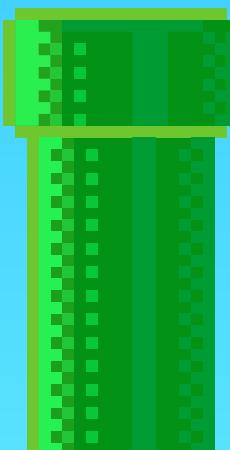
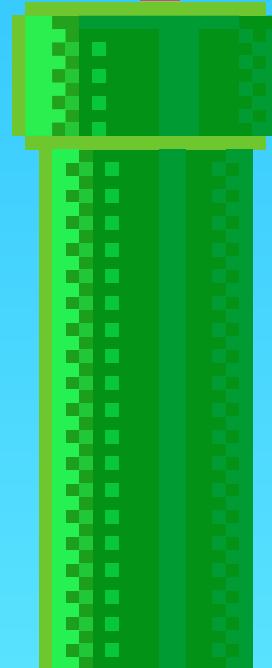
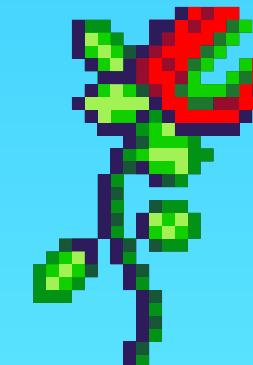
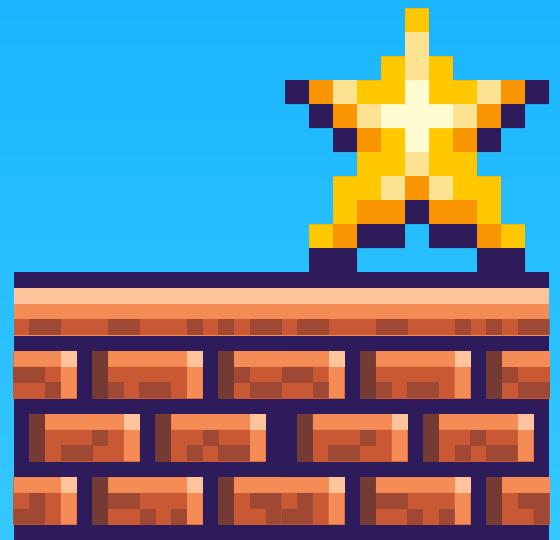


STOCK GAME

LET'S PLAY!



SELEÇÃO DE TELA

There are 6 different types of games!

Tela Principal

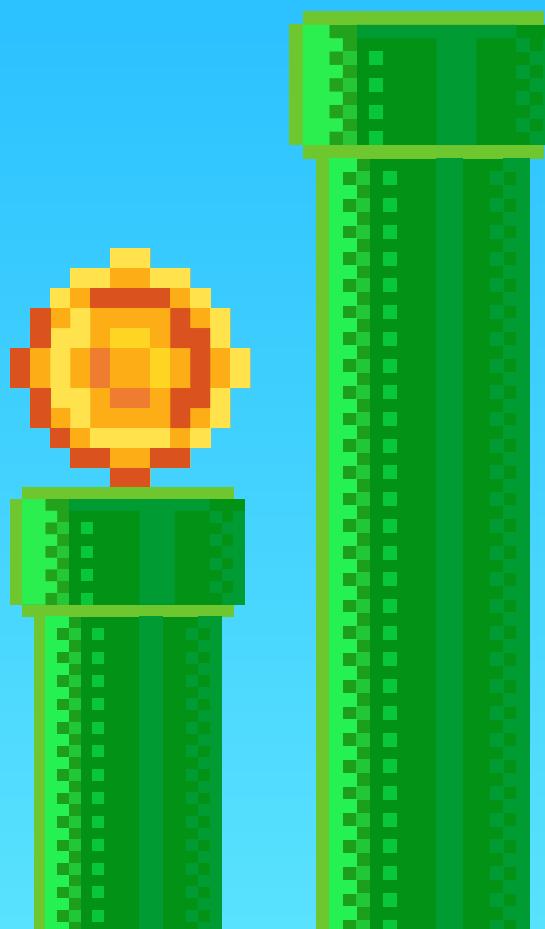
Tela login
usuário

Tela
Categoria

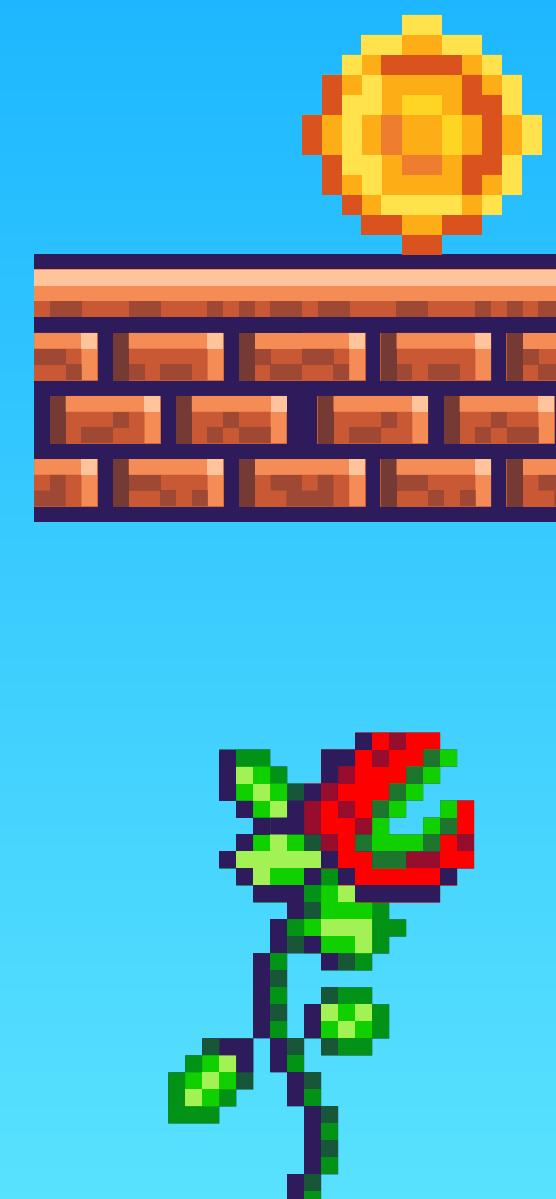
Tela
gestão de
clientes

Tela
colaborador

Tela
Produtos



Complete the word



SELEÇÃO DE TELA

There are 6 different types of games!

