectar-se ao SQL Server.
* **Construtor ClienteDao:** Recebe uma string de conexão quando uma instância da classe **ClienteDao** é criada, permitindo que essa conexão seja reutilizada para operações de banco de dados.
* **Método IncluiCliente:** Responsável por inserir um novo registro de cliente no banco de dados. Ele usa parâmetros (via **Parameters.AddWithValue**) para passar os valores para a instrução SQL, uma prática recomendada para prevenir SQL Injection, que é uma vulnerabilidade de segurança comum em aplicações web.
* **Gerenciamento de Conexão:** O uso do bloco **using** garante que a conexão com o banco de dados seja fechada automaticamente quando a operação for concluída, liberando recursos de maneira eficiente.
* **Tratamento de Exceções:** O método tenta executar a operação de inclusão e captura exceções que podem ocorrer durante o processo, lançando uma exceção com uma mensagem de erro mais clara em caso de falha.

**Código do arquivo Tecnico.cs**



**Explicação do Código:**

* **Classe Tecnico:** Serve como uma representação de um técnico ou especialista, incluindo todas as informações necessárias para gerenciar sua identidade e habilidades dentro de um sistema.
* **CodigoTecnico:** É a identificação única de cada técnico, o que permite o gerenciamento eficaz dos registros de técnicos no sistema.
* **Nome, Especialidade, e Email:** São informações fundamentais para a comunicação e atribuição adequada de tarefas baseadas em habilidades específicas.
* **Senha:** Permite o acesso seguro ao sistema, garantindo que apenas usuários autorizados possam entrar e realizar operações.
* **Obs:** Oferece flexibilidade para registrar qualquer informação adicional relevante que possa ajudar na gestão do técnico.

**Código do arquivo frmAdicionarCliente.cs**

**Formulário VISUAL**

****

**Pontos Chave do Código:**

* **Inicialização e Eventos:** O formulário é inicializado pelo método **InitializeComponent()**, que é um método gerado automaticamente pelo Windows Forms para configurar a UI baseado no designer. Os eventos **btnLimpar\_Click** e **btnSalvar\_Click** são vinculados a botões específicos no formulário para limpar campos e salvar dados, respectivamente.
* **Validação de Dados:** Antes de tentar salvar um novo cliente, o código verifica se todos os campos necessários estão preenchidos. Isso ajuda a evitar a inserção de dados incompletos ou inválidos no banco de dados.
* **Manipulação de Exceções:** O uso de um bloco **try-catch** no método **btnSalvar\_Click** é uma prática recomendada para capturar e tratar erros que podem ocorrer durante a execução do código, especialmente quando envolve operações de banco de dados que podem falhar por várias razões.
* **Feedback ao Usuário:** O código utiliza caixas de mensagem (**MessageBox.Show**) para informar o usuário sobre o sucesso da operação ou detalhar falhas que precisam ser corrigidas.