FACULDADE DE INFORMÁTICA E ADMINISTRAÇÃO PAULISTA (FIAP)

CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

GABRIEL SIQUEIRA RODRIGUES, 98626, 2TDSPF

GUSTAVO DE OLIVEIRA AZEVEDO, 550548, 2TDSPF

ISABELLA JORGE FERREIRA, 552329, 2TDSPG

MATHEUS MANTOVANI, 98524, 2TDSPF

JUAN DE GODOY, 551408, 2TDSPF

SÃO PAULO

JUNHO / 2024

Sumário

Econet

Em um cenário onde a preservação dos recursos marinhos se torna uma prioridade global, o nosso projeto surge com o objetivo de promover uma pesca mais sustentável, por meio da utilização de tecnologias avançadas para o monitoramento das espécies marinhas capturadas. Durante a atividade de pesca, as espécies de peixes serão monitoradas e identificadas, determinando se estão classificadas como Criticamente em Perigo (CR), Criticamente em Perigo Possivelmente Extinta (CR(PEX)), Em Perigo (EN), ou Vulnerável (VU). Essas informações serão georreferenciadas, ou seja, serão registradas as coordenadas de onde cada espécie foi capturada, alimentando um aplicativo que disponibilizará esses dados.

Com este sistema, os pescadores poderão evitar áreas onde há a presença de espécies ameaçadas, contribuindo para a preservação da biodiversidade marinha. Além disso, o aplicativo permitirá aos pescadores planejar suas atividades com base na disponibilidade das espécies que desejam capturar, evitando esforços infrutíferos e melhorando a eficiência da pesca. Essa ferramenta é crucial não apenas para a pesca sustentável e auxilio na tomada de decisões, mas também para a conservação das espécies, ao fornecer dados precisos e em tempo real sobre a distribuição das populações marinhas.

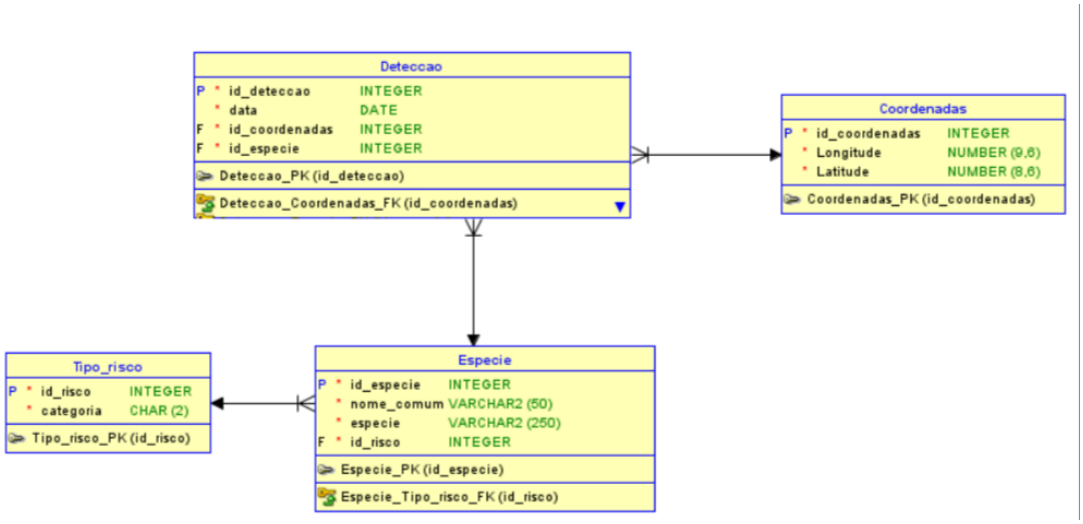
A longo prazo, o projeto também visa colaborar na fiscalização e no cumprimento das leis ambientais. As autoridades poderão utilizar os dados fornecidos pelo aplicativo para monitorar as áreas de pesca e garantir que espécies ameaçadas não sejam capturadas ilegalmente, aplicando multas e penalidades quando necessário. Além disso, biólogos e conservacionistas poderão utilizar as informações coletadas para desenvolver programas e estratégias de reprodução e conservação, combatendo de forma mais eficaz a extinção das espécies marinhas.

