# FIAP

**Logotipo, nome da empresa

Descrição gerada automaticamente**

Luccas Silva – RM: 552890

Luís Henrique – RM: 552692

Igor Gabriel– RM: 553544

**Global Solution**

**Blue Future**

**São Paulo**

**2024**

# Sumário

Descritivo do Projeto .........................................................................................3

Objetivo Geral ....................................................................................................4

Descrição das Funcionalidades de Java .............................................................5

Tabela de EndPoints ...........................................................................................6

Protótipos ..........................................................................................................8

Modelo do Banco de Dados .............................................................................14

Diagramas de Classes ......................................................................................15

Links .................................................................................................................19

# Descritivo do Projeto

Nossa proposta consiste na criação de um portal destinado à disseminação de informações e ao gerenciamento para ONGs, permitindo a interação direta com seus colaboradores, independentemente da localização geográfica.

Visando a gestão de colaboradores, implementaremos um sistema de cadastro onde os interessados poderão se inscrever diretamente nas ONGs, submetendo-se a um processo de triagem realizado pela própria organização. Em caso de aprovação, a ONG deverá registrar as informações do colaborador no sistema, garantindo uma gestão eficiente.

Nosso objetivo ao desenvolver essa solução é agilizar e auxiliar as ONGs na execução de suas atividades de maneira eficaz, minimizando os processos burocráticos. Utilizaremos diversas tecnologias para criação e gestão das informações, incluindo a API Viacep para agilizar o cadastro e o consumo de dados, e uma API desenvolvida em Java para o envio de informações ao banco de dados.

Adicionalmente, incorporaremos um sistema de Disque Denúncia, onde os usuários poderão reportar casos de poluição e/ou destruição marinha. A partir dessas denúncias, a empresa poderá buscar colaboradores na região afetada por meio de nosso portal, formando assim uma linha de ação inicial para resolver o problema.

# Objetivo Geral

A proposta da nossa solução está centrada em quatro pilares principais: captação de novos colaboradores, gestão eficiente dos colaboradores para ONGs, facilitação de denúncias, e disseminação de informações.

Nosso objetivo é proporcionar uma plataforma robusta que permita às ONGs atrair novos colaboradores de forma contínua e sistematizada. O sistema de gestão de colaboradores será projetado para ser altamente eficiente, permitindo que as ONGs administrem suas equipes de forma ágil e organizada. Além disso, a plataforma incluirá um canal dedicado para denúncias, onde os usuários poderão reportar casos de poluição e destruição marinha. Essas denúncias serão tratadas com seriedade, e a plataforma permitirá que as ONGs localizem rapidamente colaboradores na região afetada para formar uma resposta inicial eficaz.

Para garantir que todas essas funcionalidades operem de maneira eficaz, linear e simples, utilizaremos tecnologias inovadoras. Entre elas, destacam-se a API Viacep, que agiliza o processo de cadastro e o consumo de dados, e uma API desenvolvida em Java para assegurar o envio seguro e eficiente de informações para o banco de dados.

Nosso compromisso é fornecer uma solução tecnológica avançada que facilite a missão das ONGs, promovendo uma gestão interna eficiente e uma comunicação externa transparente e ágil.

# Descrição das Funcionalidades do Java

A parte de Java se destaca por sua robusta capacidade de **gerenciar e processar informações** provenientes tanto do **front-end** quanto do **banco de dados**, de forma **simultânea e eficiente**. Através de uma arquitetura robusta e bem estruturada, o projeto garante a **consistência e confiabilidade** dos dados em todo o sistema.