Tela Principal

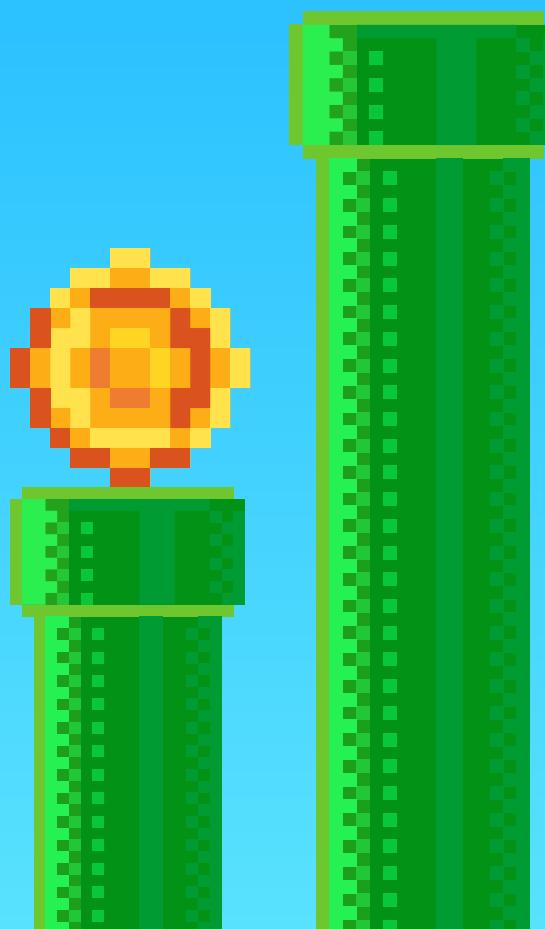
Tela login
usuário

Tela
Categoria

Tela
gestão de
clientes

Tela
colaborador

Tela
Produtos



Complete the word

Login

Faça o Seu Login Aqui

Entrar

Recuperar Senha

Entrar

SELEÇÃO DE TELA

There are 6 different types of games!

Tela Principal

Tela login
usuário

Tela
Categoria

Tela
cliente

Tela
colaborador

Tela
Produtos



Answer the question

Categorias e Subcategorias

Nome de Categoria Ativa [+ Cadastrar Categoria](#)

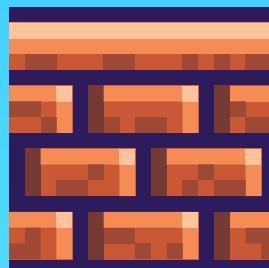
Nome de SubCategoria Smartphones Ativa [+ Cadastrar SubCategoria](#)

Categorias principais dos produtos [Recarregar](#)

ID	Nome	Status	Data Cadastrada	Data Alteração	Editar	Excluir
1	Smartphones	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Edit	Delete
2	Roupas	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Edit	Delete
3	Alimentos Não Perecíveis	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Edit	Delete
4	Móveis de Sala	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Edit	Delete
5	Equipamentos Esportivos	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Edit	Delete

Subcategorias dos produtos [Recarregar](#)

ID	Nome	Situação	Categoria	Editar	Excluir
1	Eletrônicos	<input checked="" type="checkbox"/>	0	Edit	Delete
2	Moda	<input checked="" type="checkbox"/>	0	Edit	Delete
3	Alimentos	<input checked="" type="checkbox"/>	0	Edit	Delete
4	Móveis	<input checked="" type="checkbox"/>	0	Edit	Delete
5	Esportes	<input checked="" type="checkbox"/>	0	Edit	Delete



Complete the word

Categorias e Subcategorias

Nome de Categoria Ativa [+ Cadastrar Categoria](#)

Nome de SubCategoria Smartphones Ativa [+ Cadastrar SubCategoria](#)

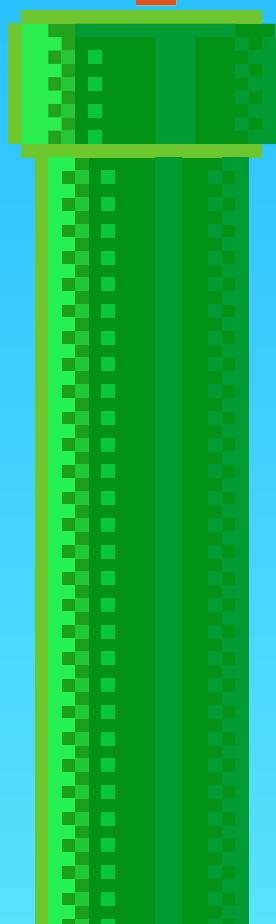
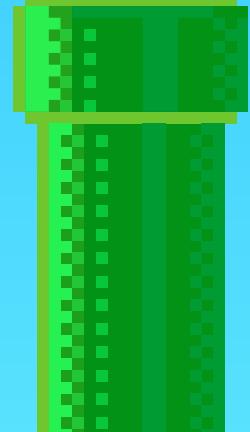
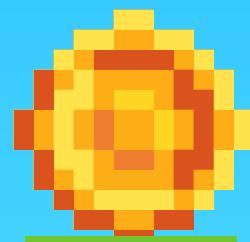
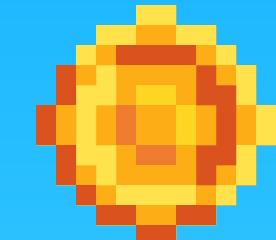
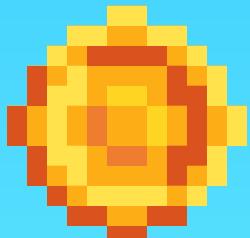
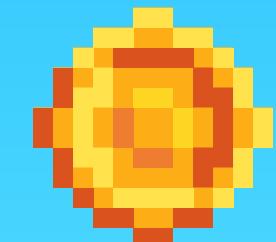
Categorias principais dos produtos [Recarregar](#)

ID	Nome	Status	Data Cadastrada	Data Alteração	Editar	Excluir
1	Smartphones	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Editar	Excluir
2	Roupas	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Editar	Excluir
3	Alimentos Não Perecíveis	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Editar	Excluir
4	Móveis de Sala	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Editar	Excluir
5	Equipamentos Esportivos	<input checked="" type="checkbox"/>	29/11/2023	01/01/0001	Editar	Excluir

