

PortoQuest Java

Leonardo Faria Salazar - RM 557484
Jonathan Matheus Rasne Pereira Fonseca - RM 558437
Alexsandro de Macedo de Jesus - RM 557068

Sumário

Descritivo	3
Diagrama UML	4
Telas implementadas	5
Modelo do Banco de Dados	10
Tabela com Endpoints	12

Descritivo

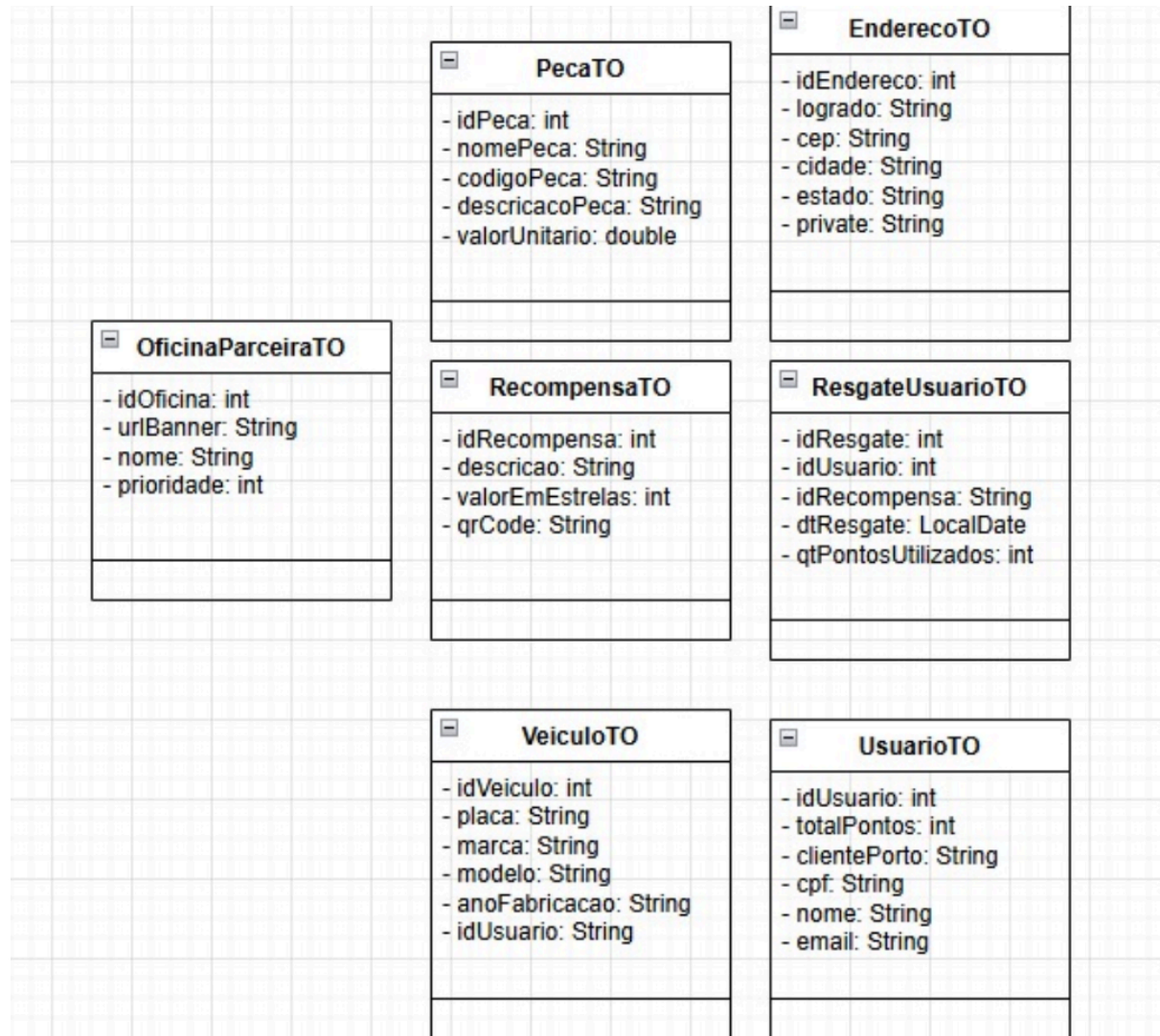
A **PortoQuest** visa criar uma experiência dinâmica e recompensadora para a manutenção de veículos, utilizando a gamificação para incentivar visitas regulares e premiar a lealdade dos usuários. Com um sistema de pontos, manutenções gratuitas, e uma plataforma de promoção para oficinas, oferecemos uma solução completa que beneficia tanto os consumidores quanto os prestadores de serviços. O suporte contínuo através de nosso chatbot e as notificações personalizadas através da IA em nosso banco de dados garantem uma comunicação eficaz e uma experiência de usuário excepcional.