Em resumo, nosso projeto representa um passo significativo em direção à pesca sustentável e à preservação dos ecossistemas marinhos. Por meio da integração de tecnologia e conservação, esse projeto oferece soluções práticas e eficazes para proteger nossos oceanos e promover práticas de pesca mais sustentáveis, garantindo a sobrevivência das espécies marinhas.

Link do repositório (GitHub)

https://github.com/EcoNet-GlobalSolution/GlobalSolution-.NET.git

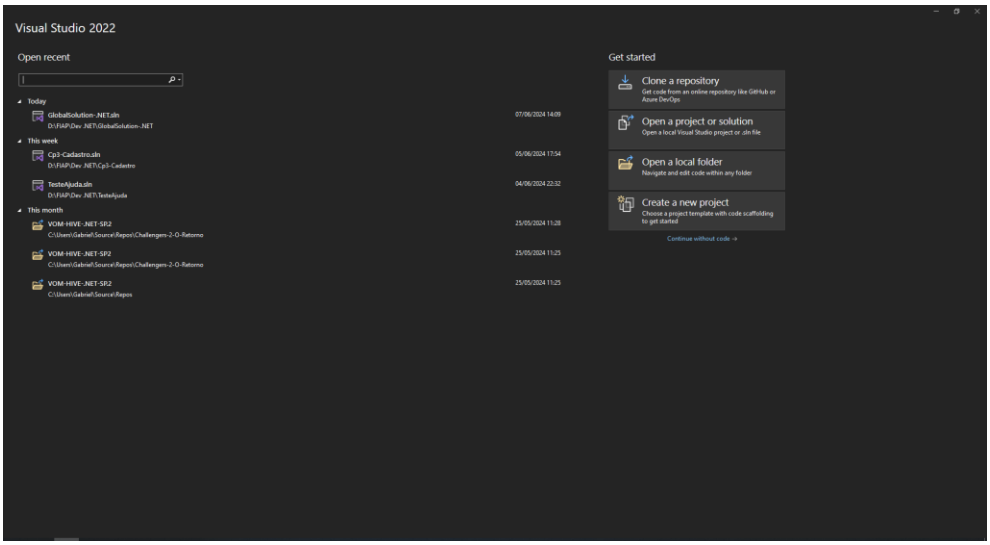
Diagrama



Como utilizar

1- Abra sua IDE (Exemplo pelo Visual Studio 2022).

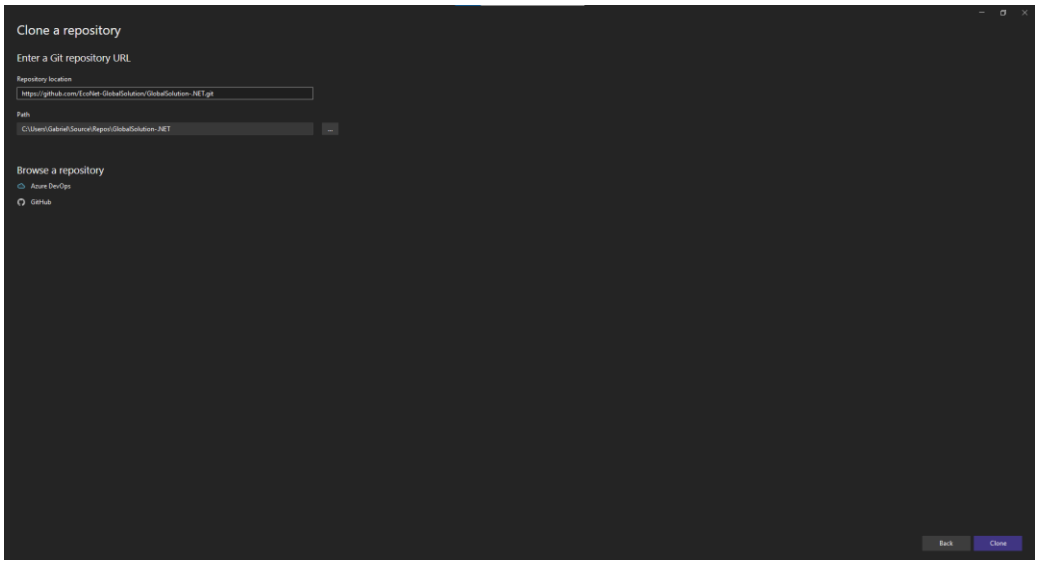
2- Selecione “Clone a repository” ou “Open a local folder”.



