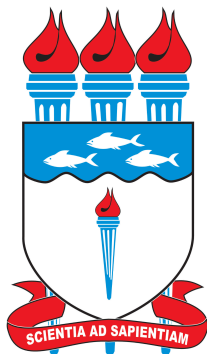


RELATÓRIO DE PROJETO

PROGRAMAÇÃO 2 (POO) 2024.1

MyFood



Pedro Henrique Feliciano da Silva
Bruno Henrique Silva Rocha

MACEIÓ - ALAGOAS

17/10/2024

Classes

- Empresa (superclasse)
 - criação atributos
 - uso do objeto usuario para representar dono
 - inicialização de uma nova instância por meio do uso de construtor
 - getters and setters dos atributos
 - adicionarProduto - adicionar o produto a uma lista de produtos da empresa
 - editarProduto - editar produto usando o id, e lança uma exceção se o id não for encontrado
 - findProdutoById - encontra o produto usando o id
 - listarProdutos - retorna uma lista com nomes dos produtos
 - adicionarEntregador - adiciona um entregador a uma lista de entregadores associados a empresa
- Entrega (classe)
 - criação dos atributos
 - inicialização de uma nova instância por meio do uso de construtor
 - caso destino seja nulo, é atribuído o endereço do cliente
 - getters e setters dos atributos
- Farmacia (subclasse de empresa)
 - inicialização dos atributos por meio de herança
 - criação dos atributos aberto24Horas e numeroFuncionarios
 - inicialização de uma nova instância por meio do uso de construtor
 - getters dos atributos
- Mercado (subclasse de empresa)
 - inicialização dos atributos por meio de herança
 - criação dos atributos abre, fecha e tipoMercado
 - inicialização de uma nova instância por meio do uso de construtor
 - getters e setters dos atributos
- Pedido
 - criação dos atributos
 - inicialização de uma nova instância por meio do uso de construtor
 - getters e setters dos atributos
 - adicionarProduto - adiciona um produto a uma lista de produtos, se o pedido não estiver “fechado”
- Produto
 - criação dos atributos
 - inicialização de uma nova instância por meio do uso de construtor
 - getters e setters dos atributos

- Restaurante (subclasse de empresa)
 - inicialização dos atributos por meio de herança
 - criação do atributo tipoCozinha
 - inicialização de uma nova instância por meio do uso de construtor
 - getters e setters dos atributos
- Usuario (superclasse)
 - criação dos atributos
 - inicialização de uma nova instância por meio do uso de construtor
 - getters e setters dos atributos
- UsuarioCliente (subclasse de usuario)
 - inicialização dos atributos por meio de herança
 - criação dos atributos
 - inicialização de uma nova instância por meio do uso de construtor
- UsuarioDono (subclasse de usuario)
 - inicialização dos atributos por meio de herança
 - criação do atributo cpf
 - inicialização de uma nova instância por meio do uso de construtor
 - getters e setters dos atributos
- UsuarioEntregador (subclasse de usuario)
 - inicialização dos atributos por meio de herança
 - criação dos atributos veiculo e placa
 - inicialização de uma nova instância por meio do uso de construtor
 - getters dos atributos
 - comparação dos ids de dois UsuarioEntregador por meio do equals(Object o)
 - hashCode() - retornar um valor hash para o objeto UsuarioEntregador baseado no seu id