Breve descrição sobre as funcionalidades:

1. **Gamificação para incentivar visitas:** Utiliza técnicas de gamificação para motivar os usuários a visitarem regularmente as oficinas mecânicas, tornando a manutenção do veículo uma experiência mais envolvente e divertida.
2. **Sistema de Pontos e Recompensas:** Os usuários acumulam pontos a cada visita ou serviço realizado, que podem ser trocados por descontos, manutenções gratuitas ou outras recompensas exclusivas.
3. **Promoção de Oficinas Parceiras:** Oferece uma plataforma para oficinas mecânicas promoverem seus serviços, atraindo mais clientes e gerando maior visibilidade no mercado.
4. **Suporte Contínuo via Chatbot:** Um chatbot integrado fornece suporte imediato aos usuários, ajudando com dúvidas, agendamentos e informações sobre serviços e promoções.
5. **Notificações Personalizadas com IA:** Utilizamos inteligência artificial para enviar notificações personalizadas, como lembretes de manutenção, promoções especiais ou mensagens de fidelização, com base no histórico e comportamento dos usuários.

Essas funcionalidades visam melhorar a experiência do usuário, aumentar a fidelização e promover o crescimento das oficinas parceiras.

Diagrama UML



Telas implementadas



SEJA BEM VINDO!

E-MAIL

SENHA

ESQUECI MINHA SENHA

ENTRAR

Tela de login

CADASTRAR-SE

E-MAIL

CPF

SENHA

CONFIRMA SENHA

CADASTRAR

Tela de Cadastro

CONFIRA NOSSOS PARCEIROS:



A PORTOQUEST VISA CRIAR UMA EXPERIÊNCIA DINÂMICA E RECOMPENSADORA PARA A MANUTENÇÃO DE VEÍCULOS, UTILIZANDO A GAMIFICAÇÃO PARA INCENTIVAR VISITAS REGULARES E PREMIAR A LEALDADE DOS USUÁRIOS. COM UM SISTEMA DE PONTOS, MANUTENÇÕES GRATUITAS, E UMA PLATAFORMA DE PROMOÇÃO PARA OFICINAS, OFERECEMOS UMA SOLUÇÃO COMPLETA QUE BENEFICIA TANTO OS CONSUMIDORES QUANTO OS PRESTADORES DE SERVIÇOS. O SUPORTE CONTÍNUO ATRAVÉS DE NOSSO CHATBOT E AS NOTIFICAÇÕES PERSONALIZADAS GARANTEM UMA COMUNICAÇÃO EFICAZ E UMA EXPERIÊNCIA DO USUÁRIO EXCEPCIONAL.

