

**Site Integrado com gameificação para melhorar vivência em
condomínio**

Arthur Eduardo Luna Pulini - RM 554848

Lucas Almeida Fernandes de Moraes - RM 557569

Victor Nascimento Cosme - RM 55885

Sumário

Descritivo: 3

Objetivo: 3

Descritivo das classes:..... 4

 Classes Domínio: 4

 Classes DAO: 5

 Classe Service: 6

 Classe Controller: 7

Protótipos de telas:..... 8

DER (Diagrama Entidade-Relacionamento):..... 12

Diagrama de Classe: 13

Requisições:..... 15

Descritivo:

O GreenMeter é uma plataforma patenteada e desenvolvida pela QuantumLeap para o Condomínio Nascer do Sol para que os moradores possam cadastrar seus dados no sistema para que então possam registrar suas contas de energia mês a mês de modo que o consumo de energia seja utilizado como métrica para gerar um ranking de moradores ordenado pelo menor consumo e, ao final de cada mês, os melhores colocados no ranking ganham premiações no condomínio, como descontos, bônus etc. A principal vantagem da solução é instigar a economia de energia pelos moradores de forma lúdica.

Objetivo:

O GreenMeter é uma plataforma desenvolvida pela QuantumLeap para o Condomínio Nascer do Sol, com o objetivo de promover a economia de energia entre os moradores de forma lúdica e engajante. A plataforma permite que os moradores registrem mensalmente suas contas de energia, utilizando esses dados para gerar um ranking baseado no menor consumo. Ao final de cada mês, os moradores melhor colocados são recompensados com benefícios, como descontos e bônus no condomínio, incentivando hábitos sustentáveis e a redução do consumo energético.

Descritivo das classes:

Classes Domínio:

- Apartamento: Os atributos são, numeroApartamento, idMorador, idCondominio que respectivamente armazena o identificador do Apartamento.
- Condominio: Os atributos são, contaCondominio que serve para instanciar uma classe Condominio.
- ConsumoMorador: Os atributos são, nomeMorador, numeroApartamento, consumoTotalKwh que respectivamente armazena o identificador do ConsumoMorador utilizado mais para um tipo na aplicação web.
- ContaDeEnergia: Os atributos são, valorConta, dataConta, consumoKwh, idApartamento que respectivamente armazena o identificador do ContaDeEnergia, e também tem uma função chamada verificaCarro que recebe um List de veículos para verificar se o carro é elétrico assim adicionando na conta do morador um consumo.
- Desconto: Os atributos são, valor desconto, descricaoDesconto, dataDesconto, idApartamento que serve para instanciar uma classe Desconto usada mais para fazer um histórico de descontos.
- Morador: Os atributos são, nomeMorador, cpf, email, telefone, que respectivamente armazena o identificador do Morador.
- Veiculos: Os atributos são, placaVeiculo, anoVeiculo, isEletrico, idApartamento, que respectivamente armazena o identificador do Veiculo.

Classes DAO:

- ApartamentoDAO: Classe que faz conexão com o banco de dados, tem como objetivo realizar inserções, consultas, atualizações e exclusões no banco, com foco na classe Apartamento.
- CondominioDAO: Classe que faz conexão com o banco de dados, tem como objetivo realizar inserções, consultas, atualizações e exclusões no banco, com foco na classe Consominio.
- ConsumoMoradorDAO: Classe que faz conexão com o banco de dados, tem como objetivo realizar inserções, consultas, atualizações e exclusões no banco, com foco na classe ConsumoMorador.
- ContaDeEnergiaDAO: Classe que faz conexão com o banco de dados, tem como objetivo realizar inserções, consultas, atualizações e exclusões no banco, com foco na classe ContaDeEnergia.
- ConnectionFactory: Classe de configuração de credenciais do banco.
- DescontoDAO: Classe que faz conexão com o banco de dados, tem como objetivo realizar inserções, consultas, atualizações e exclusões no banco, com foco na classe Desconto.
- MoradorDAO: Classe que faz conexão com o banco de dados, tem como objetivo realizar inserções, consultas, atualizações e exclusões no banco, com foco na classe Morador.
- VeiculosDAO: Classe que faz conexão com o banco de dados, tem como objetivo realizar inserções, consultas, atualizações e exclusões no banco, com foco na classe Veiculos.