2.1- Clone a repository.

2.1.1- No campo Repository location, cole o link https://github.com/EcoNet-GlobalSolution/GlobalSolution-.NET.git

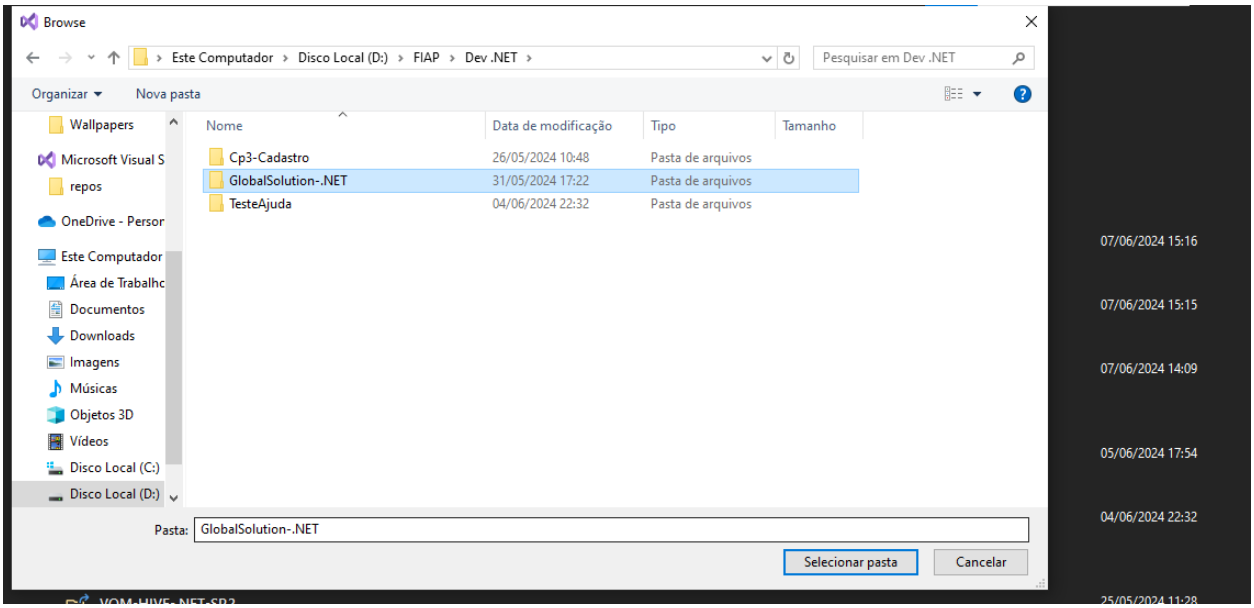
2.1.2- Clique em Clone.



2.2- Open a local folder.

2.2.1- Localize a pasta em seu computador e selecione a pasta chamada “GlobalSolution-.NET” .

2.2.2- Clique em “Selecionar pasta”.



2.2.3- Na sua Solution explorer, localize esse ícone.

