

Relatório Trabalho Programação orientação objeto

Projeto Gestao de Energia

Ana Flavia de O. Costa, Derick L. A. Rodrigues ,Matheus Antonio V. Freitas, Sabrina Luiza M. da

¹Sistemas de Informação - Pontifícia Universidade Católica de Minas Gerais (PUC- Minas)
Minas Gerais – Brasil

Abstract. *The system developed in Windows Forms and C for calculating energy consumption and amount to be paid was designed following the Model-View-Presenter (MVP) architectural pattern. This pattern separates presentation logic from business logic, providing more modular, testable, and maintainable code.*

*The application uses SQL Server for data storage and manipulation, ensuring the persistence of information related to energy consumption. The project was implemented with the aim of offering an intuitive interface for users, allowing quick and accurate calculations of electricity consumption and the corresponding amount to be paid. **Keywords:** Windows Forms. SQL Server. C. Database. MPV. Desenvolvimento.*

Resumo. *O sistema desenvolvido em Windows Forms e C para cálculo de consumo de energia e valor a ser pago foi projetado seguindo o padrão arquitetural Model-View-Presenter (MVP). Este padrão separa a lógica de apresentação da lógica de negócios, proporcionando um código mais modular, testável e de fácil manutenção.*

A aplicação utiliza o SQL Server para armazenamento e manipulação dos dados, garantindo a persistência das informações relacionadas ao consumo de energia. O projeto foi implementado com o objetivo de oferecer uma interface intuitiva para os usuários, permitindo cálculos rápidos e precisos do consumo de energia elétrica e o valor correspondente a ser pago.

Palavras-chave: *Windows Forms. SQL Server. C. Banco de dados. MPV. Desenvolvimento.*

1. Arquitetura

O padrão arquitetural Model-View-Presenter (MVP) é uma variação do padrão Model-View-Controller (MVC) e é amplamente utilizado no desenvolvimento de interfaces gráficas. Ele promove a separação entre a lógica de apresentação e a lógica de negócios, resultando em um código mais modular e testável. No MVP, temos três componentes principais:

Model: Representa os dados e a lógica de negócios da aplicação. Em um projeto C, é composto por classes que encapsulam dados e operações sobre esses dados. No diagrama, exemplos de classes de Model incluem PessoaModel, PessoaJuridicaModel, PessoaFisicaModel, ContaDeEnergiaModel, entre outros.

View: Responsável pela apresentação e interação com o usuário. A View contém apenas operações de interface do usuário (UI) e não possui lógica de negócios. No diagrama, as classes `PessoaJuridicaRepository`, `PessoaFisicaRepository`, `LeituristaRepository`, e `AdministradorRepository` podem ser vistas como componentes que interagem com a interface de usuário, embora tecnicamente sejam partes do modelo, mostrando como os dados são acessados.

Presenter: Atua como intermediário entre o Model e a View. Ele recebe as entradas do usuário através da View, processa essas entradas (se necessário, interagindo com o Model), e atualiza a View com os resultados. No diagrama, `PresenterCliente` e `ValidacaoContaDeEnergia` podem ser considerados componentes que exercem a função de Presenter, gerenciando a lógica de apresentação e a interação com os modelos de dados.

A adoção do padrão MVP promove a separação de responsabilidades dentro da aplicação, tornando o código mais organizado, testável e fácil de manter. No entanto, a complexidade do MVP pode aumentar em projetos maiores, sendo necessário um cuidado adicional no design e na implementação. Em sistemas complexos como o demonstrado no diagrama, uma boa prática é garantir que cada componente do Presenter esteja bem definido e limitado a um conjunto específico de responsabilidades, para evitar sobrecargas e dependências desnecessárias.