Classe Service:

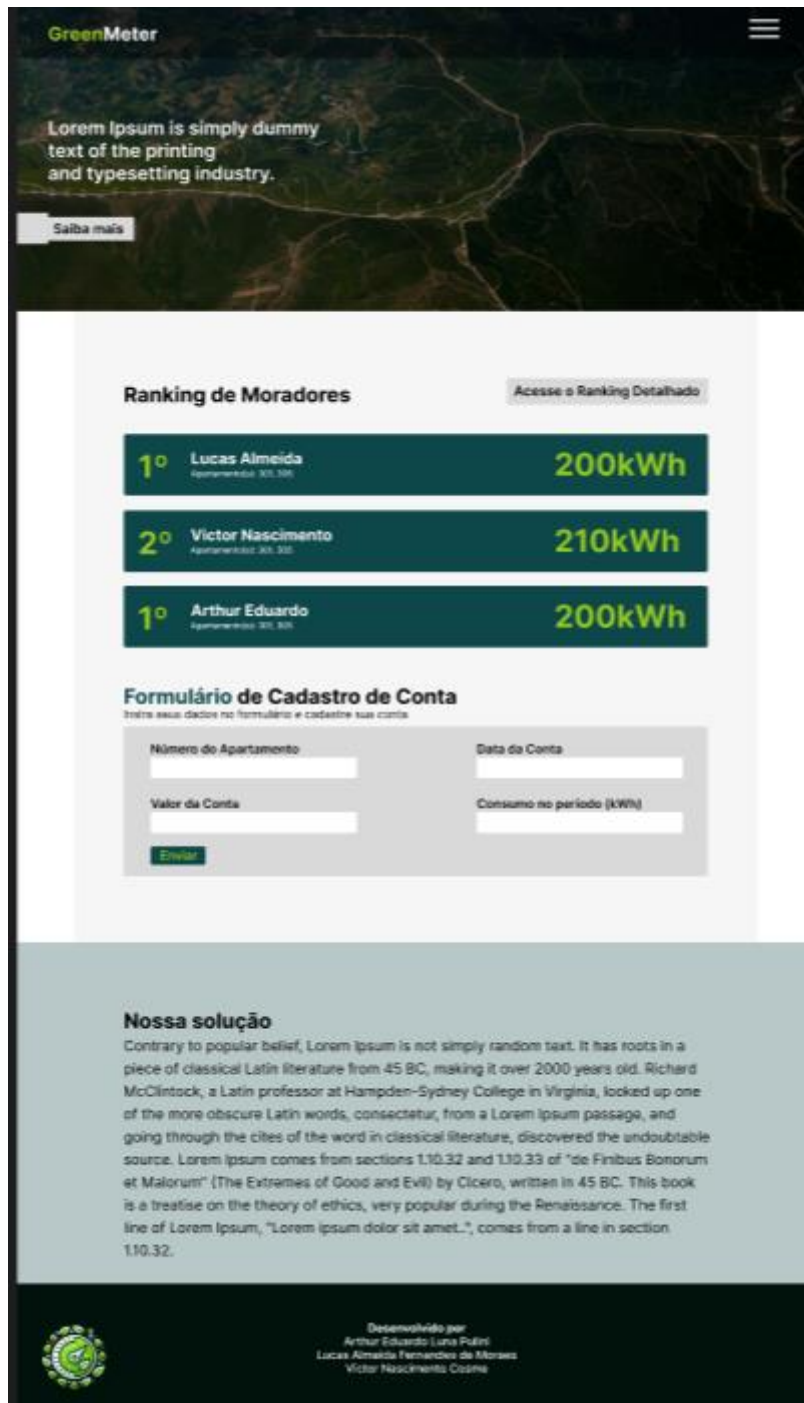
- ApartamentoService: Classe que faz o agrupamento da lógica de negocios a classe DAO, para fazer todo o processamento de dados e que intermedia a comunicação com a classe Controller.
- CondominioService: Classe que faz o agrupamento da lógica de negocios a classe DAO, para fazer todo o processamento de dados e que intermedia a comunicação com a classe Controller.
- ConsumoMoradorService: Classe que faz o agrupamento da lógica de negocios a classe DAO, para fazer todo o processamento de dados e que intermedia a comunicação com a classe Controller.
- CotaDeEnergiaService: Classe que faz o agrupamento da lógica de negocios a classe DAO, para fazer todo o processamento de dados e que intermedia a comunicação com a classe Controller.
- DescontoService: Classe que faz o agrupamento da lógica de negocios a classe DAO, para fazer todo o processamento de dados e que intermedia a comunicação com a classe Controller.
- MoradorService: Classe que faz o agrupamento da lógica de negocios a classe DAO, para fazer todo o processamento de dados e que intermedia a comunicação com a classe Controller.
- VeiculosService: Classe que faz o agrupamento da lógica de negocios a classe DAO, para fazer todo o processamento de dados e que intermedia a comunicação com a classe Controller.