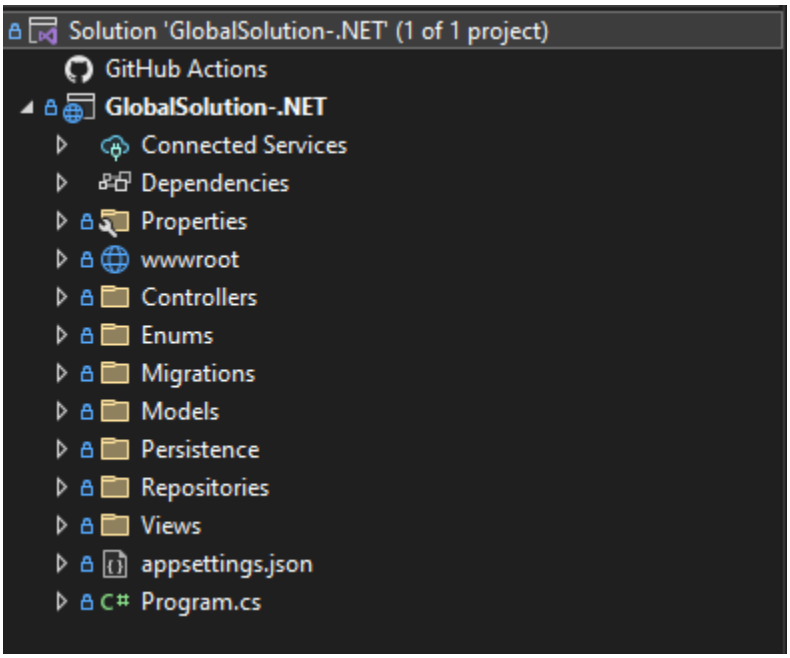
2.2.4- Clique duas vezes sobre ele.

3- No arquivo “appsettings.json” troque o Id e Password para suas credenciasi (usuário e senha)

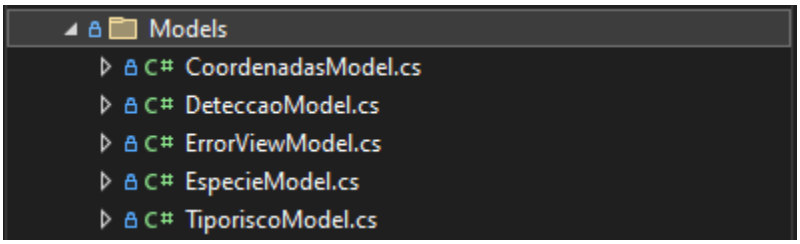
4- Localize o seu Package Manager Console e digite “update-database”

5- Localize esse ícone e clique nele

Estrutura de pastas



Models



Todas Models configuradas de acordo com o Diagrama da nossa solução.

CoordenadasModel

Id_coordenadas: Identificador único das coordenadas.

Longitude: Latitude da coordenada.

Latitude: Longitude da coordenada.

DeteccaoModel

Id_deteccao: Identificador único da detecção.

Data: Registra a data e hora de quando foi realizado a detecção.

EspecieModel

Id_especie: Identificador único da espécie

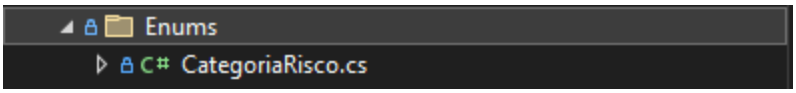
Nome_comum: Nome comum da espécie de peixe detectada

Especie: Nome científico da espécie detectada.

TiporiscoModel

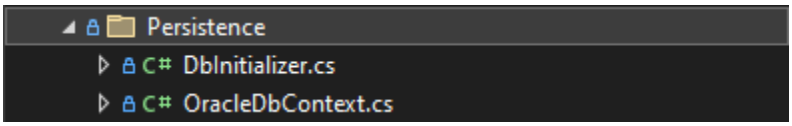
Id_risco: Identificador único do tipo risco.

Categoria: Qual categoria do risco que aquela espécie de peixe está, CR - Criticamente em Perigo, CR(PEX) - Criticamente em Perigo, Possivelmente Extinta, EN - Em Perigo, VU – Vulnerável (está dentro da pasta Enums)



Nesse arquivo CategoriaRisco.cs, foi definido um Enum para uma criação e lista suspensa na hora do preenchimento do CRUD.