Subcategorias dos produtos [Recarregar](#)

ID	Nome	Situação	Categoria	Editar	Excluir
0		<input checked="" type="checkbox"/>		Editar	Excluir
0		<input checked="" type="checkbox"/>		Editar	Excluir
0		<input checked="" type="checkbox"/>		Editar	Excluir
0		<input checked="" type="checkbox"/>		Editar	Excluir
0		<input checked="" type="checkbox"/>		Editar	Excluir

Tabelas atualizadas com sucesso [OK](#)



Complete the word

```
using StockApp.Core.Entidades.Comun;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace StockApp.Core.Repository
{
    4 references
    public interface ICategoriaRepository
    {
        2 references
        bool Incluir(Categoria categoria);

        3 references
        bool Atualizar(Categoria categoria, int id);

        2 references
        bool Deletar(int id);

        4 references
        List<Categoria> ObterTodos();

        1 reference
        List<Categoria> FiltrarCategoria(string nome);

        1 reference
        List<Categoria> ObterTodosPorStatus(int status);
        3 references
        List<Categoria> ObterPorNome(string nomeCategoria);
        1 reference
        List<Categoria> ObterPorId(int id);
    }
}
```

```
using StockApp.Core.Entidades;
using StockApp.Core.Entidades.Comun;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace StockApp.Core.Repository
{
    3 references
    public interface ICategoriaRepository
    {
        2 references
        bool Incluir(Categoria categoria);

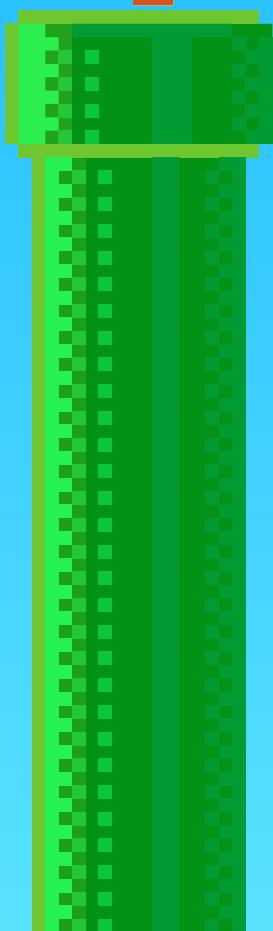
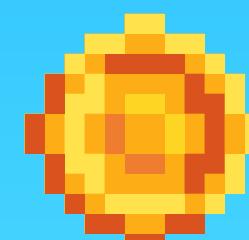
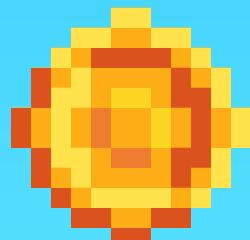
        2 references
        bool Atualizar(Categoria categoria);

        2 references
        bool Deletar(int id);

        4 references
        List<Categoria> ObterTodos();

        1 reference
        List<Categoria> FiltrarCategoria(string nome);

        1 reference
        List<Categoria> ObterTodosPorStatus(int status);
        1 reference
        Categoria ObterPorId(int id);
        3 references
        List<Categoria> ObterPorNome(string nomeCategoria);
    }
}
```



SELEÇÃO DE TELA

There are 6 different types of games!