Classe Controller:

- ApartamentoController: Classe que serve para fazer a utilização das demais classes através da classe Service e tem como função a criação dos endpoints para ser usado em outros lugares.
- CondominioController: Classe que serve para fazer a utilização das demais classes através da classe Service e tem como função a criação dos endpoints para ser usado em outros lugares.
- ConsumoMoradorController: Classe que serve para fazer a utilização das demais classes através da classe Service e tem como função a criação dos endpoints para ser usado em outros lugares.
- CotaDeEnergiaController: Classe que serve para fazer a utilização das demais classes através da classe Service e tem como função a criação dos endpoints para ser usado em outros lugares.
- DescontoController: Classe que serve para fazer a utilização das demais classes através da classe Service e tem como função a criação dos endpoints para ser usado em outros lugares.
- MoradorController: Classe que serve para fazer a utilização das demais classes através da classe Service e tem como função a criação dos endpoints para ser usado em outros lugares.
- VeiculosController: Classe que serve para fazer a utilização das demais classes através da classe Service e tem como função a criação dos endpoints para ser usado em outros lugares.

Protótipos de telas:

Tela inicial onde será utilizada uma API para listar o ranking:



Página dos integrantes:


GreenMeter

Menu icon

Lorem Ipsum is simply dummy text of the printing and typesetting industry.



