PITAYA COMPANY -

Caio, João C., João V., Vinicius, Fabio, Hudson

PROJETO BANCO DE DADOS NÃO RELACIONAL

Integração utilizando CRUD e MongoDB

Professor: Gilberto de Oliveira Antonio

São Paulo-SP

2024

**RESUMO**

**Este projeto consiste na produção de um sistema utilizando CRUD e MongoDB na qual o mesmo seja composto por uma tela de login (login, senha e botões ‘’OK’’ e ‘’Limpar’’), tela de listagem de clientes (Campo de pesquisa, grid para listagem de clientes com botões de editar e excluir cada registro e uma botão de Criar) e uma tela de cadastro (Nome, Idade, Telefone, Opção Fumante e Sexo) e ao final sendo exportado para o GitHub para análise do professor e criação de um portifólio de projetos para os alunos.**

**1 INTRODUÇÃO**

O sistema é composto pelos sistemas CRUD, MongoDB, WinForms para exibição e a linguagem de programação C#. A escolha específica do MongoDB foi dada pelo fato de ser um banco de dados não relacional grátis e ‘’simples’’ de utilizar além de ser a matéria e tema estudado pelos alunos no momento. O C# foi escolhido pelo grupo por ser uma linguagem que tinham mais facilidade para trabalhar, porém, também sendo possível a produção utilizando Java, e o CRUD fazendo a ligação do banco de dados entre tudo isso pois compõe os conceitos fundamentais do desenvolvimento pelo fato de realizar as operações básicas de armazenamento consistente, como ‘’Create, Read, Update e Delete’’.

**2** **PRODUÇÃO DO SISTEMA**

A produção do sistema foi divido por etapas, da qual se inicia com o design em si, montando onde e como ficará cada detalhe, botão, texto etc. Logo em seguida a ligação com o CRUD para a execução das operações e pôr fim a conexão com o banco de dados não relacional para a persistência de dados.

**3 GITHUB**

O trabalho se encontra na plataforma GitHub, um local onde pessoas do mundo inteiro compartilham e guardam seus projetos, é possível inclusive analisar o nosso pelo link: <https://github.com/Pitaya-Company>.

**4 MONGODB**

O MongoDB funciona com uma conexão por meio de CRUD no qual uma string connection requisita a conexão do banco de dados para o sistema atual, criando um cliente e pesquisando as informações deste cliente por uma leitura privada de um ‘’\_database = client.GetDatabase("PitayaCompanyDb");’’ e logo após segue com um GET dos usuários e clientes cadastrados.

Dentro do MongoDB em si a persistência de dados é feita por meio de linhas de códigos em JSON ou BSON da qual não requer um esquema fixo, sendo completamente diferente de um banco relacional SQL, por exemplo, e o que por ser feito de modo horizontal no MongoDB facilita a escalabilidade do banco de dados.

**5 TELAS, DESIGNS E CONSTRUÇÕES**

A primeira tela é a tela de login, como comentado no resumo ela contém o login, senha e botões de ‘’OK’’ e ‘’Limpar’’.

As informações de login são procuradas no banco de dados, e caso encontradas e coincidirem o usuário é liberado.

Essa análise de informações é feita por meio de gets e sets da qual no banco conferem se as strings no forms são as mesmas colocadas pelo usuário.

Caso não coincidirem há uma MessageBox informando que o e-mail ou senha estão incorretos.

**6 TELA DE CADASTRO**

Outra tela que encontramos é a de cadastro, na qual temos também outro forms para a pessoa ir respondendo conforme o que for requisitado, como nome, idade, telefone etc. Na qual conforme todas forem respondidas ele adiciona dentro da variável cliente um novo cliente, e assim envia as informações ao banco de dados e exibe uma mensagem de cliente criado com sucesso. Em caso de erro em algum campo também há uma MessageBox de erro de cadastro.

**7 TELA DE LISTAGEM**

A tela de listagem depende de um elemento do Windows Forms chamado DataGrid, esse elemento é uma tabela com propriedades de automatização, ela é gerada a partir da coleção de clientes, fornecida pelo ClienteService, sem necessidade de definir as colunas diretamente.

TelaListagem possui métodos separados para controlar a visualização do Grid, em especial o método AtualizarDataGrid é constantemente chamado para redefinir a coleção e povoar a tabela. Um método captura o instante em que a caixa de busca sofre mudanças e chama o método GetClientesAsync em ClienteService com o parâmetro de nome, alterando a coleção de clientes, chamando AtualizarDataGrid novamente e assim possibilitando a pesquisa em tempo real. Duas colunas são adicionadas manualmente com o conteúdo de botões, são chamadas de colunas Ação e nelas estão Editar e Excluir, toda linha do DataGrid gera automaticamente esses dois botões.

Ao pressionar Excluir, uma janela se abre para confirmar a exclusão do objeto Cliente na linha, o método chama DeleteClienteAsync em ClienteService. Ao pressionar em Editar, um método prepara o processo de edição. O processo de edição começa criando um backup dos dados na linha selecionada e tornando a linha “ReadOnly=false”, possibilitando editar o texto dos campos, e alterando os botões de Ação para “Ok” e “Cancelar”, ao pressionar Cancelar, o Backup dos dados é estabelecido novamente na tabela, ao pressionar Ok, as alterações feitas na linha são persistidas no banco de dados.

Além disso, há dois botões nas extremidades da caixa de busca para Sair — fecha TelaListagem, retornando para TelaLogin — e Cadastrar — esconde TelaListagem e abre TelaCadastro.

**8 COMO UTILIZAR O SISTEMA**

O passo a passo consiste em:

* Crie uma variável de ambiente com uma connection string válida para um cluster mongodb/instância local, a variável deve ter como nome PITAYACONNECTIONSTRING;
* Baixe o executável abaixo;
* Execute.

Você pode encontrar o link para instalação e código logo abaixo:

<https://github.com/Pitaya-Company/pitaya-crud/releases/tag/pitaya>