Tela Principal

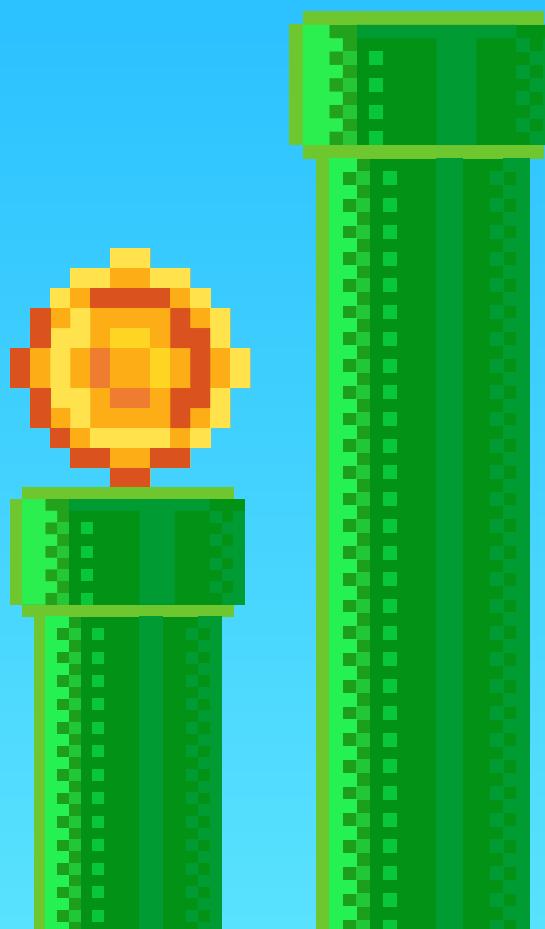
Tela login
usuário

Tela
Categoria

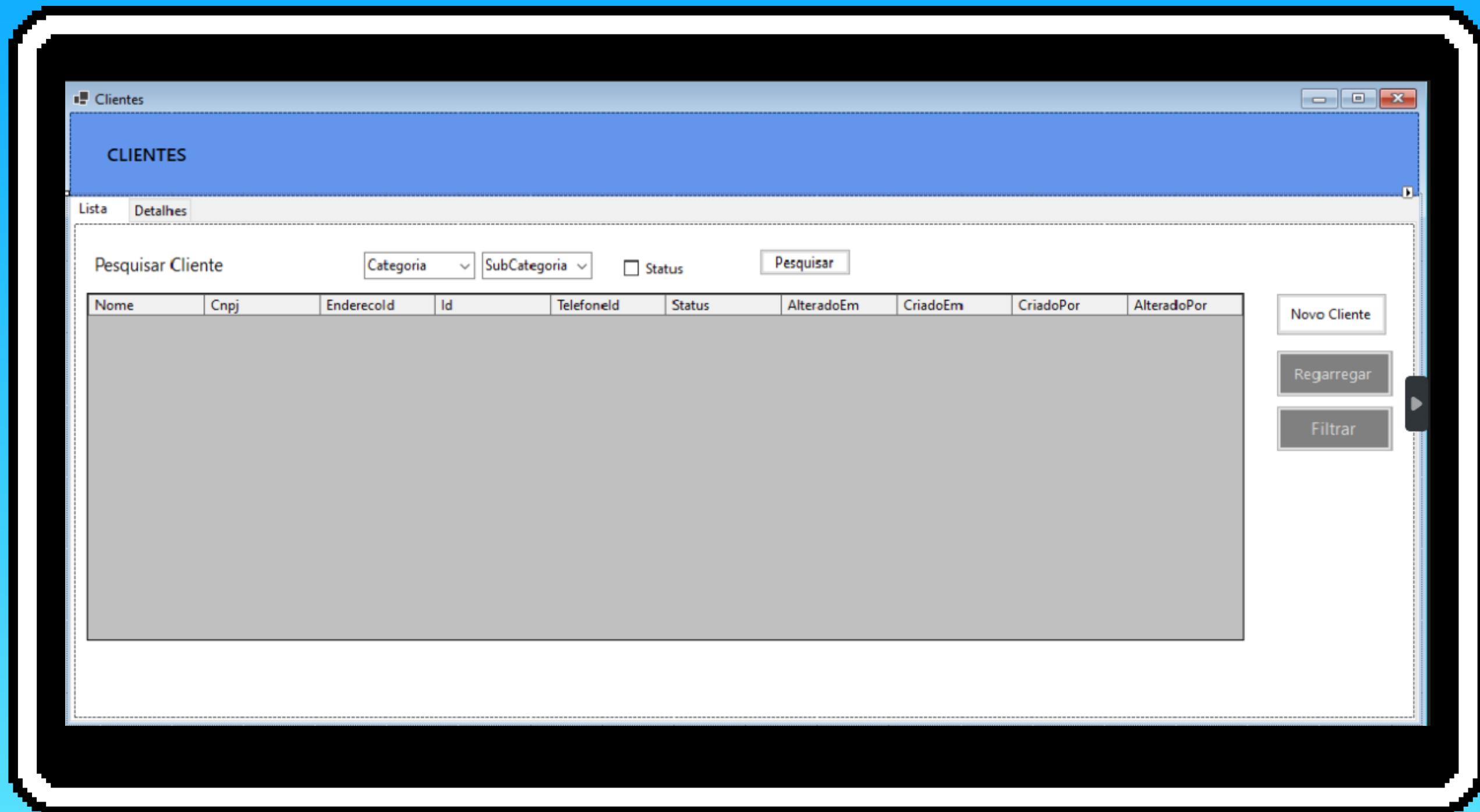
Tela
cliente

Tela
colaborador

Tela
Produtos



Answer the question



Complete the word

Clients

CLIENTES

[Lista](#) [Detalhes](#)

Cliente ativo



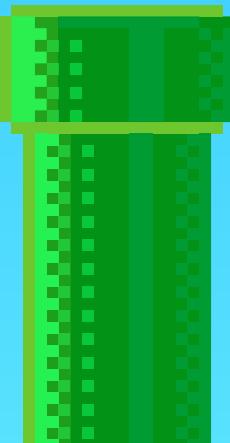
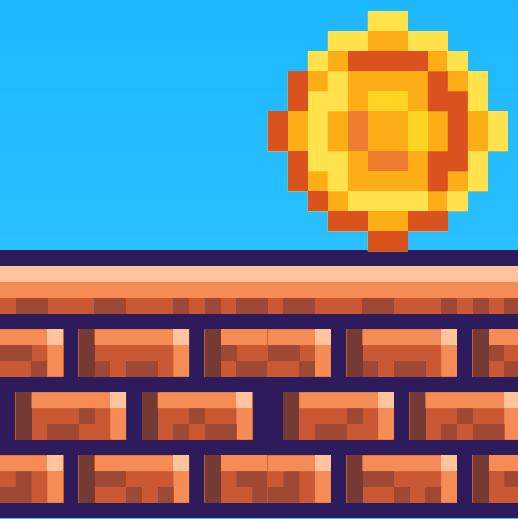
Nome	Cnpj	Telefone
<input type="text"/>	<input type="text"/>	<input type="text"/>

Endereço	Cep	Estado
<input type="text"/>	<input type="text"/>	<input type="text"/>

Cidade	Bairro	Rua
<input type="text"/>	<input type="text"/>	<input type="text"/>

Numero

Status



Complete the word

```
namespace STOCKAPP.DATA.SQLSERVER.REPOSITORIO
{
    4 references
    public class ClienteRepository : IClientesRepository
    {
        1 reference
        public bool Atualizar(Cliente cliente)
        {
            try
            {
                using (var connection = new SqlConnection(SqlServerConnection.Conexao))
                {

                    var sql = @"UPDATE [dbo].[Colaborador]
                               SET
                                   [Nome] = @nome,
                                   [Cpf] = @Cpf,
                                   [Telefone] = @Telefone,
                                   [Email] = @Email,
                                   [Departamento] = @Departamento,
                                   [DataNascimento] = @DataNascimento,
                                   [Bairro] = @Bairro
                                   [Cidade] = @Cidade
                                   [Complemento] = @Complemento,
                                   [Uf] = @Uf,
                                   [Logradouro] = @Logradouro,
                                   [Cep] = @Cep,
                               WHERE
                                   [Id] = @id";
                    //Protege os valores que estão chegando pela Classe Cargo de SqlInjection
                    //E passa para o Dapper Substituir no "var sql" os valores @ pelo valor que chegou
                    //no parametro.
                    var parametros = new DynamicParameters();
                    parametros.Add("@Nome", cliente.Nome);
                    parametros.Add("@Telefone", cliente.TelefoneId);
                    parametros.Add("@Cnpj", cliente.Cnpj);
                    parametros.Add("@EnderecoId", cliente.EnderecoId);
                    parametros.Add("@Status", cliente.Status);
                    parametros.Add("@CriadoEm", cliente.CriadoEm);
                    parametros.Add("@CriadoPor", cliente.CriadoPor);
                    parametros.Add("@AlteradoEm", cliente.AlteradoEm);
                    parametros.Add("@AlteradoPor", cliente.AlteradoPor);
                    parametros.Add("@Id", cliente.Id);

                    var linhasAfetadas = connection.Execute(sql, parametros);

                    return linhasAfetadas == 1;
                }
            }
        }
    }
}

> using StockApp.Core.Entidades;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

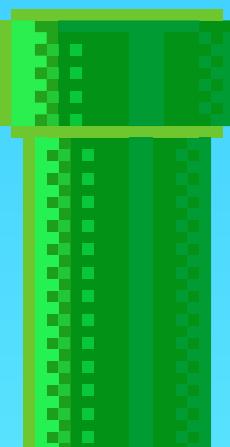
namespace StockApp.Core.Repository
{
    1 reference
    public interface IClientesRepository
    {
        1 reference
        bool Incluir(Cliente cliente);
        1 ref readonly struct System.Boolean
        Represents a Boolean (true or false) value.

        1 reference
        bool Deletar (int id);

        3 references
        List<Cliente> ObterTodos();

        1 reference
        List<Cliente> ObterTodosPorStatus (int status);

        1 reference
        Cliente ObterPorId (int id);
    }
}
```