Persistence



OracleDbContext

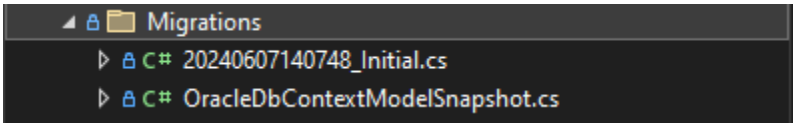
O OracleDbContext é uma classe essencial em nossa aplicação, responsável pela interação com o banco de dados Oracle utilizando o Entity Framework Core. Esta classe herda de DbContext, fornecendo uma API para realizar operações CRUD (Create, Read, Update, Delete) sobre as entidades definidas no modelo de dados.

O objetivo principal do OracleDbContext é gerenciar a conexão com o banco de dados e mapear as entidades do domínio da aplicação para as tabelas do banco de dados. Ele serve como um intermediário entre a aplicação e o banco de dados, facilitando a execução de consultas e a persistência de dados.

DbInitializer

A classe DbInitializer é fundamental para inicializar e popular o banco de dados da aplicação com dados pré-definidos. Ela garante que o banco de dados esteja preparado com os dados essenciais para que a aplicação funcione corretamente desde o início.

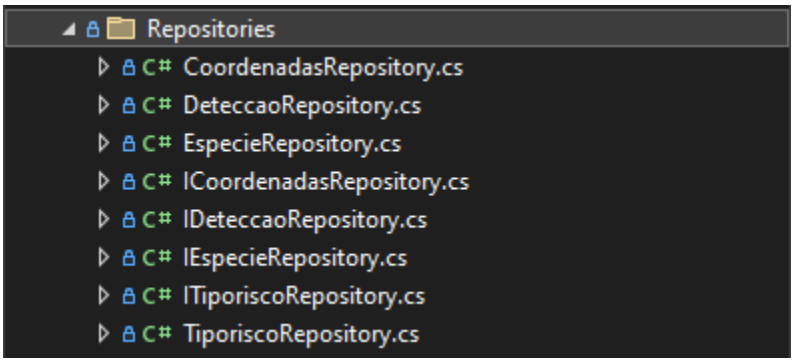
Migrations



A pasta Migrations é uma parte essencial do projeto, responsável por gerenciar as mudanças no esquema do banco de dados ao longo do desenvolvimento. Utilizando o Entity Framework, as migrações permitem que o banco de dados evolua conforme novas funcionalidades são adicionadas, garantindo que a estrutura do banco de dados esteja sempre em sincronia com os modelos de dados definidos no código.

A pasta Migrations tem como objetivo registrar e aplicar mudanças incrementais ao esquema do banco de dados, incluindo a criação, alteração ou remoção de tabelas, colunas, índices e outras estruturas de dados.

Repositories



Repository

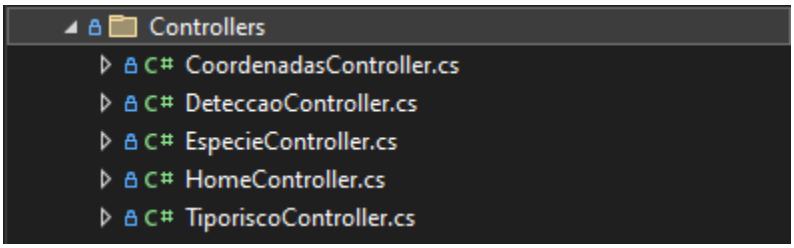
A pasta Repositories contém as implementações das interfaces `IRepository` para cada Model do projeto. Estas implementações são responsáveis por realizar as operações de CRUD (Create, Read, Update, Delete) no banco de dados, garantindo que os dados sejam manipulados de maneira consistente e eficiente.