Saiba mais


Integrantes do Grupo



Arthur Eduardo Luna Pulini



RM 554848






Lucas Almeida Fernandes de Moraes



RM 557569






Victor Nascimento Cosme

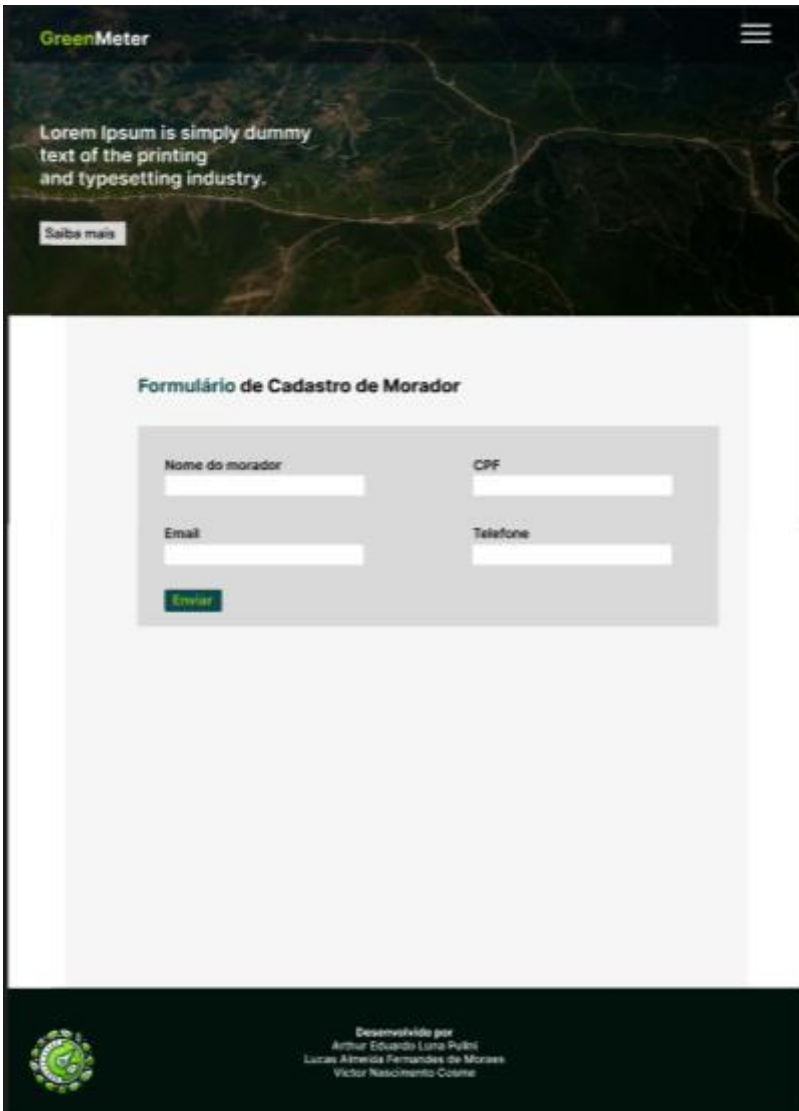
RM 558856





Desenvolvido por
Arthur Eduardo Luna Pulini
Lucas Almeida Fernandes de Moraes
Victor Nascimento Cosme

Página de cadastro, onde serão cadastradas as contas:



The image shows a web page for 'GreenMeter' with a dark green background featuring a satellite map of a forested area. The page includes a header with the 'GreenMeter' logo and a hamburger menu icon. Below the header, there is a placeholder text 'Lorem Ipsum is simply dummy text of the printing and typesetting industry.' and a 'Saiba mais' button. The main content area is a light gray box titled 'Formulário de Cadastro de Morador'. It contains four input fields: 'Nome do morador', 'CPF', 'Email', and 'Telefone'. A green 'Enviar' button is positioned below the 'Email' field. The footer is dark green and contains a circular logo on the left and the text 'Desenvolvido por Arthur Eduardo Lima Pulini, Lucas Almeida Fernandes de Moraes, Victor Nascimento Colina' on the right.

GreenMeter

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Saiba mais

Formulário de Cadastro de Morador

Nome do morador

CPF

Email

Telefone

Enviar

Desenvolvido por
Arthur Eduardo Lima Pulini
Lucas Almeida Fernandes de Moraes
Victor Nascimento Colina

Página de consulta de contas por CPF onde aparecerá um histórico de contas onde elas poderão ser alteradas ou excluídas:

GreenMeter

Lorem Ipsum is simply dummy text of the printing and typesetting industry.

Saiba mais

Busque suas Contas de Energia

Insira seu CPF abaixo e busque todas as contas de energia associadas ao seu CPF.

CPF

Enviar

Data da Conta	Valor da Conta	Consumo de Energia (kWh)	Editar/Excluir
01/jan/24	R\$ 180,00	50	Editar/Excluir
01/fev/24	R\$ 150,00	20	Editar/Excluir
01/mar/24	R\$ 55,00	40	Editar/Excluir
01/abr/24	R\$ 91,00	35	Editar/Excluir
01/mai/24	R\$ 180,00	95	Editar/Excluir
01/jun/24	R\$ 99,00	100	Editar/Excluir
01/jul/24	R\$ 200,00	23	Editar/Excluir
01/ago/24	R\$ 135,00	82	Editar/Excluir
01/set/24	R\$ 175,00	33	Editar/Excluir
01/out/24	R\$ 200,00	75	Editar/Excluir

Desenvolvido por

Arthur Eduardo Luna Pulini

Lucas Almeida Fernandes de Moraes

Victor Nascimento Cosme

DER (Diagrama Entidade-Relacionamento):