SELEÇÃO DE TELA

There are 6 different types of games!

Tela Principal

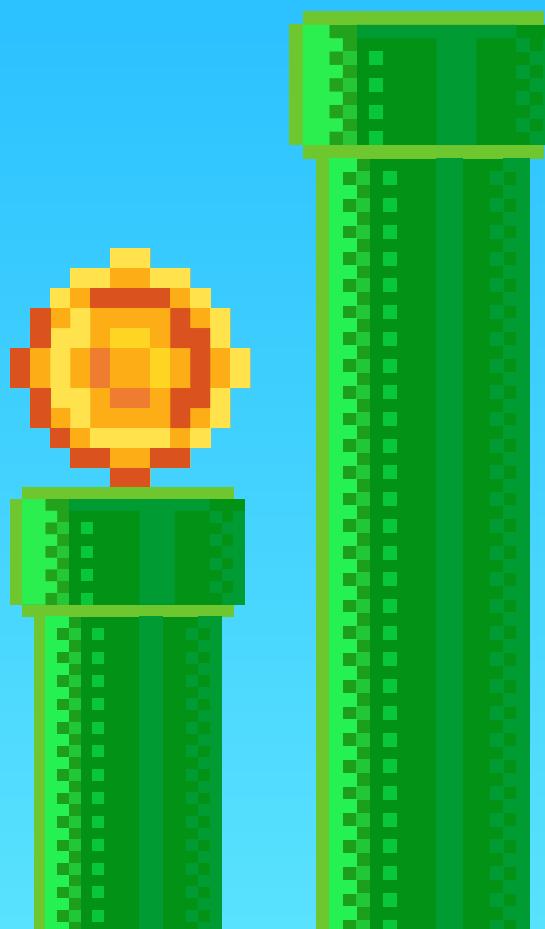
Tela login
usuário

Tela
Categoria

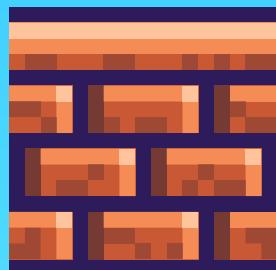
Tela
cliente

Tela
colaborador

Tela
Produtos



Answer the question



Answer the question

COLABORADORES

[Lista](#) [Detalhes](#)

Pesquisar Colaborador Status Ativo

	Nome	CPF	Telefone	Data de Nascimento	Departamento	Email	Status	Criado Em	Criado Por	Alterado Em
▶	987.654.321-03			05/15/1995 00:00:00	Analista	maria.oliveira@...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
	555.888.777-09			08/25/1980 00:00:00	Financeiro	pedro.santos@...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
	777.888.999-11			11/15/1984 00:00:00	RH	cristina.oliveira...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
	123.456.789-12			03/08/1990 00:00:00	Juridico	juliana.silva@e...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
	987.654.321-23			09/12/1986 00:00:00	Legislativo	fabio.oliveira@...	True	29/11/2023 20:08	Admin	29/11/2023 20:08

Complete the word

COLABORADORES

[Lista](#) [Detalhes](#)

Pesquisar Colaborador Status Ativo

	Nome	CPF	Analista Financeiro RH Juridico Legislativo	Data de Nascimento	Departamento	Email	Status	Criado Em	Criado Por	Alterado Em
▶		987.65		05/15/1995 00:0...	Analista	maria.oliveira@e...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
		555.888.777-09		08/25/1980 00:0...	Financeiro	pedro.santos@e...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
		777.888.999-11		11/15/1984 00:0...	RH	cristina.oliveira...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
		123.456.789-12		03/08/1990 00:0...	Juridico	juliana.silva@e...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
		987.654.321-23		09/12/1986 00:0...	Legislativo	fabio.oliveira@e...	True	29/11/2023 20:08	Admin	29/11/2023 20:08

Multiple choice

COLABORADORES

[Lista](#) [Detalhes](#)

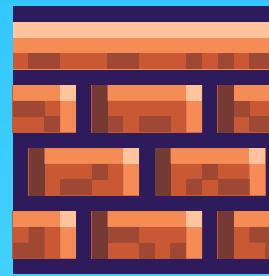
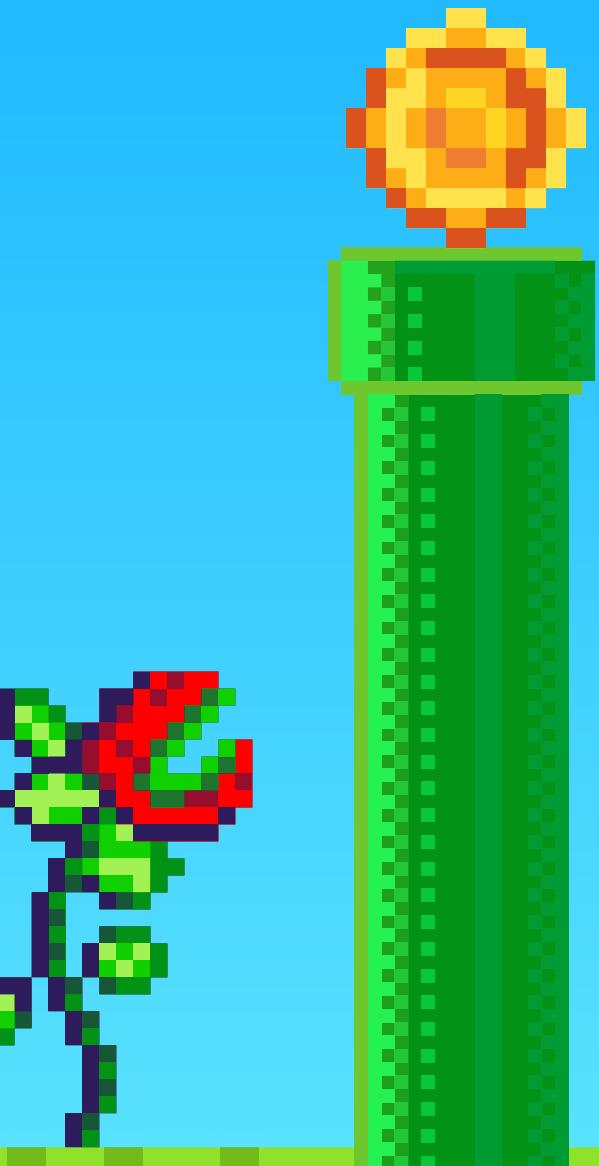
 Nome CPF Colaborador ativo

