



COLÉGIO PEDRO II – U. E. SÃO CRISTÓVÃO III

Professor: João Lagôas

Coordenação: Flavio Costa

TESTE - (LINGUAGEM DE PROGRAMAÇÃO 2 – 4,0 PTS)

NOME: _____ NÚMERO: _____ TURMA: DS _____ DATA: __ / __ / __

RECURSO – BANCO DE DADOS

Considere um sistema composto por um banco de dados denominado **pokedex** contendo 2 tabelas. Uma instância deste banco é mostrada através das tabelas abaixo. O nome da coluna sublinhada significa uma chave primária enquanto a coluna em negrito uma **chave estrangeira**.

pokemon(<u>id_pokemon</u> , nome, descrição, tipo1 , tipo2)					tipo(<u>id_tipo</u> , nome, Cor)		
<u>id_pokemon</u>	nome	descricao	tipo1	tipo2	id_tipo	nome	Cor
					0		#####
1	Bulbasaur	Bulbasaur é um Pokémon bonito nascido com uma gran...	10	14	1	inseto	#A6B91A
2	Ivysaur	Ivysaur (em japonês: フシギソウ Fushigisou) é um Pokémo...	10	14	2	noturno	#705746
3	Venusaur	Fisiologia. Venusaur é um Pokémon quadrúpede com	10	14	3	dragão	#6F35FC
4	Charmander	Charmander é um Pokémon lagarto de fogo que ...	7	0	4	elétrico	#F7D02C
5	Charmeleon	Charmeleon tem uma natureza perversa e constanteme...	7	0	5	fada	#D685AD
6	Charizard	Tem um fogo intenso dentro do seu corpo. As chamas...	7	8	6	lutador	#C22E28
7	Squirtle	Squirtle (em japonês ゼニガメ Zenigame) é um Pokémon....	18	0	7	fogo	#EE8130
8	Wartortle	Wartortle são pequenos, bípedes, tartaruga como Po...	18	0	8	voador	#A98FF3
9	Blastoise	Descrição. Blastoise tem canhões de água que se pr...	18	0	9	fantasma	#735797
					10	grama	#7AC74C
					11	terra	#E2BF65
					12	gelo	#96D9D6
					13	normal	#A8A77A
					14	veneno	#A33EA1
					15	psíquico	#F95587
					16	pedra	#B6A136
					17	metal	#B7B7CE
					18	água	#6390F0





RECURSO - SISTEMA

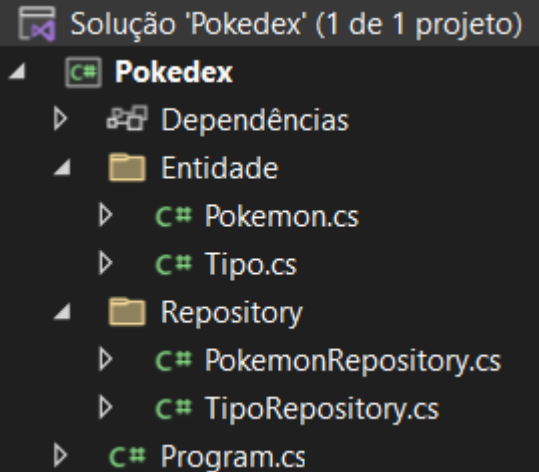
Com base no banco de dados apresentado, um sistema C# foi implementado e apresenta a seguinte estrutura. Os códigos dos arquivos Tipo.cs e TipoRepository.cs também são mostrados.

```
class Tipo
{
    public int IdTipo { get; set; }
    public string Nome { get; set; }
    public string Cor { get; set; }
}
```

```
class TipoRepository
{
    private readonly string _connectionString;

    public TipoRepository(string connectionString)
    {
        _connectionString = connectionString;
    }

    public Tipo ObterTipoPorId(int idTipo)
    {
        Tipo tipo = null;
        using (var connection = new MySqlConnection(_connectionString))
        {
            connection.Open();
            string query = "SELECT * FROM tipo WHERE id_tipo = @idTipo";
            using (var command = new MySqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@idTipo", idTipo);
                using (var reader = command.ExecuteReader())
                {
                    if (reader.Read())
                    {
                        tipo = new Tipo
                        {
                            IdTipo = reader.GetInt32("id_tipo"),
                            Nome = reader.GetString("nome"),
                            Cor = reader.GetString("cor")
                        };
                    }
                }
            }
        }
        return tipo;
    }
}
```