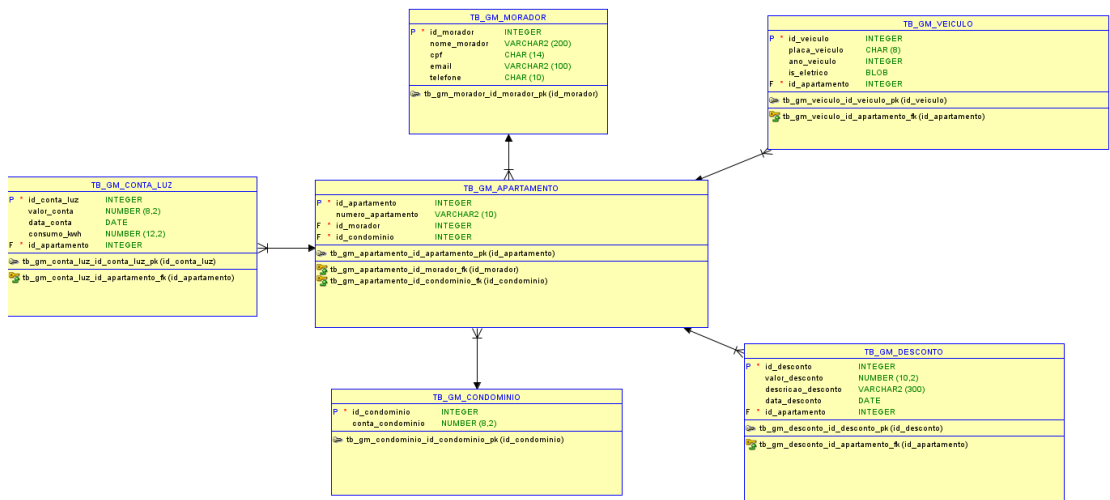
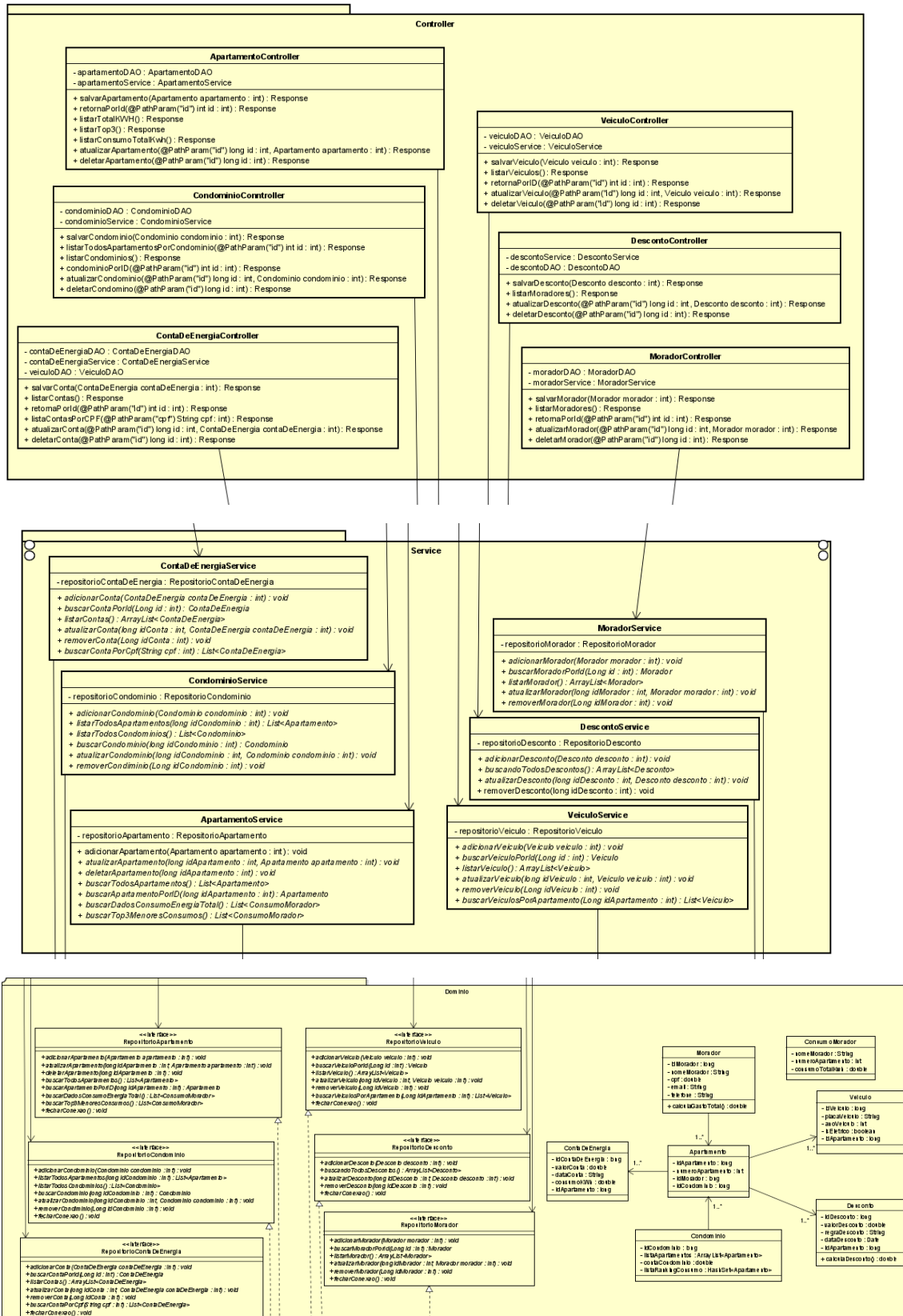
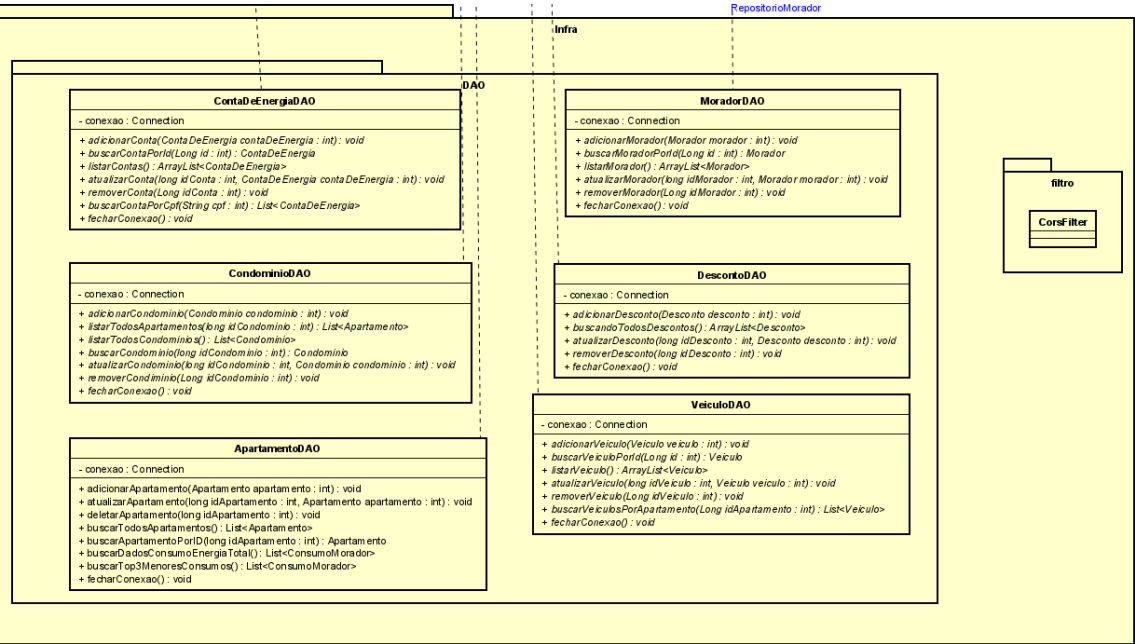