**Funcionalidades Detalhadas:**

* **Criação e Gerenciamento de Tabelas do Banco de Dados:** O projeto permite a criação e o gerenciamento completos de tabelas no banco de dados, atendendo às necessidades específicas da aplicação. Isso garante que os dados sejam armazenados de forma organizada e eficiente, facilitando o acesso e a consulta.
* **Desenvolvimento de API para o Front-end:** Através da implementação de APIs RESTful, o projeto fornece uma interface amigável para o front-end interagir com o back-end. Isso permite que os desenvolvedores front-end consumam dados do banco de dados e realizem operações com facilidade, sem se preocupar com os detalhes da implementação do back-end.
* **Camadas de Abstração para Maior Clareza e Flexibilidade:** O projeto segue os princípios da arquitetura em camadas, dividindo a lógica em camadas bem definidas:
  + **Resource:** Define os endpoints da API e como eles serão acessados pelo front-end.
  + **Service:** Implementa a lógica de negócio da aplicação, encapsulando as regras e operações sobre os dados.
  + **Model:** Representa os modelos de dados da aplicação, definindo a estrutura das entidades e seus atributos.
  + **DAO (Data Access Object):** Abstrai o acesso ao banco de dados, isolando a lógica de negócio da tecnologia de banco de dados específica. Essa separação de responsabilidades torna o código mais claro, fácil de manter e testável.

**Benefícios do Projeto:**

* **Arquitetura robusta e escalável:** A arquitetura em camadas garante que o projeto seja **robusto, escalável e fácil de manter**, mesmo em cenários com alto volume de dados e usuários.
* **Maior produtividade:** A utilização de APIs e a abstração do acesso ao banco de dados permitem que os desenvolvedores se concentrem na lógica de negócio da aplicação, **aumentando a produtividade e a qualidade do código.**

**Consistência e confiabilidade dos dados:** A centralização do gerenciamento de dados garante que os dados sejam **consistentes e confiáveis** em todo o sistema, minimizando o risco de erros e inconsistências.

# Tabela de EndPoints

## Usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verbo HTTP | URI | Descrição | Request Body | Response Body |
| POST | /usuario | Cadastra um novo usuário | JSON contendo objeto Usuario | URI do usuário criado (Status: 201 Created) |
| PUT | /usuario/{Id\_Usuario} | Atualiza um usuário existente | JSON contendo objeto Usuario | Status 200 OK (sem corpo) |
| DELETE | /usuario/{Id\_Usuario} | Deleta um usuário |  | Status 200 OK (sem corpo) |
| GET | /usuario | Seleciona todos os usuários |  | Lista de objetos Usuario em JSON (Status: 200 OK) |

## ONG

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verbo HTTP | URI | Descrição | Request Body | Response Body |
| POST | /ong | Cadastra uma nova ONG. | JSON contendo os dados da ONG. | Status 201 Created (URI da nova ONG) |
| PUT | /ong/{Id\_ONG} | Atualiza uma ONG existente. | JSON contendo os dados atualizados da ONG. | Status 200 OK (sem corpo) |
| DELETE | /ong/{Id\_ONG} | Deleta uma ONG existente. |  | Status 200 OK (sem corpo) |
| GET | /ong | Seleciona todas as ONGs cadastradas. |  | Lista de ONGs em JSON |

## NewsLetter

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verbo HTTP | URI | Descrição | Request Body | Response Body |
| POST | /newsletter | Cadastrar uma nova Newsletter | NewsLetter (objeto JSON) | Status: 201 Created |
| PUT | /newsletter/{Id\_NewsLetter} | Atualizar uma Newsletter existente | NewsLetter (objeto JSON) | Status: 200 OK |
| DELETE | /newsletter/{Id\_NewsLetter} | Deletar uma Newsletter |  | Status 200 OK |
| GET | /newsletter | Selecionar todas as Newsletters |  | Lista de NewsLetter (objetos JSON) |

## Denuncia

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verbo HTTP | URI | Descrição | Request Body | Response Body |
| POST | /denuncia | Cadastrar uma nova denuncia | denuncia (objeto JSON) | Status: 201 Created |
| PUT | /denuncia/{Id\_Denuncia} | Atualizar uma denuncia existente | denuncia (objeto JSON) | Status: 200 OK |
| DELETE | /denuncia/{Id\_Denuncia} | Deletar uma denuncia |  | Status 200 OK |
| GET | /denuncia | Selecionar todas as denuncia |  | Lista de denuncia (objetos JSON) |

