

Projeto **Guinchei**

**Integrantes**:

Augusto Barcelos Barros - **RM98078**

Gabriel Gribl de Carvalho - **RM96270**

Gabriel Souza de Queiroz - **RM98570**

Gabriela Zanotto Alves Rodrigues - **RM551629**

**Sumário:**

1. Descritivo com explicação do projeto, com justificativa/objetivos
2. Modelagem UML das classes

**Objetivo e breve descrição do Projeto:**

Os objetivos primordiais desse projeto incluem:

* Automatizar integralmente o processo de chamados de guincho na Porto Seguro;
* Implementar um sistema de escolha automática do modal de guincho mais adequado para cada chamado;
* Aprimorar a eficiência e a qualidade desse serviço prestado pela Porto Seguro.

A escolha automática do modal de guincho é implementada por meio de um algoritmo inteligente de alocação de recursos. Este algoritmo, desenvolvido em Python, utiliza técnicas avançadas de aprendizado de máquina para considerar fatores como tipo de veículo, local do acidente e condições climáticas. A comunicação entre o algoritmo e o sistema é facilitada por APIs RESTful.

Além do cadastro, o aplicativo pode fornecer ao usuário uma solicitação de serviço de guincho em caso de necessidade. Para isso, ele poderá informar a localização exata do veículo e detalhes do problema ou acidente. Essas informações serão enviadas para o sistema, que vai registrar as informações e fazer um chamado para o serviço de guincho mais próximo da região.

No contexto do projeto, serão desenvolvidas classes como “User” para representar as informações dos usuários cadastrados, e “Casos” para representar as informações relacionadas ao acidente que decidiram na escolha do melhor modal de guincho

O envio dessas informações será feito pelo Site ou aplicativo até o servidor, utilizando-se da comunicação de uma Api Restful Java com Tomcat, sendo os endpoints mostrados abaixo:

**Tabela dos Endpoints:**

**🚘 Usuário**

|  |  |  |
| --- | --- | --- |
| **Método** | **Endpoint** | **Descrição** |
| GET | /api/user | Listar todos os users |
| GET | /api/user/<id> | Buscar user pelo id |
| GET | /api/user | Buscar user pelo nome |
| POST | /api/user | Cadastrar um user |
| PUT | /api/user/<id> | Atualizar um user |
| DELETE | /api/user/<id> | Deletar um user |

**📝 Casos**

|  |  |  |
| --- | --- | --- |
| **Método** | **Endpoint** | **Descrição** |
| GET | /api/casos | Listar todos os casos |
| GET | /api/caso/user/<nome> | Buscar casos pelo nome do usuário |
| GET | /api/casos/<id> | Buscar caso pelo id |
| GET | /api/caso/status/<status> | Buscar casos pelo status |
| POST | /api/casos | Cadastrar um caso |
| PUT | /api/casos/<id> | Atualizar um caso |
| DELETE | /api/casos/<id> | Deletar um caso |

Diagrama

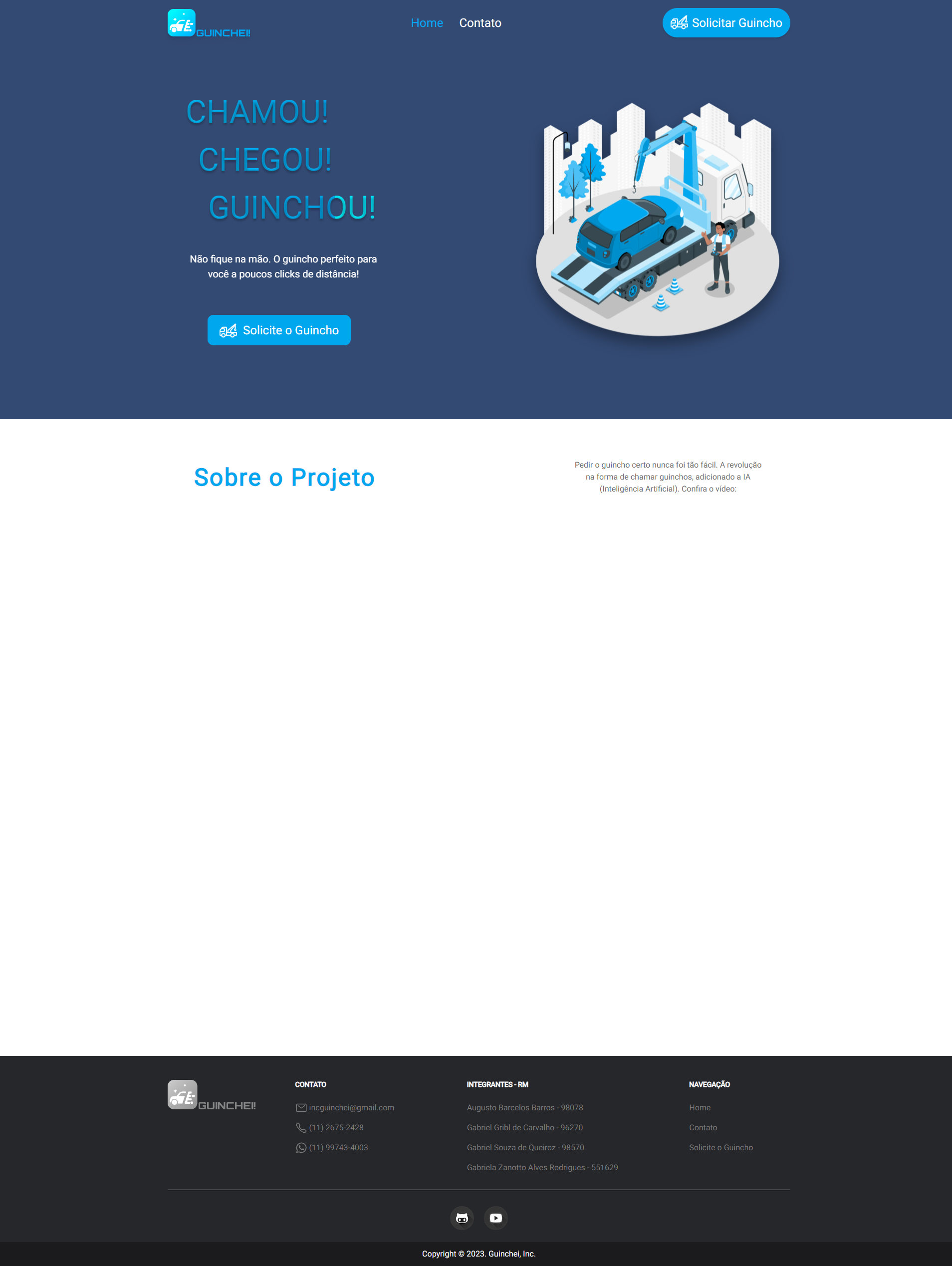
Descrição gerada automaticamente**Modelo necessário para a Api de Java:**