Diagrama de Classe:





Requisições:

"apartamentos"	POST	201
"apartamentos/{id}"	GET	200
"apartamentos/consumoKWH"	GET	200
"apartamentos/menoresGastos"	GET	200
"apartamentos"	GET	200
"apartamentos/{id}"	PUT	200
"apartamentos/{id}"	DELETE	200
"contominios"	POST	201
"contominios/apartamentoPorCondominio/{id}"	GET	200
"contominios"	GET	200
"contominios/{id}"	GET	200
"contominios/{id}"	PUT	200
"contominios/{id}"	DELETE	200
"contas"	POST	201
"contas"	GET	200
"contas/{id}"	GET	200
"contas/buscaPorCPF/{cpf}"	GET	200
"contas/{id}"	PUT	200
"contas/{id}"	DELETE	200
"descontos"	POST	201
"descontos"	GET	200
"descontos/{id}"	PUT	200
"descontos/{id}"	DELETE	200
"moradores"	POST	201
"moradores"	GET	200
"moradores/{id}"	GET	200
"moradores/{id}"	PUT	200
"moradores/{id}"	DELETE	200
"veiculos"	POST	201
"veiculos"	GET	200
"veiculos/{id}"	GET	200
"veiculos/{id}"	PUT	200
"veiculos/{id}"	DELETE	200