# Protótipos

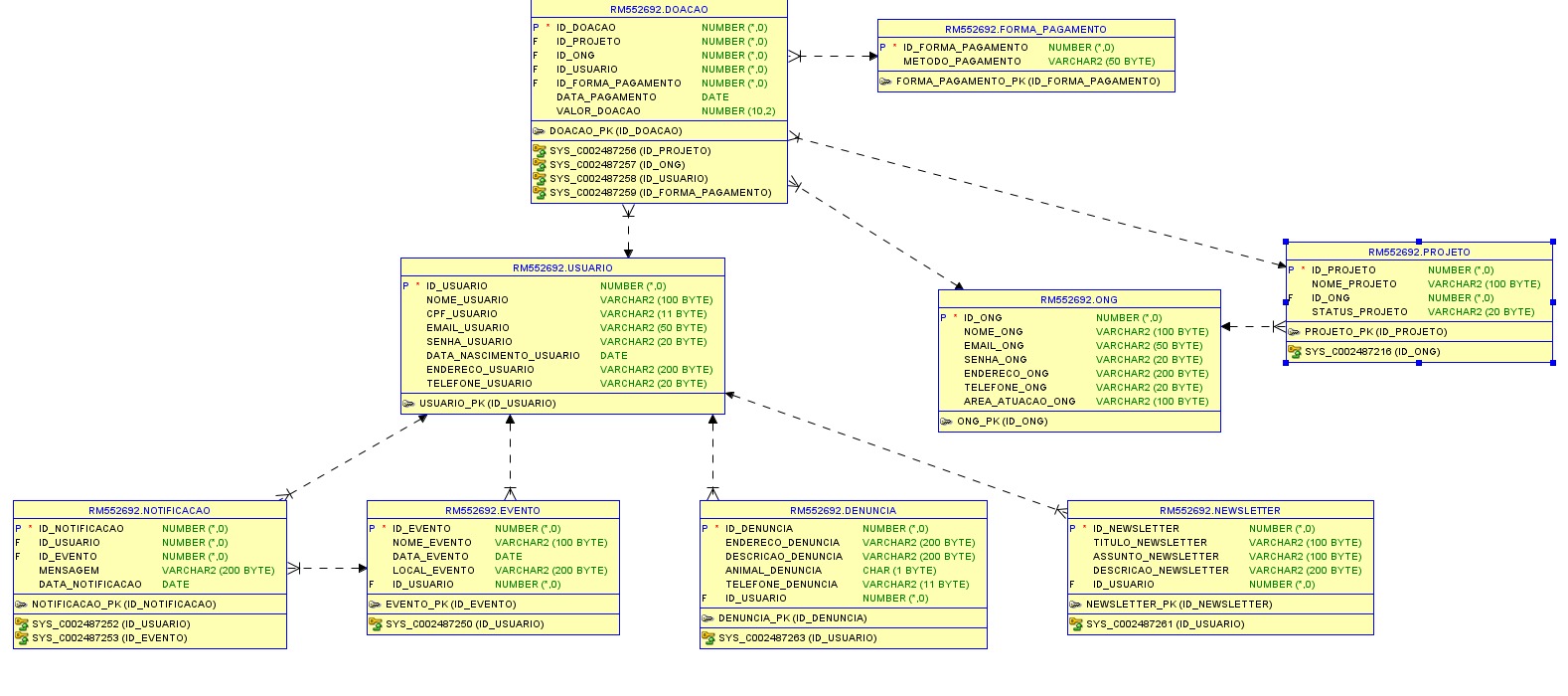
# Interface gráfica do usuário, Aplicativo Descrição gerada automaticamente

