

CycleSecure

Domain Driven Design

Camila Soares Pedra RM 98246

Rafael da Silva Camargo RM 551127

GUSTAVO BERTTI RM 552243

GUSTAVO MACEDO DA SILVA RM 552333

Sumário

1. Explicação do projeto e funcionalidades	03
2. Modelagem das classes	04
3. Protótipo.1	05
4. Protótipo.2	06
5. Protótipo.3	07
6. Protótipo.4	08
7. Procedimentos para rodar a aplicação.....	09
8. Tabela dos endpoints (API Restful).....	09, 10
9. Modelo do banco de dados	11

Explicação do projeto:

Nossa solução origina-se da análise do atual processo de vistoria feito pela mão de obra humana. Com a ideia de fazer uma transformação digital nesse processo, nosso projeto traz uma automatização para isso, reduzindo custos e possíveis falhas humanas.

A solução possui 6 classes models sendo elas:

- Pessoa: Classe abstrata;
- Cliente: Herda da classe abstrata Pessoa;
- Colaborador Herda da classe abstrata Pessoa;
- Bicicleta;
- Seguro;
- Sinistro;

No Pacote dao contém a DAO de cada classe model com os métodos executando a query do banco.

No pacote services temos uma classe cada classe acima, exceto a classe abstrata(Pessoa)

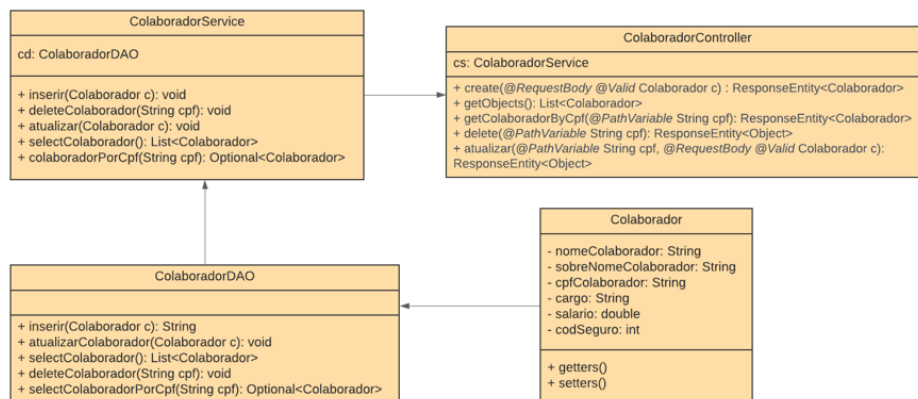
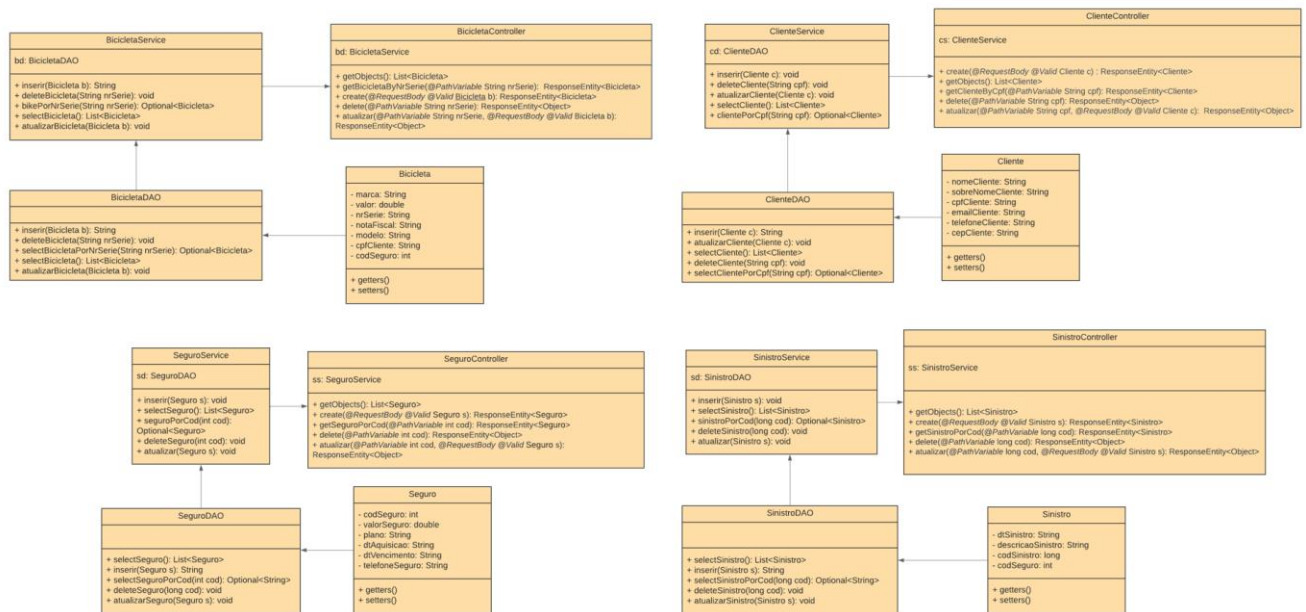
Contendo em cada uma os métodos que chamam o insert, delete, update, select e select por primary key de cada classe model.