Interfaces (IRepository)

As interfaces IRepository definem um conjunto de métodos essenciais para a manipulação das Models. Cada Model tem sua própria interface, garantindo que as operações específicas possam ser implementadas de acordo com as necessidades de cada entidade. Os métodos configurados nas interfaces são:

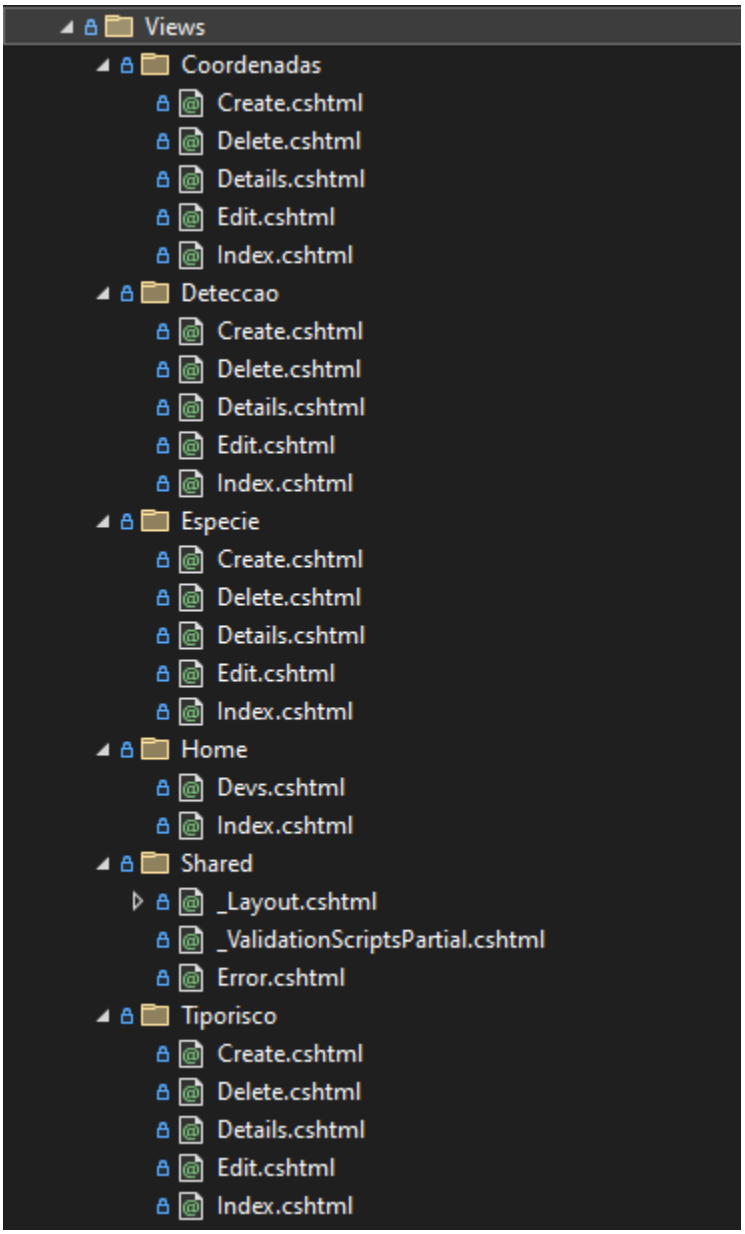
- **ListarPorId:** Retorna uma instância específica da Model com base no ID fornecido.
- **BuscarTodos:** Retorna uma lista de todas as instâncias da Model.
- **Adicionar:** Adiciona uma nova instância da Model ao banco de dados.
- **Atualizar:** Atualiza uma instância existente da Model no banco de dados.
- **Apagar:** Remove uma instância específica da Model do banco de dados.

Controllers



As controllers são fundamentais para o gerenciamento das operações de CRUD (Create, Read, Update, Delete) para todas as Models configuradas, garantindo a interação eficiente entre o usuário e o banco de dados. Cada controller é projetada para lidar com uma entidade específica do modelo, utilizando métodos definidos nas interfaces IRepository e implementados nas classes de repositório.

Views



Create

View responsável pela operação de Create do CRUD

Delete

View responsável pela operação de Delete do CRUD

Details

View que detalha os dados que selecionar para Details.

Edit

View responsável pela operação de Edit do CRUD.

Index

View responsável pela operação de Read do CRUD.

Devs

View que contém nome, rm e turma de todos os integrantes do grupo.