

Diagrama de Classes

Main

entities

Ciente

- int id;
- String nome;
- String sobrenome;
- LocalDate dataNascimento;
- String telefone;
- Endereco endereco;
- String email
- String usuario;
-String senha;

+ isDateValid();
+ isMaiorDeldade();

_BaseEntity

- int id;

Endereco

- int id;
- String cep;
- String logradouro;
- String numero;
- String complemento;
- String bairro;
- String cidade;
- String estado;
- String pais;
- Cliente cliente

Produto

- int id;
- String nome;
- String descricao;
- double preco;
- int estoque;
- Cliente cliente;

+ verificarCobertura(String procedimento)

connection

DatabaseConnection

- String URL;
- String USER;
- String PASSWORD;

+ java.sql.Connection getConnection()

ResetDatabase

DatabaseConnection connection;

resources

ClienteResource

- Log4jLogger<Cliente> logger = new Log4jLogger<>(Cliente.class);
- ClienteRepository clienteRepository;

+ List<Cliente> getAllClientes()
+ Response createCliente(Cliente cliente)
+ Response updateCliente(@PathParam("id") int id, Cliente cliente)
+ Response deleteCliente(@PathParam("id") int id)
+ Response getCliente(@PathParam("id") int id)

EnderecoResource

- Log4jLogger<Endereco> logger = new Log4jLogger<>(Endereco.class);
- EnderecoRepository enderecoRepository;

+ List<Endereco> getAllEnderecos()
+ Response createEndereco(Endereco endereco)
+ Response updateEndereco(@PathParam("id") int id, Endereco endereco)
+ Response deleteEndereco(@PathParam("id") int id)
+ Response getEndereco(@PathParam("id") int id)

ProdutoResource

- Log4jLogger<Produto> logger = new Log4jLogger<>(Produto.class);
- ProdutoRepository produtoRepository;

+ List<Produto> getAllProdutos()
+ Response createProduto(Produto produto)
+ Response updateProduto(@PathParam("id") int id, Produto produto)
+ Response deleteProduto(@PathParam("id") int id)
+ Response getProduto(@PathParam("id") int id)

repositories

<<interface>>
_BaseRepository <T extends _BaseEntity>

+ void Create(T entity);
+ List<T> ReadAll();
+ boolean DeleteById(int id);
+ boolean UpdateById(int id);
+ T ReadById(int id);

_BaseRepositoryImpl <T extends _BaseEntity>
implements _BaseRepository<T>

Log4jLogger<T> logger

+ void Create(T entity);
+ List<T> ReadAll();
+ boolean DeleteById(int id);
+ boolean UpdateById(int id);
+ T ReadById(int id);

ProdutoRepository extends
_BaseRepositoryImpl<Produto>

DatabaseConnection connection;
String TB_NAME;
Map<String, String> TB_COLUMNS;

- Initialize();
+ void Create(Produto entity);
+ List<Produto> ReadAll();
+ boolean DeleteById(int id);
+ boolean UpdateById(int id);
+ Produto ReadById(int id);

ClienteRepository extends
_BaseRepositoryImpl<Cliente>

DatabaseConnection connection;
String TB_NAME;
Map<String, String> TB_COLUMNS;

- Initialize();
+ void Create(Cliente entity);
+ List<Cliente> ReadAll();
+ boolean DeleteById(int id);
+ boolean UpdateById(int id);
+ Cliente ReadById(int id);

EnderecoRepository extends
_BaseRepositoryImpl<Endereco>

DatabaseConnection connection;
String TB_NAME;
Map<String, String> TB_COLUMNS;

- Initialize();
+ void Create(Endereco entity);
+ List<Endereco> ReadAll();
+ boolean DeleteById(int id);
+ boolean UpdateById(int id);
+ Endereco ReadById(int id);

Test

Test

+ main (String [])

MyResourceTest

- HttpServer server
- WebTarget target

+ setUp();
+ tearDown();
+ testGetIt();