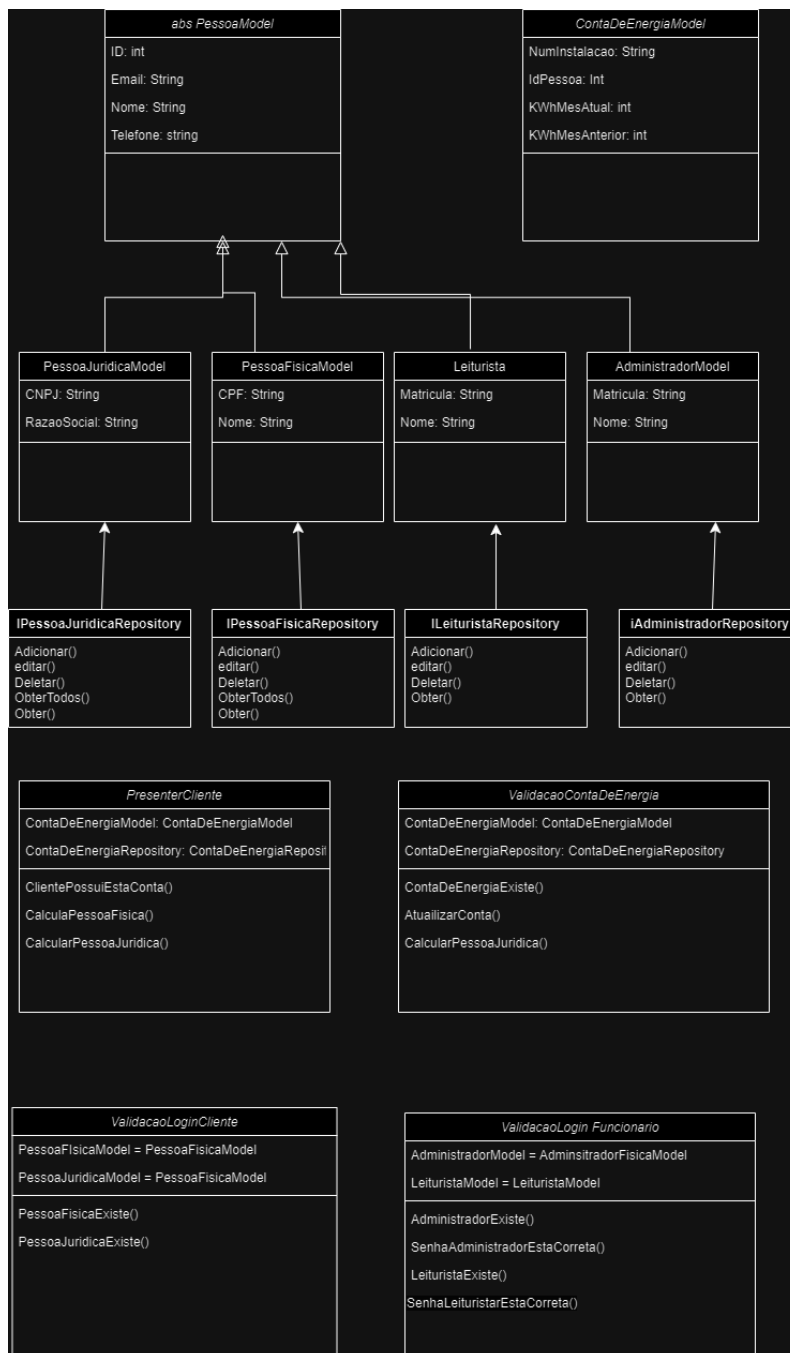


Figura 1. Diagrama

2. Funcionamento e passo a passo de como utilizar

2.1. Primeiros Passos

Para utilizar o programa vai ser necessário abrir seu SQL server, rodar o script de banco de dados e também dentro do código alterando o conector.

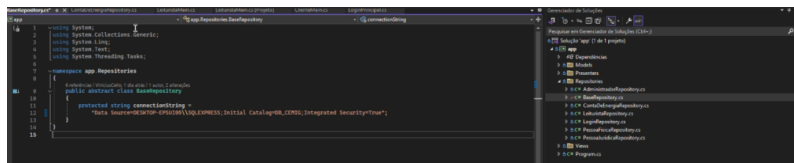


Figura 2. tela de como conectar

No primeiro acesso, você criará seu usuário administrador dentro do próprio banco de dados, com ele poderá utilizar o sistema.

2.2. Administrador

Realiza o cadastro de usuários tanto de Pessoa Física quanto pessoa Jurídica.

 The image shows a screenshot of a web application window titled 'AdministradorMain'. The window has a tabbed interface with 'Clientes' and 'Cadastrar Cliente'. The 'Cadastrar Cliente' tab is active, displaying a form titled 'CADASTRAMENTO'. The form has a light green background and contains several input fields and a dropdown menu. The fields are: 'Tipo' (a dropdown menu with 'Pessoa Física' and 'Pessoa Jurídica' options), 'Documento' (a text input field), 'Email' (a text input field), 'Telefone' (a text input field), 'DDD' (a small text input field), 'Nome/Razão Social' (a text input field), and 'Número de registro' (a text input field). At the bottom of the form is a green button labeled 'CADASTRAR'.