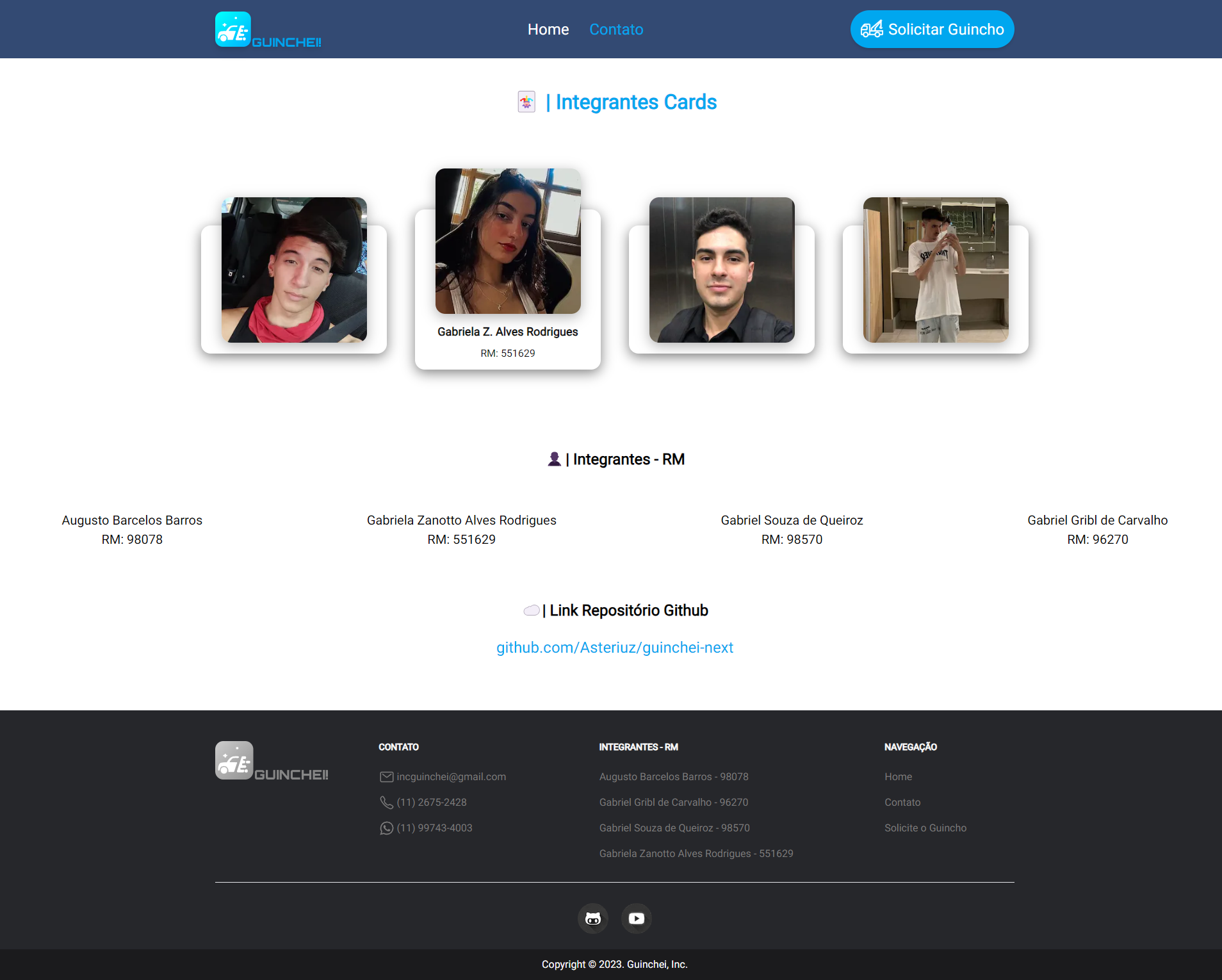
Diagrama

Descrição gerada automaticamente**Modelo do banco de dados:**

**Protótipo (Telas Implementadas)**

**Ícone

Descrição gerada automaticamenteHome:**

**Contato:**

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamenteSolicitar Guincho:**

**Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamenteEnvio do caso recebido pelo Servidor Java:**