Facade

- construtor (Facade) - inicializa a fachada do sistema
- zerarSistema - reseta o sistema.
- criarUsuario - cria um usuário cliente
- criarUsuario - cria um usuário dono
- criarUsuario - cria um usuário entregador
- criarEmpresa - cria um restaurante.
- criarEmpresa - cria um mercado.
- criarEmpresa - cria uma farmácia
- criarProduto - cria um produto
- adicionarProduto - adiciona um produto a um pedido
- criarPedido - cria um pedido
- criarEntrega - cria uma entrega
- login - faz login de um usuário
- editarProduto - edita um produto
- listarProdutos - lista produtos de uma empresa
- removerProduto - remove um produto de um pedido
- alterarFuncionamento - altera o horário de funcionamento de um mercado
- cadastrarEntregador - cadastra um entregador em uma empresa
- obterPedido - obtém um pedido para um entregador
- liberarPedido - libera um pedido
- fecharPedido - fecha um pedido
- entregar - marca uma entrega como concluída
- getEmpresas - retorna as empresas de um entregador
- getEmpresasDoUsuario - retorna as empresas de um dono
- getProduto - retorna um atributo de um produto
- getEntregadores - retorna os entregadores de uma empresa
- getPedidos - retorna um atributo de um pedido
- getNumeroPedido - retorna o número de um pedido
- getEntrega - retorna um atributo de uma entrega
- getAtributoUsuario - retorna um atributo de um usuário
- getAtributoEmpresa - retorna um atributo de uma empresa
- getIdEmpresa - retorna o ID de uma empresa
- getIdEntrega - retorna o ID de uma entrega
- encerrarSistema - salva os dados de usuários e encerra o sistema
- getSistema - retorna a instância do sistema

Sistema

- *import br.ufal.ic.p2.myfood.Exceptions:*
 - importa todas as exceções personalizadas definidas no pacote Exceptions, todas as exceções do projeto foram cadastradas no pacote
- *import com.google.gson.*:*
 - importa as classes da biblioteca gson, usada para manipular arquivos json
- *import java.io.*:*
 - importa classes para manipular entrada e saída de arquivos
- *import java.lang.reflect.Type:*
 - importa a classe type para usar em operações de reflexão
- *import java.text.DecimalFormat:*
 - importa classe para formatação de numeros decimais
- *import java.text.DecimalFormatSymbols:*
 - importa simbolos especificos para formatação de numeros
- *import java.time.LocalDateTime:*
 - importa a classe LocalDateTime para manipulação de tempos locais
- *import java.util.*:*
 - importa as classes utilitarias, como mapas, listas e conjuntos
- *import java.util.stream.Collectors:*
 - importa a classe Collectors usada para coletar elementos de coleções como maps e listas
- *construtor sistema:*
 - inicializa uma nova instancia do sistema, criando novas hashmaps para usuarios, empresas, pedidos e entregas
- *zerarSistema:*
 - reseta o sistema, redefine os ids, limpa as hashmaps e chama salvarUsuarios() garantindo o reset.
- *configurarGson:*
 - configura o gson, registrando o UsuarioTypeAdapter e o EmpresaTypeAdapter

- criarUsuario:
 - verifica se os parametros são validos
 - verifica se o email existe
 - cria um UsuarioCliente
 - salva o novo usuário

- criarUsuario (caso tenha cpf):
 - verifica se os parametros são validos
 - verifica se o email existe
 - cria um UsuarioDliente
 - salva o novo usuário

- criarUsuario(caso tenha placa):
 - verifica se os parametros são valido
 - verifica se o email e placa existem
 - cria um UsuarioEntregador
 - salva o novo usuário

- criarEmpresa (verifica se é tipo restaurante):
 - verifica se o usuário é um dono
 - verifica se os parametros são validos
 - verificação de nome e endereço, garante que não exista outra empresa com o mesmo nome e endereço
 - cria um restaurante
 - salva a nova empresa

- criarEmpresa (verifica se é tipo mercado):
 - verifica se o usuário é um dono
 - verifica se os parametros são validos
 - verificação de nome e endereço, garante que não exista outra empresa com o mesmo nome e endereço
 - cria um mercado
 - salva a nova empresa

- criarEmpresa(verifica se é tipo farmacia):
 - verifica se o usuário é um dono
 - verifica se os parametros são validos
 - verificação de nome e endereço, garante que não exista outra empresa com o mesmo nome e endereço
 - cria uma farmaciasalva a nova empresa