QUESTÃO DISCURSIVA

- 1) De acordo com todas as informações dadas, faça o que é pedido (2,0pts):
- Crie uma classe de entidade Pokémon, adicionando as propriedades necessárias que representam um registro (uma instância) de Pokémon (0,5 pt).
 - Crie uma classe PokémonRepository de acordo com o padrão de projeto Repository usado no sistema. Nesta classe, implemente os seguintes métodos (1,5 pts):
 - `public List<Pokemon> ObterTodosPokemons()`
 - `public int InserirPokemon(Pokemon pokemon)`
 - `public List<Pokemon> ObterTodosPokemonsPorTipo(int idTipo)`
Nota: Este método deve retornar todos os pokémon que são do tipo (tipo1 ou tipo2) idTipo.
- 2) O arquivo Program.cs contém essencialmente as instruções de entrada e saída de dados no console. Observe o código abaixo e faça o que é pedido (2,0 pts).

```
string connectionString = "Server=localhost;Database=pokedex;User=root;Password=";
PokemonRepository pokemonRepo = new PokemonRepository(connectionString);
TipoRepository tipoRepo = new TipoRepository(connectionString);

while (true)
{
    Console.WriteLine("\nEscolha uma opção:");
    Console.WriteLine("1 - Listar todos os Pokémon");
    Console.WriteLine("2 - Adicionar um novo Pokémon");
    Console.WriteLine("3 - Listar Pokémon por tipo");
    Console.WriteLine("4 - Sair");

    string opcao = Console.ReadLine();

    switch (opcao)
    {
        case "1":
            // .. I ..
            break;
        case "2":
            // .. II ..
            break;
        case "3":
            Console.Write("ID do tipo: ");
            int idTipo = Convert.ToInt32(Console.ReadLine());
            List<Pokemon> pokemons = pokemonRepo.ObterTodosPokemonsPorTipo(idTipo);

            for(int i = 0; i < pokemons.Count; i+=2)
            {
                Console.WriteLine($"Nome: {pokemons[i].Nome}");
            }
            break;
        case "4":
            return;
        default:
            Console.WriteLine("Opção inválida! Tente novamente.");
            break;
    }
}
```





- a) Implemente o trecho de código marcado por `..l..`. Este código deve imprimir na tela todos os pokémon registrados no banco de acordo com o seguinte formato de saída (1,0 pt):

```
ID: 1, Nome: Bulbasaur, Tipo1: grama, Tipo2: veneno
ID: 2, Nome: Ivysaur, Tipo1: grama, Tipo2: veneno
ID: 3, Nome: Venusaur, Tipo1: grama, Tipo2: veneno
ID: 4, Nome: Charmander, Tipo1: fogo, Tipo2:
ID: 5, Nome: Charmeleon, Tipo1: fogo, Tipo2:
ID: 6, Nome: Charizard, Tipo1: fogo, Tipo2: voador
ID: 7, Nome: Squirtle, Tipo1: água, Tipo2:
ID: 8, Nome: Wartortle, Tipo1: água, Tipo2:
ID: 9, Nome: Blastoise, Tipo1: água, Tipo2:
```

Exemplo de saída

- b) Implemente o trecho de código marcado por `..ll..`. Este código deve receber informações do usuário e inserir um novo registro de pokémon no banco (0,5 pt).
- c) Suponha que ao executar a função Main, e considerando que todos os demais códigos foram implementados corretamente, um usuário entra com as seguintes informações (0,5 pt):

```
3
ID do tipo: 7
```

O que será mostrado no Console?