Telefone Email Data de Nascimento
 Departamento 29/11/2023

Cep: Buscar

Logradouro Numero Complemento

Bairro Cidade UF Rua



Answer the question

COLABORADORES

[Lista](#) [Detalhes](#)

Pesquisar Colaborador Status Ativo [Pesquisar](#) [Filtrar](#) [Regarregar](#) [Novo Colaborador](#)

	Nome	CPF	Telefone	Data de Nascimento	Departamento	Email	Status	Criado Em	Criado Por	Alterado Em
▶		987.654.321-03		05/15/1995 00:00:00	Analista	maria.oliveira@...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
		555.888.777-09		08/25/1980 00:00:00	Financeiro	pedro.santos@...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
		777.888.999-11		11/15/1984 00:00:00	RH	cristina.oliveira...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
		123.456.789-12		03/08/1990 00:00:00	Juridico	juliana.silva@e...	True	29/11/2023 20:08	Admin	29/11/2023 20:08
		987.654.321-23		09/12/1986 00:00:00	Legislativo	fabio.oliveira@...	True	29/11/2023 20:08	Admin	29/11/2023 20:08

Answer the question

```
using StockApp.Core.Entidades.Comum;

namespace StockApp.Core.Entidades
{
    public class Colaborador1 : EntidadeBase
    {
        public string Nome { get; set; }
        public string Cpf { get; set; }
        public string Telefone { get; set; }
        public string Departamento { get; set; }
        public string Cargo { get; set; }
        public int TelefoneId { get; set; }
        public int CargoId { get; set; }
        public int EnderecoId { get; set; }
    }
}
```



Answer the question

```
int IdTelefone = connection.QueryFirst<int>(sqlTele, colaborador);

//-----CARGO-----
string sqlCargo = @"INSERT INTO Cargo (NomeCargo, DepartamentoId) OUTPUT INSERTED.Id
VALUES
(@NomeCargo, @DepartamentoId);";

var parametrosCargo = new DynamicParameters();
parametrosCargo.Add("@NomeCargo", colaborador.Cargo);

int IdDepartamento = connection.QueryFirst<int>(sqlCargo, colaborador);

//-----COLABORADOR-----
string sqlColab = @"INSERT INTO Colaborador
(DepartamentoId, Departamento, Nome, Cpf, Email, DataNascimento,
TelefoneId, EnderecoId, Status, CriadoEm, CriadoPor, AlteradoEm, AlteradoPor)
VALUES
(@DepartamentoId, @Departamento, @Nome, @Cpf, @Email, @DataNascimento,
@TelefoneId, @EnderecoId, @Status, @CriadoEm, @CriadoPor, @AlteradoEm, @AlteradoPor)";

var parametrosColab = new DynamicParameters();
parametrosColab.Add("@DepartamentoId", IdDepartamento);
parametrosColab.Add("@Departamento", colaborador.Departamento);
parametrosColab.Add("@Nome", colaborador.Nome);
parametrosColab.Add("@Cpf", colaborador.Cpf);
parametrosColab.Add("@Email", colaborador.Email);
parametrosColab.Add("@DataNascimento", colaborador.DataNascimento);
parametrosColab.Add("@TelefoneId", IdTelefone);
parametrosColab.Add("@EnderecoId", IdEndereco);
parametrosColab.Add("@Status", colaborador.Status);
parametrosColab.Add("@CriadoEm", colaborador.CriadoEm);
parametrosColab.Add("@CriadoPor", colaborador.CriadoEm);
parametrosColab.Add("@AlteradoEm", colaborador.AlteradoEm);
parametrosColab.Add("@AlteradoPor", colaborador.AlteradoPor);

var linhasAfetadas = connection.Execute(sqlColab, colaborador);

return linhasAfetadas == 1;
}
catch (Exception ex)
{
    // Faça algo com a exceção, como logar ou lançar uma exceção personalizada com uma mensagem descritiva
    throw new Exception("Erro ao incluir Colaborador.", ex);
}
```

Multiple choice

Solution 'Stockgame' (3 of 3 projects)

- Aplicacao
- StockApp.Core
 - Dependencies
 - Entidades
 - Comum
 - Categoria.cs
 - Cliente.cs
 - Colaborador.cs
 - Colaborador1.cs
 - EmailService.cs
 - Produto.cs
 - SubCategoria.cs
 - Usuario.cs
 - Repository
 - ICategoriaRepository.cs
 - IClientesRepository.cs
 - IColaboradorRepository.cs
 - IProdutoRepository.cs
 - ISubCategoria.cs
 - IUsuarioRepository.cs
 - Templates
 - EmailTemplate.cs
 - recuperar-senha.html
 - Ultilidade
 - GerarSenha.cs
 - Validator
 - ValidacaoCpf.cs

SELEÇÃO DE TELA

There are 6 different types of games!