Figura 3. Tela de cadastro

Aqui vemos como é o funcionamento da tela do cadastramento

CADASTRAMENTO

Tipo: Pessoa Fisica

CPF: 52123123131

Email: derick@gmail.com

Telefone: DDD 31, 918398123

Nome: Derick Lucas

Número de registro: 2024423323424

CADASTRAR

Sucesso!
Adicionado com sucesso!
OK

Figura 4. Tela de cadastro preenchida

Listagem de clientes, bem como realizar associação de contas.

LISTA DE CLIENTES

Pesquisar

	CPF	Nome	Id	Email	Telefone
▶	14342046695	Vinicius Pessoa ...	2009	vinicius@gmail...	31986825699
	14387659687	Joao PF	2008	joao@pessoafis...	31991456798
	22222222222	22222222222	2014	22222222@22....	22222222222

Associar conta

Pesquisar

	CNPJ	RazaoSocial	Id	Email	Telefone
▶	12312312322222	1231231 SA	2013	12312312@GM...	33231233333
	12333123233333	SAF SA	2012	132131@CNPJ....	31987654389

Remover cliente

Figura 5. Listas de clientes

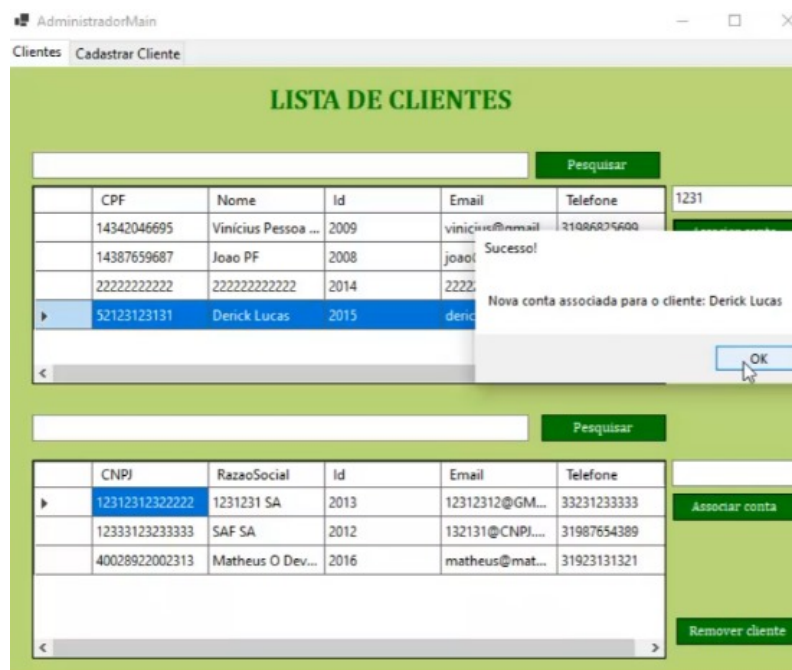


Figura 6. Associacao

2.3. Leiturista

Tem acesso a apenas uma unica tela, onde a mesma é para upar registro de contas, que no momento que voce upa uma nova medida, ela vira a atual no exato momento