Tela inicial

Utilizada para divulgação de oficinas parceiras

MEUS PONTOS: 1★

DATA DE VIGENCIA: 01/01/2024 A 01/01/2025



Tela Meus Pontos

Utilizada para visualização dos prêmios disponíveis e pontos do usuário, também é possível acessar a aba de “Resgatar” o item

RESGATAR: TROCA DE PASTILHAS DE FREIO




APRESENTE ESSE QR CODE NA OFICINA MECÂNICA

Tela Resgatar Recompensa

Tela de resgate do item selecionado pelo usuário, exibindo o QR Code que deverá ser apresentado pelo usuário na oficina mecânica

SAIR 



LEONARDO SALAZAR
 TOTAL DE PONTOS: 1

[TROCAR SENHA](#)

[HISTÓRICO DE RECOMPENSAS](#)

Tela de Perfil

Utilizada para visualização de dados do usuário

[SAIR](#)

MEUS RESGATES	
RESGATE: TROCA DE PASTILHA DE FREIO DATA: 01/01/2024	2 ★
RESGATE: TROCA DE ÓLEO DATA: 01/04/2024	5 ★
RESGATE: REVISÃO GRATUITA DATA: 01/08/2024	1 ★

Tela de Meus Resgates
Utilizada como histórico para as recompensas do usuário

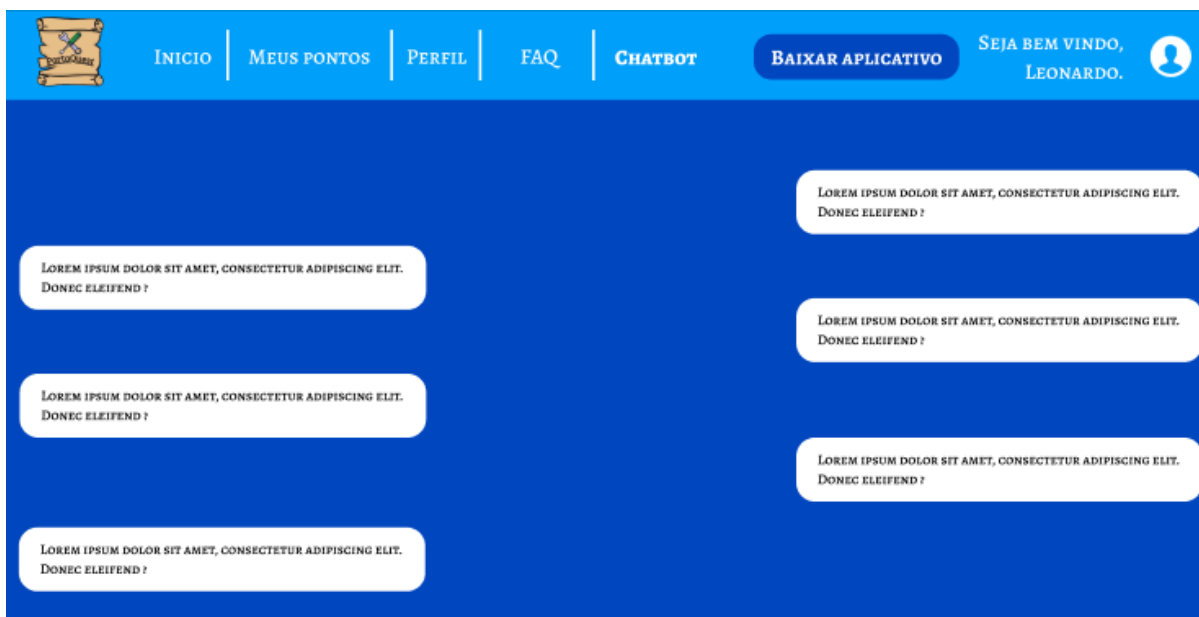
COMO FUNCIONA A PORTO QUEST ?

- LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. DONEC ELEIFEND ?
 
- LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. DONEC ELEIFEND ?
 

LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. DONEC ELEIFEND, ODIO CURSUS ELEMENTUM LACINIA, TORTOR NULLA LAOREET SAPIEN, IN CURSUS URNA PURUS IN NISI. UT IPSUM ORCI, EFFICITUR ET TRISTIQUE AT, SOLLICITUDIN AT FELIS. PRAESENT IMPERDIET TELLUS AC MAXIMUS PORTTITOR. SED QUIS NUNC PHARETRA, CURSUS ODIO VEL, MOLLIS NIBH. VESTIBULUM MALESUADA DUI ID ENIM SOLLICITUDIN ACCUMSAN. NULLA FACILISI. NULLA RUTRUM NULLA ET JUSTO FEUGIAT, AC CONGUE LEO ELEIFEND. IN HAC HABITASSE PLATEA DICTUMST. ALIQUAM ERAT VOLUTPAT. DONEC NIBH DUI, VARIUS IN EGESTAS AC, ULTRICIES EGEGT QUAM.
- LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. DONEC ELEIFEND ?
 
- LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. DONEC ELEIFEND ?
 

Tela de FAQ “Perguntas Frequentes”

Utilizada para perguntas mais frequentes dos usuários sobre como funciona o app, sistema de pontuação, oficinas parceiras, etc

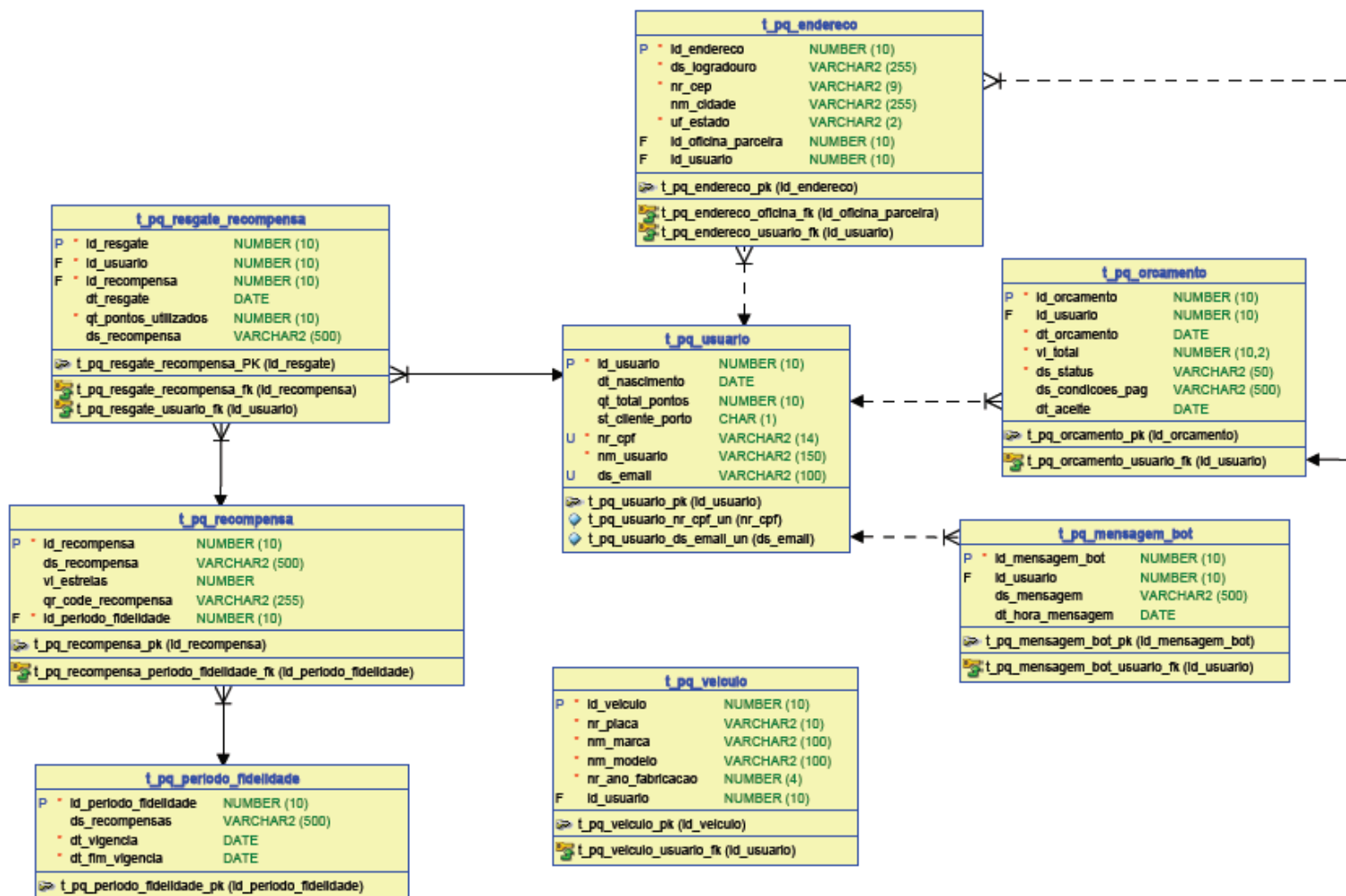


PERGUNTE AO NOSSO CHATBOT:

Tela do Chatbot

ChatBot para auxílio em tempo real sobre as principais dúvidas do usuário em relação ao app

Modelo do Banco de Dados



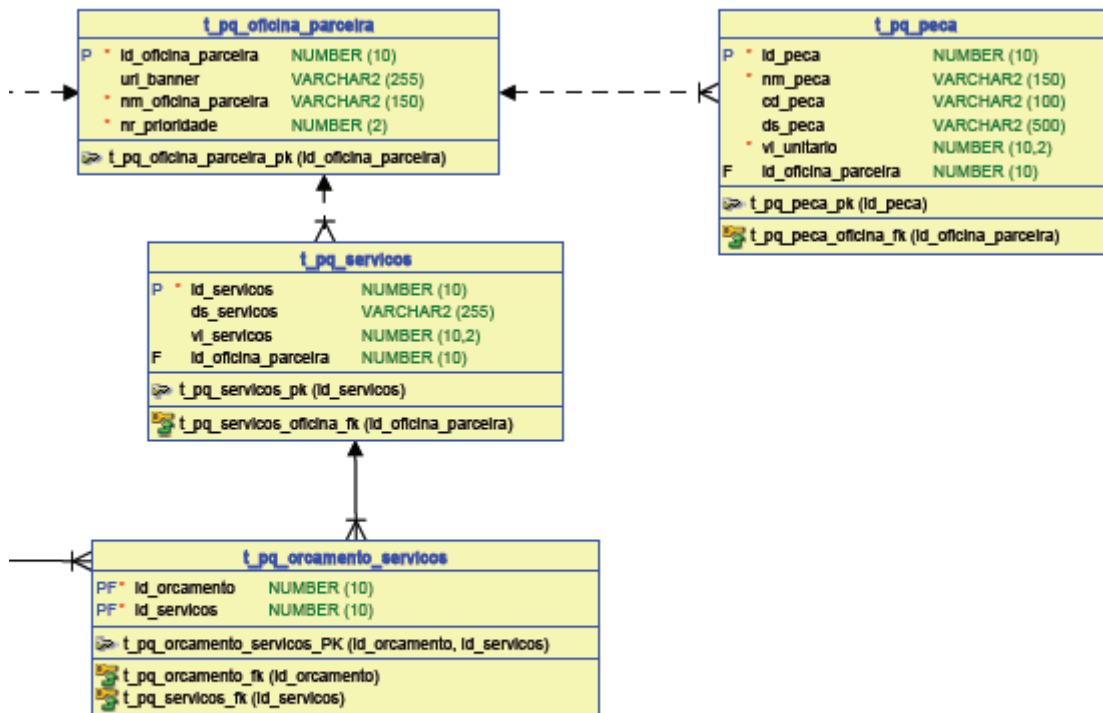


Tabela com os endpoints

Veículo:

Endpoint: `/veiculo`

Método HTTP: GET

Descrição: Retorna todos os veículos cadastrados.