- carregarDados:
 - cria uma instancia de sistema e configura o Gson
 - carregando usuarios:
 - carrega os usuarios, lendo o users.json
 - atualiza o mapa de usuarios do sistema e atualiza o id do proximo usuario

- carregando empresas:
 - carrega as empresas, lendo os dados de empresas.json
 - atualiza o mapa de empresas do sistema e atualiza o id da proxima empresa
- carregando pedidos:
 - carrega os pedidos, lendo os dados de pedidos.json
 - atualiza o mapa de pedidos do sistema e atualiza o id do proximo pedido
- carregando entregas:
 - carrega as entregas, lendo entregas.json
 - atualiza o mapa de entregas do sistema e atualiza o id da proxima entrega
- adicionarUsuario:
 - adiciona um usuário e salva os dados
- criarProduto:
 - cria um novo produto para uma empresa e salva os dados
- adicionarProduto:
 - adiciona um produto a um pedido existente e salva os dados
- criarPedido:
 - cria um novo pedido para um cliente e uma empresa e salva os dados
- criarEntrega:
 - cria uma entrega e atribui a um pedido e um entregador e salva os dados
- login:
 - verifica o email e a senha do usuário
- editarProduto:
 - edita os detalhes do produto
- listarProdutos:
 - lista os produtos de uma determinada empresa, retorna em formato JSON
- removerProduto:
 - garante que todos os detalhes sejam validos e remove um produto de um pedido específico
- alterarFuncionamento:
 - garante que todos os detalhes sejam validos e altera o horário de funcionamento de um mercado
- cadastrarEntregador:
 - garante que todos os detalhes sejam validos e registra um entregador para uma empresa

- obterPedido:
 - recupera um pedido pronto para um entregador, dando prioridade para farmácias
- liberarPedido:
 - marca um pedido como pronto
- fecharPedido:
 - altera o estado de um pedido para "preparando"
- entregar:
 - garante que todos os detalhes sejam validos e marca uma entrega como concluída
- getEmpresas:
 - retorna uma lista de empresas associadas a um entregador específico
- getEmpresasDoUsuario:
 - retorna uma lista de empresas de um usuariodono em formato JSON
- getProduto:
 - retorna um atributo especifico de um produto
- getEntregadores:
 - retorna uma lista de emails dos entregadores de uma empresa especifica
- getPedidos:
 - retorna um atributo especifico de um pedido
- getNumeroPedido:
 - retorna o número de um pedido especifico de um cliente em uma empresa
- getEntrega:
 - retorna um atributo especifico de uma entrega
- getAtributoUsuario:
 - retorna um atributo especifico de um usuário
- getAtributoEmpresa:
 - retorna um atributo especifico de uma empresa
- getIdEmpresa:
 - retorna o id de uma empresa de acordo com o nome e o dono
- getIdEntrega:
 - retorna o id de uma entrega associada a um pedido especifico

- salvarUsuarios:
 - salva a lista de usuários em um arquivo
- salvarEmpresas:
 - salva a lista de empresas em um arquivo
- salvarPedidos:
 - salva a lista de pedidos em um arquivo
- salvarEntregas:
 - salva a lista de entregas em um arquivo
- findUsuarioById:
 - procura um usuário pelo ID
- findEmpresaById:
 - procura uma empresa pelo ID
- findProdutoById:
 - procura um produto pelo ID em todas as empresas
- findPedidoById:
 - procura um pedido pelo ID
- UsuarioTypeAdapter:
 - desserializa um arquivo json, convertendo de volta a objeto em uma instancia da classe Usuario correta, baseado em seu "tipo". tambem o serializa, convertendo uma instancia de usuario em um arquivo JSON, adicionando o "tipo"
- EmpresaTypeAdapter:
 - desserializa um arquivo json, convertendo de volta a objeto em uma instancia da classe Empresa apropriada, baseado em seu "tipo"