Por último no pacote Controller é onde a API é criada para requisição do front

Principais funcionalidades são responsáveis pelas classes abaixo:

- SinistroController: Responsável pelo relato imediato do sinistro ocorrido e já guardando as informações referentes a descrição e data.
- ClienteController, BicicletaController, ColaboradorController: Responsável pelo cadastro das classes (Cliente, Colaborador, Bicicleta). Armazena informações referente ao sinistro e gera um código único que será utilizado como identificador no momento de acessar esse sinistro.

Modelagem das classes:



Protótipo:


Tela inicial - Contém botão para realizar cadastro da bike e ao canto superior direito botão para acessar a área do cliente.

CycleSecure

Área do cliente

Além do carro,
cuidando da sua bike
com soluções completas.

Nosso seguro tem a oferecer assistência 24h,
revisões remotas de forma intuitiva, rápida e segura,
apenas com algumas fotos da bicicleta.



Realizar cadastro

Tela Cadastro de bike – Ao clicar o botão ‘Realizar cadastro’ na tela inicial é direcionado à esta tela de cadastro com campos de preenchimento obrigatório.

CycleSecure

Digite marca da bicicleta:

Digite valor da bicicleta:


Digite número de série:

Digite nota fiscal da bicicleta:

Digite modelo da bicicleta:

Finalizar

Área do cliente - tela de login para acessar sua conta caso já tenha cadastro, do contrário clicar no botão 'Ir para o cadastro' para realizar o mesmo.



CycleSecure

Acessar Conta:

CPF ou CNPJ:

Senha:

esqueceu sua senha?

[Ir para o cadastro](#)

Entrar

Cadastro do cliente - Tela para realizar o cadastro do cliente.

CycleSecure

Digite seu nome:


Digite seu sobrenome:

Digite seu CPF:

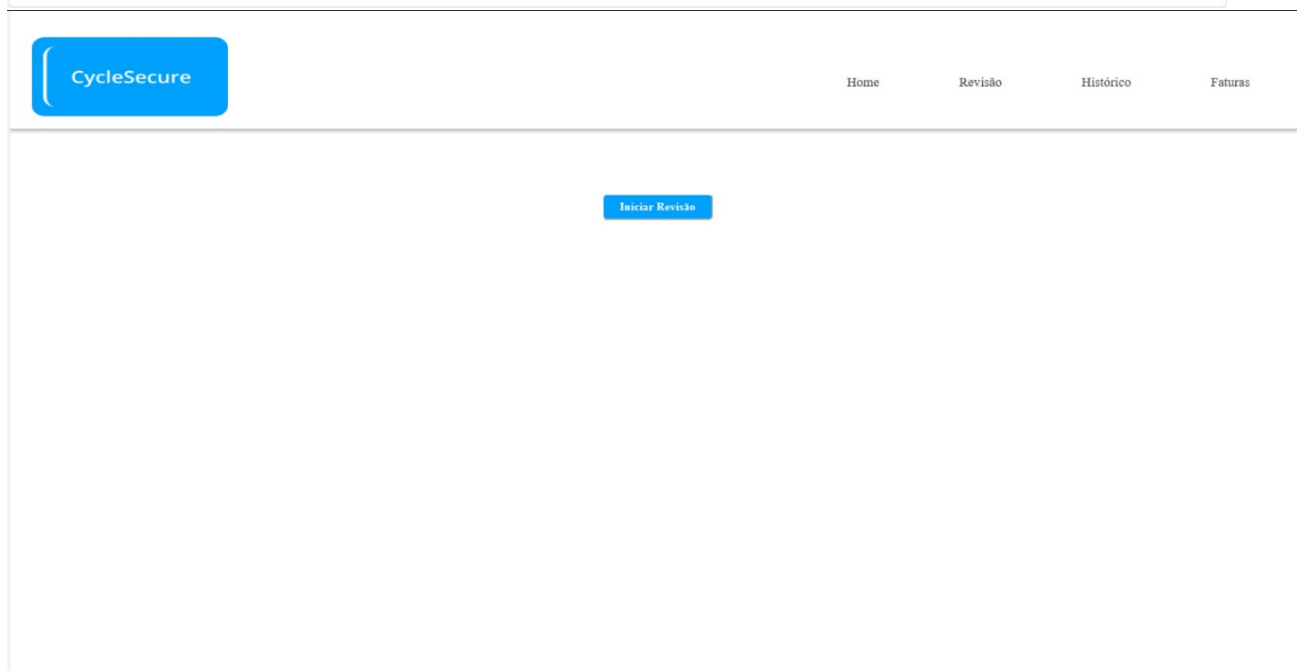
Digite seu e-mail:

Digite seu CEP:

Digite seu telefone:



Área do cliente – Possui o botão para upar uma foto e iniciar o processo de revisão



Histórico - Após a revisão é possível ver o resultado da mesma na aba histórico

CycleSecure

RevisãoHistóricoFaturas

Data: 26/07/2023
Modelo: Caloi
Número de série: 12345
Conclusão: Bicicleta não precisa de manutenção

✓

Faturas - Permite ver as faturas referentes as revisões e manutenções feitas na bike.

CycleSecure

RevisãoHistóricoFaturas

Data de emissão: 5/07/2023
Data de vencimento: 31/07/2023
Valor: R\$ 300,00
Número de código de barra: 123456789102123131
Status: Pagamento não realizado

Procedimentos para rodar a aplicação

1.Rodar o Sprint4Application(main)

URIs, verbo HTTP e Códigos de status de respostas

CLIENTE

Nome	Path	Verbo	Ação	Código de status
create	/api/cliente	POST	Cria um novo cliente	201 Created
getObjects	/api/cliente	GET	Lista todos os clientes	200 OK
atualizar	/api/cliente/{cpf}	PUT	Atualiza um cliente	200 OK
getClienteByCpf	/api/cliente/{cpf}	GET	Mostra dados de um cliente	200 OK
delete	/api/cliente/{cpf}	DELETE	Deleta um cliente	204 No Content

COLABORADOR

Nome	Path	Verbo	Ação	Código de status
create	/api/ colaborador	POST	Cria um novo colaborador	201 Created
getObjects	/api/ colaborador	GET	Lista todos os colaboradores	200 OK
atualizar	/api/ colaborador /{cpf}	PUT	Atualiza um colaborador	200 OK
getColaborador ByCpf	/api/ colaborador /{cpf}	GET	Mostra dados de um colaborador	200 OK
delete	/api/ colaborador /{cpf}	DELETE	Deleta um colaborador	204 No Content

SINISTRO

Nome	Path	Verbo	Ação	Código de status
create	/api/ sinistro	POST	Cria um novo sinistro	201 Created
getObjects	/api/ sinistro	GET	Lista todos os sinistros	200 OK
atualizar	/api/ sinistro /{cod}	PUT	Atualiza um sinistro	200 OK
getSinistroPorCod	/api/ sinistro /{cod}	GET	Mostra dados de um sinistro	200 OK
delete	/api/ sinistro /{cod}	DELETE	Deleta um sinistro	204 No Content

SEGURO

Nome	Path	Verbo	Ação	Código de status
------	------	-------	------	------------------

create	/api/ seguro	POST	Cria um novo seguro	201 Created
getObjects	/api/ seguro	GET	Lista todos os seguros	200 OK
atualizar	/api/ seguro /{cod}	PUT	Atualiza um seguro	200 OK
getSeguroPorCod	/api/ seguro /{cod}	GET	Mostra dados de um seguro	200 OK
delete	/api/ seguro /{cod}	DELETE	Deleta um seguro	204 No Content

BICICLETA

Nome	Path	Verbo	Ação	Código de status
create	/api/ bike	POST	Cria uma nova bicicleta	201 Created
getObjects	/api/ bike	GET	Lista todas as bicicletas	200 OK
atualizar	/api/ bike /{nrSerie}	PUT	Atualiza uma bicicleta	200 OK
getBicicletaByNrSerie	/api/ bike /{nrSerie}	GET	Mostra dados de uma bicicleta	200 OK
delete	/api/ bike /{nrSerie}	DELETE	Deleta uma bicicleta	204 No Content

MODELO DO BANCO DE DADOS