Status Code: Retorna 200 (OK) se houver veículos, ou 404 (Not Found) se não houver.

Endpoint: `/veiculo/{id}`

Método HTTP: GET

Descrição: Retorna um veículo específico pelo seu ID.

Status Code: Retorna 200 (OK) se o veículo for encontrado, ou 404 (Not Found) se o veículo não existir.

Endpoint: `/veiculo`

Método HTTP: POST

Descrição: Adiciona um novo veículo ao sistema.

Status Code: Retorna 201 (Created) se o veículo for adicionado com sucesso, ou 400 (Bad Request) em caso de erro.

Endpoint: `/veiculo/{id}`

Método HTTP: DELETE

Descrição: Exclui um veículo específico pelo seu ID.

Status Code: Retorna 200 (OK) se o veículo for excluído com sucesso, ou 404 (Not Found) se o veículo não for encontrado.

Endpoint: `/veiculo/{id}`

Método HTTP: PUT

Descrição: Atualiza as informações de um veículo específico pelo seu ID.

Status Code: Retorna 200 (OK) se o veículo for atualizado com sucesso, ou 404 (Not Found) se o veículo não for encontrado.

Usuário:

Endpoint: `/usuario`

Método HTTP: GET

Descrição: Retorna todos os usuários cadastrados.

Status Code: Retorna 200 (OK) se houver usuários, ou 404 (Not Found) se não houver.

Endpoint: `/usuario/{id}`

Método HTTP: GET

Descrição: Retorna um usuário específico pelo seu ID.

Status Code: Retorna 200 (OK) se o usuário for encontrado, ou 404 (Not Found) se o usuário não existir.

Endpoint: `/usuario`

Método HTTP: POST

Descrição: Adiciona um novo usuário ao sistema.

Status Code: Retorna 201 (Created) se o usuário for adicionado com sucesso, ou 400 (Bad Request) em caso de erro.

Endpoint: `/usuario/{id}`

Método HTTP: DELETE

Descrição: Exclui um usuário específico pelo seu ID.

Status Code: Retorna 200 (OK) se o usuário for excluído com sucesso, ou 404 (Not Found) se o usuário não for encontrado.

Endpoint: `/usuario/{id}`

Método HTTP: PUT

Descrição: Atualiza as informações de um usuário específico pelo seu ID.

Status Code: Retorna 200 (OK) se o usuário for atualizado com sucesso, ou 404 (Not Found) se o usuário não for encontrado.

Resgate Usuário:

Endpoint: `/resgate-usuario`

Método HTTP: GET

Descrição: Retorna todos os registros de resgate de usuários.

Status Code: Retorna 200 (OK) se houver registros, ou 404 (Not Found) se não houver.

Endpoint: `/resgate-usuario/{id}`

Método HTTP: GET

Descrição: Retorna um registro de resgate de usuário específico pelo seu ID.

Status Code: Retorna 200 (OK) se o registro for encontrado, ou 404 (Not Found) se o registro não existir.

Endpoint: `/resgate-usuario`

Método HTTP: POST

Descrição: Adiciona um novo registro de resgate de usuário ao sistema.

Status Code: Retorna 201 (Created) se o registro for adicionado com sucesso, ou 400 (Bad Request) em caso de erro.

Endpoint: `/resgate-usuario/{id}`

Método HTTP: DELETE

Descrição: Exclui um registro de resgate de usuário específico pelo seu ID.

Status Code: Retorna 200 (OK) se o registro for excluído com sucesso, ou 404 (Not Found) se o registro não for encontrado.

Endpoint: `/resgate-usuario/{id}`

Método HTTP: PUT

Descrição: Atualiza as informações de um registro de resgate de usuário específico pelo seu ID.

Status Code: Retorna 200 (OK) se o registro for atualizado com sucesso, ou 404 (Not Found) se o registro não for encontrado.

Recompensa:

Endpoint: `/recompensas`

Método HTTP: GET

Descrição: Retorna todas as recompensas cadastradas.