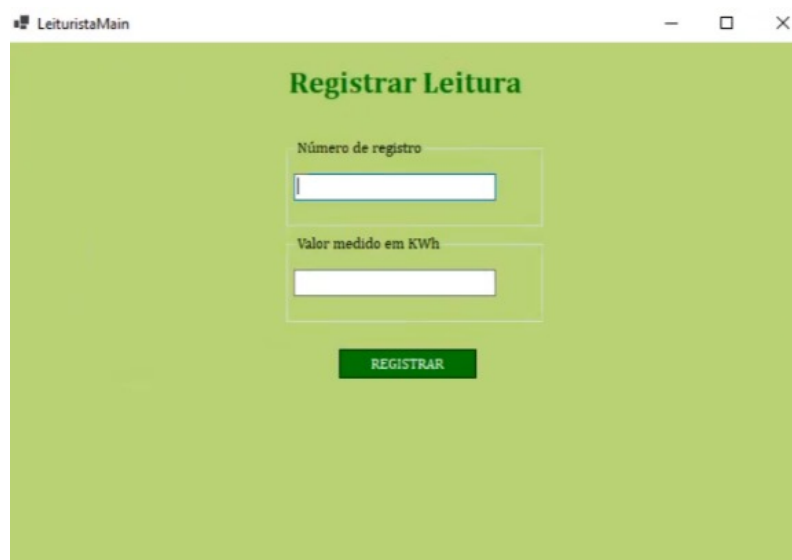


Figura 7. Tela de leiturista

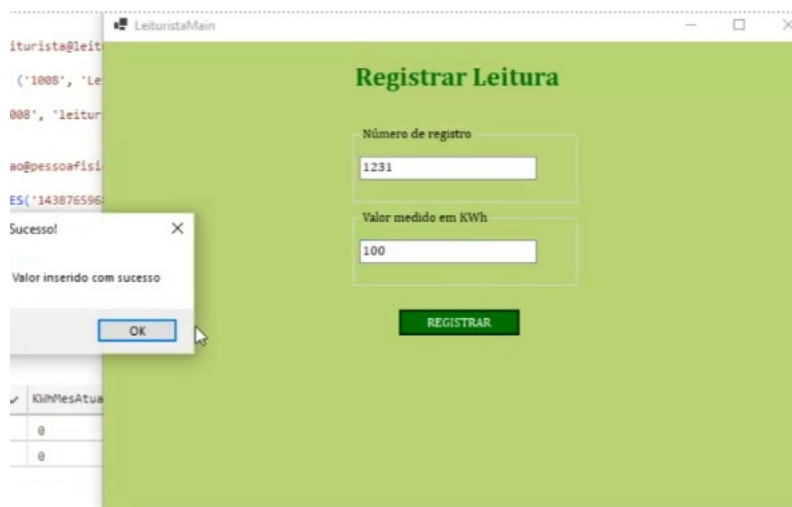


Figura 8. Tela Preenchida

2.4. Usuário

Para o usuário temos a tela de login, onde será possível acessar sua conta através da opção Pessoa Física ou Jurídica e preenchimento de seu CPF ou CNPJ.

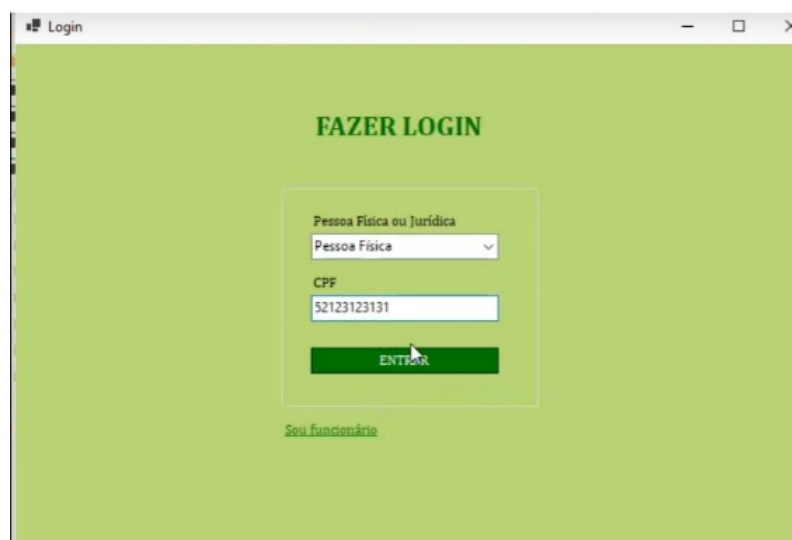


Figura 9. Tela de Login usuario



Figura 10. Tela de valor conta

3. Resultados

Com a aplicação em C# Windows Forms desenvolvida, foi obtido o resultado que era se esperado da tecnologia. Foram alcançadas as informações preenchidas pelo os administradores do sistema, como leiturista. De fácil utilização para o usuário final, com salvamento de informações e gerenciamento de dados, utilizando tecnologias e métodos de orientação objeto e estrutura de dados.

4. Considerações finais

O desenvolvimento do projeto permitiu ao grupo adquirir novos conhecimentos, como de arquiteturas e habilidades na linguagem de programação C, além de ter proporcionado uma experiência na prática de criação de um sistema com a arquitetura MVP e orientação objeto.

5. References

Documentação Windows Forms: <https://learn.microsoft.com/pt-br/dotnet/desktop/winforms/?view=netdesktop-7.0>
Documentação do C#: <https://learn.microsoft.com/pt-br/dotnet/csharp/>
Cursos: <https://blog.balta.io/>
Cursos: <https://rjcodeadvance.com/>