Tela Principal

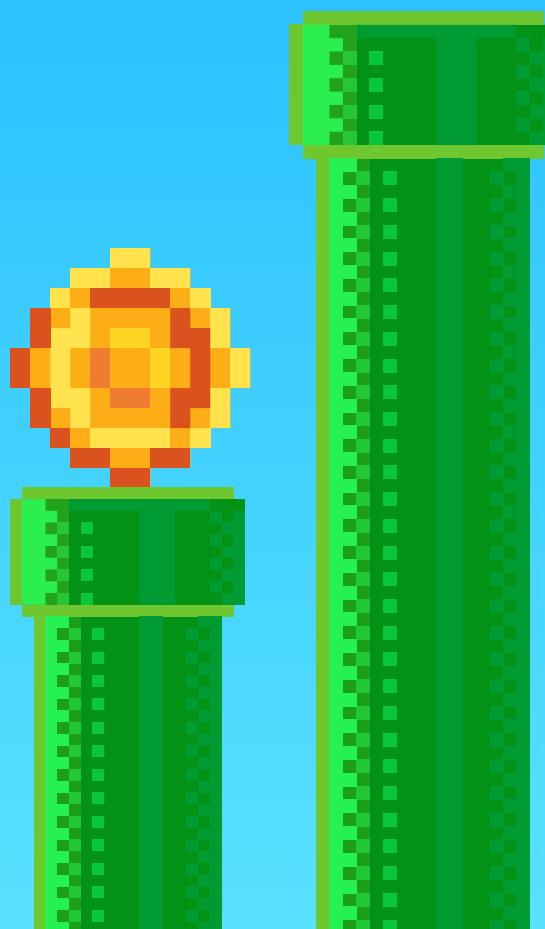
Tela login
usuário

Tela
Categoria

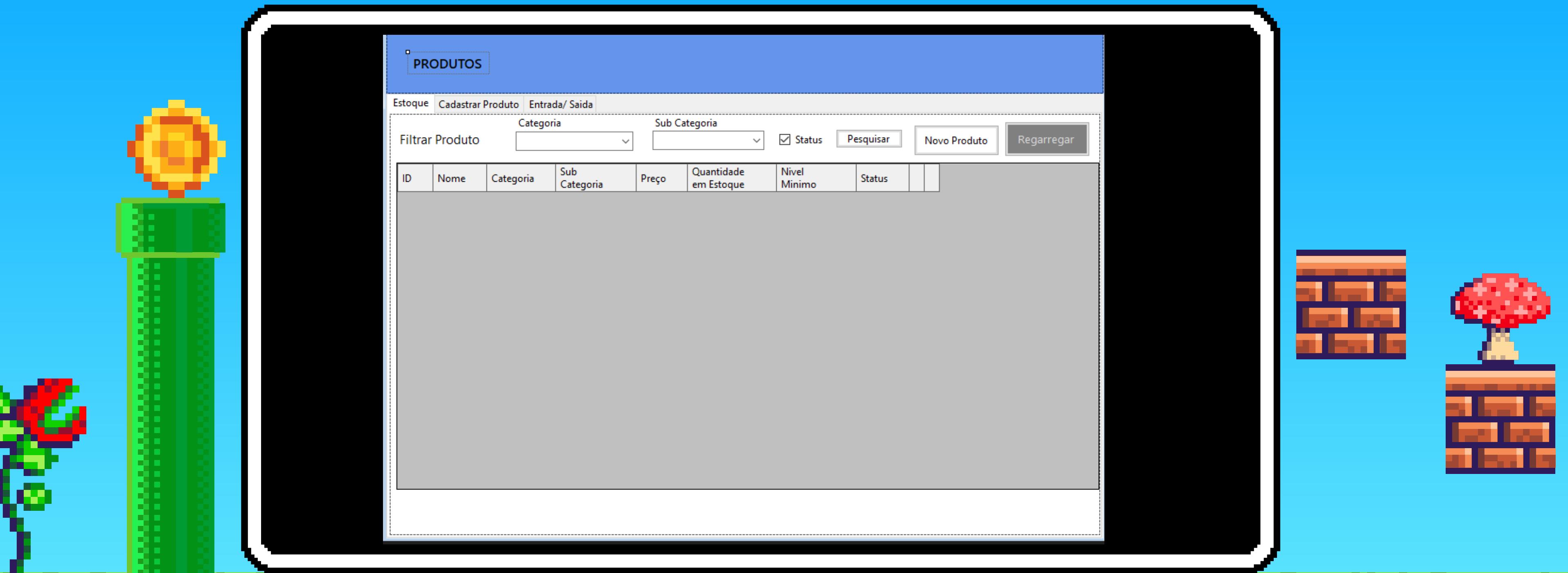
Tela
cliente

Tela
colaborador

Tela
Produtos



Multiple choice



Complete the word

PRODUTOS

Estoque Cadastrar Produto Entrada/ Saída

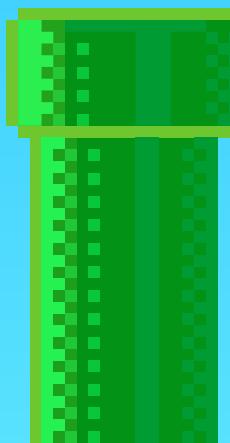
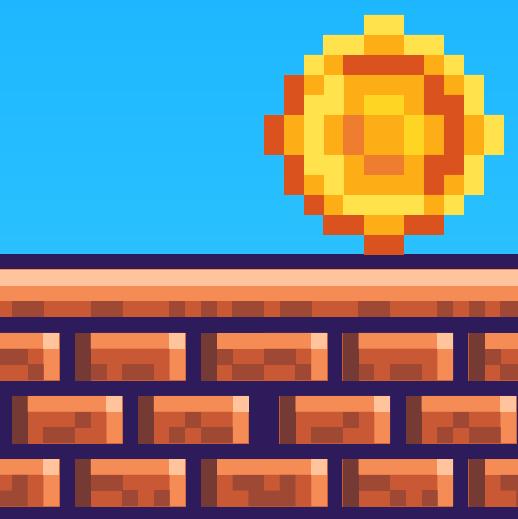
Produto	Id	Quantidade atual	Quantidade	Tipo	Cliente	Registrar
---------	----	------------------	------------	------	---------	-----------