Status Code: Retorna 200 (OK) se houver recompensas, ou 404 (Not Found) se não houver.

Endpoint: `/recompensas/{id}`

Método HTTP: GET

Descrição: Retorna uma recompensa específica pelo seu ID.

Status Code: Retorna 200 (OK) se a recompensa for encontrada, ou 404 (Not Found) se a recompensa não existir.

Endpoint: `/recompensas`

Método HTTP: POST

Descrição: Adiciona uma nova recompensa ao sistema.

Status Code: Retorna 201 (Created) se a recompensa for adicionada com sucesso, ou 400 (Bad Request) em caso de erro.

Endpoint: `/recompensas/{id}`

Método HTTP: DELETE

Descrição: Exclui uma recompensa específica pelo seu ID.

Status Code: Retorna 204 (No Content) se a recompensa for excluída com sucesso, ou 404 (Not Found) se a recompensa não for encontrada.

Endpoint: `/recompensas/{id}`

Método HTTP: PUT

Descrição: Atualiza as informações de uma recompensa específica pelo seu ID.

Status Code: Retorna 201 (Created) se a recompensa for atualizada com sucesso, ou 400 (Bad Request) em caso de erro.

Peca:

Endpoint: `/peca`

Método HTTP: GET

Descrição: Retorna todas as peças cadastradas.

Status Code: Retorna 200 (OK) se houver peças, ou 404 (Not Found) se não houver.

Endpoint: `/peca/{id}`

Método HTTP: GET

Descrição: Retorna uma peça específica pelo seu ID.

Status Code: Retorna 200 (OK) se a peça for encontrada, ou 404 (Not Found) se a peça não existir.

Endpoint: `/peca`

Método HTTP: POST

Descrição: Adiciona uma nova peça ao sistema.

Status Code: Retorna 201 (Created) se a peça for adicionada com sucesso, ou 400 (Bad Request) em caso de erro.

Endpoint: `/peca/{id}`

Método HTTP: DELETE

Descrição: Exclui uma peça específica pelo seu ID.

Status Code: Retorna 200 (OK) se a peça for excluída com sucesso, ou 404 (Not Found) se a peça não for encontrada.

Endpoint: `/peca/{id}`

Método HTTP: PUT

Descrição: Atualiza as informações de uma peça específica pelo seu ID.

Status Code: Retorna 200 (OK) se a peça for atualizada com sucesso, ou 404 (Not Found) se a peça não for encontrada.

Oficina Parceira:

Endpoint: `/oficinas`

Método HTTP: GET

Descrição: Retorna todas as oficinas parceiras cadastradas.

Status Code: Retorna 200 (OK) se houver oficinas cadastradas, ou 404 (Not Found) se não houver.

Endpoint: `/oficinas/{id}`

Método HTTP: GET

Descrição: Retorna uma oficina parceira específica pelo seu ID.

Status Code: Retorna 200 (OK) se a oficina for encontrada, ou 404 (Not Found) se a oficina não existir.

Endpoint: `/oficinas`

Método HTTP: POST

Descrição: Adiciona uma nova oficina parceira ao sistema.

Status Code: Retorna 201 (Created) se a oficina for adicionada com sucesso, ou 400 (Bad Request) em caso de erro.

Endpoint: `/oficinas/{id}`

Método HTTP: DELETE

Descrição: Exclui uma oficina parceira específica pelo seu ID.

Status Code: Retorna 204 (No Content) se a oficina for excluída com sucesso, ou 404 (Not Found) se a oficina não for encontrada.

Endpoint: `/oficinas/{id}`

Método HTTP: PUT

Descrição: Atualiza as informações de uma oficina parceira específica pelo seu ID.

Status Code: Retorna 201 (Created) se a oficina for atualizada com sucesso, ou 400 (Bad Request) em caso de erro.

Esse formato fornece uma visão clara e organizada de cada endpoint da classe

`OficinaParceiraResource`