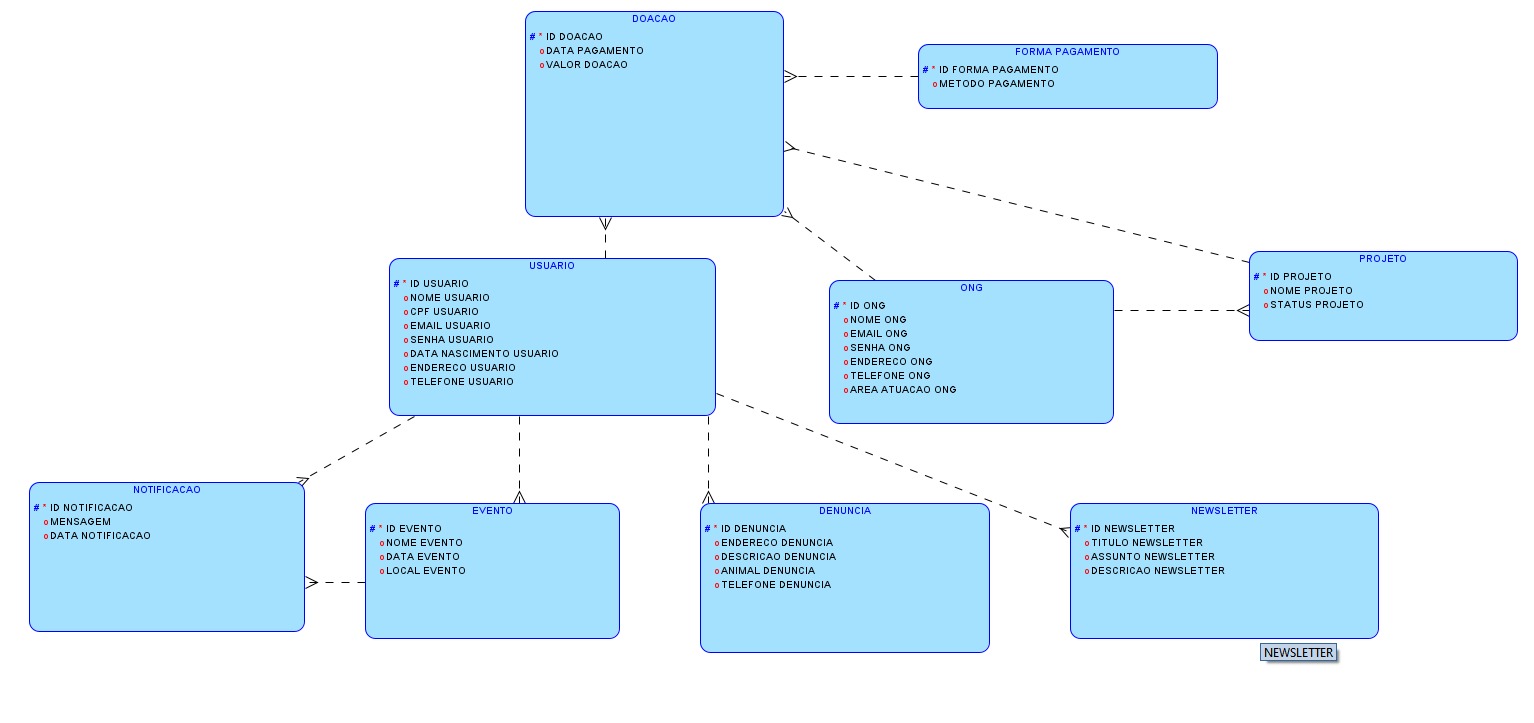
# Interface gráfica do usuário, Aplicativo, Word Descrição gerada automaticamenteInterface gráfica do usuário, Aplicativo Descrição gerada automaticamente

# Interface gráfica do usuário, Aplicativo, Word Descrição gerada automaticamenteInterface gráfica do usuário, Aplicativo Descrição gerada automaticamente

# Interface gráfica do usuário, Aplicativo Descrição gerada automaticamente

# Modelo do Banco de Dados





# Diagramas de Classes

**Diagrama

Descrição gerada automaticamente**

**Diagrama

Descrição gerada automaticamente**

**Diagrama

Descrição gerada automaticamente**

**Diagrama

Descrição gerada automaticamente**

**Diagrama

Descrição gerada automaticamente com confiança média**

**Interface gráfica do usuário

Descrição gerada automaticamente com confiança média**

# Links

Repositório no GitHub: <https://github.com/Luccas-Silva/Fiap_GlobalSolution_Poseidon>