Entrada

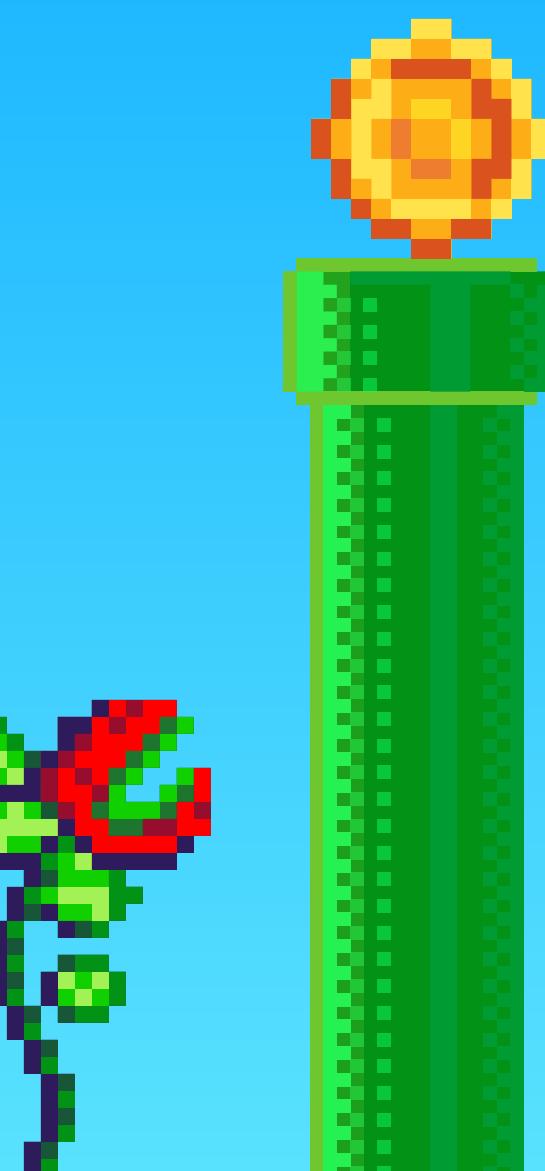
ID	Nome	Quantidade	Data Entrada
----	------	------------	--------------

Saída

ID	Nome	Quantidade	Data Saída	Cliente
----	------	------------	------------	---------



Multiple choice



A screenshot of Microsoft Visual Studio showing a solution named 'Stockgame' with three projects: 'Aplicacao', 'StockApp.Core', and 'BancoDados'. The 'StockApp.Core' project is expanded, showing its structure including 'Entidades' (with 'Comum' and 'Produto' classes), 'Apresentacao' (with 'StockApp.Desktop' and 'Produtos' classes), and 'BancoDados' (with 'StockApp.Data.SqlServer' and 'Repositorio' classes). The 'Entidades' folder contains the 'Produto.cs' file, which is open in the code editor. The code editor shows the implementation of the 'Produto' class with properties like Id, Cliente, Nome, Categoria, SubCategoria, Preco, Quantidade, Status, DataEntrada, DataSaida, CriadoPor, CriadoEm, AlteradoPor, and AlteradoEm. It also includes methods for inserting into the Produto and Estoque tables and updating the Produto table.

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using StockApp.Core.Entidades.Comum;

namespace StockApp.Core.Entidades
{
    public class Produto
    {
        public int Id { get; set; }

        public string Cliente { get; set; }

        public string Nome { get; set; }

        public string Categoria { get; set; }

        public string SubCategoria { get; set; }

        public int Preco { get; set; }

        public int Quantidade { get; set; }

        public int NvMinimo { get; set; }

        public bool Status { get; set; }

        public int SubCategoriaId { get; set; }

        public DateTime DataEntrada { get; set; }

        public DateTime DataSaida { get; set; }

        public string CriadoPor { get; set; }

        public DateTime CriadoEm { get; set; }

        public string AlteradoPor { get; set; }

        public DateTime AlteradoEm { get; set; }
    }
}

public bool Incluir(Produto produto)
{
    using(var connection = new SqlConnection(SqlServerConnection.Conexao))
    {
        var sql = @"INSERT INTO Produto (Nome, subCategoriaId, Preco, Status, CriadoEm, CriadoPor, AlteradoEm, OUTPUT INSERTED.Id
VALUES (@nome, @subcategoriaId, @Preco, @Status, @CriadoEm, @CriadoPor,@AlteradoEm,
@CriadoPor, @AlteradoEm)";

        var parametro = new DynamicParameters();
        parametro.Add("@nome", produto.Nome);
        parametro.Add("@subcategoriaId", produto.SubCategoriaId);
        parametro.Add("@Preco", produto.Preco);
        parametro.Add("@Status", produto.Status.ToString());
        parametro.Add("@CriadoEm", DateTime.Now);
        parametro.Add("@CriadoPor", "ADM");
        parametro.Add("@AlteradoEm", DateTime.Now);
        parametro.Add("@AlteradoPor", "ADM");

        int idProduto;
        idProduto = connection.QueryFirstOrDefault<int>(sql,parametro);

        var sqlEstoque = @"INSERT INTO Estoque (ProdutoId,Quantidade, NvMinimo)
VALUES
(@productId,@quantidade, @nvMinimo)"
        ;
        var parametroEstoque = new DynamicParameters();
        parametroEstoque.Add("@productId", idProduto);
        parametroEstoque.Add("@quantidade", 0);
        parametroEstoque.Add("@nvMinimo", produto.NvMinimo);

        var resultadoEstoque = connection.Execute(sqlEstoque,parametroEstoque);
        if (resultadoEstoque == 1)
        {
            return true;
        }
        return false;
    }
}

public bool Atualizar(Produto produto, int subCategoriaId)
{
    using (var connection = new SqlConnection(SqlServerConnection.Conexao))
    {
        try
        {
            var sql = @"UPDATE Produto SET Nome = @Nome, subCategoriaId = @subCategoriaId, Preco = @Preco, Status = @Status, CriadoEm = @CriadoEm, CriadoPor = @CriadoPor, AlteradoEm = @AlteradoEm WHERE Id = @Id";
            var parametro = new DynamicParameters();
            parametro.Add("@Nome", produto.Nome);
            parametro.Add("@subCategoriaId", produto.SubCategoriaId);
            parametro.Add("@Preco", produto.Preco);
            parametro.Add("@Status", produto.Status.ToString());
            parametro.Add("@CriadoEm", DateTime.Now);
            parametro.Add("@CriadoPor", "ADM");
            parametro.Add("@AlteradoEm", DateTime.Now);
            parametro.Add("@AlteradoPor", "ADM");
            var resultado = connection.Execute(sql,parametro);
            if (resultado == 1)
            {
                return true;
            }
        }
        catch (Exception ex)
        {
            throw ex;
        }
    }
}
```



GAME OVER

END

