

Seja o sistema de vendas RangoNaWeb, que controla a venda de quentinhas e bebidas via web. O cliente cadastrado liga, faz o pedido da comida baseado num menu disponível, seleciona a bebida e agenda a hora de entrega.

Teste com Junit

Utilizando o Junit, faça o teste unitário da classe Pedido que registra dia e hora do pedido, dia e hora da entrega, dados do cliente códigos dos produtos, preços, quantidade de cada um, etc vendidos.

Roteiro:

- Defina os requisitos que estão sendo implementados no caso de uso Fazer Pedido
- Implemente a classe Pedido em Java. **De forma proposital**, gere defeitos (bugs) no código.
- Construa os stubs e drivers que forem necessários, se for o caso.
- Faça um projeto do que você vai testar (tabela: dados X RE X RO). Utilize as técnicas de partições e limites para definir o que testar.
- Implemente os testes utilizando JUnit.
- Execute o teste, preencha o campo RO e avalie o resultado.

Observações

- Grupo de até 4 pessoas
- O arquivo deve ser em TXT ou PDF (não zipar)
- Data da entrega: 16/ 04/ 2019

-----//-----

Nomes: Fábio Arnon Campidelli de Pádua, Igor Sabarense Coelho Silva e Rithie Natan Carvalhaes Prado

Requisitos

Requisitos Funcionais	
ID	Descrição do Requisito
#01	<CRUD> Cardápio(Bebidas Quentinha)
#02	<CRUD> Cliente
#03	<CRUD> Endereço
#04	<CRUD> Pedido
#05	O sistema deve registrar a data e o horário do pedido feito.
#06	O cliente deve ter acesso ao cardápio do sistema de vendas.

#07	O cliente deve ter acesso ao comprovante do pedido ao final da execução do pedido.
Requisitos Não Funcionais	
#01	Desenvolvimento do projeto em Java.
#02	Utilização do JUnit para a execução do teste unitário.

Código

Classe Cliente

```
package rangonaweb.entity;

public class Cliente {
    private String nome, cpf;
    private Endereco endereco;

    public Cliente() {
    }
    public Cliente(String cpf, String nome, Endereco endereco) {
        setNome(nome);
        setCpf(cpf);
        setEndereco(endereco);
    }

    String getNome() {
        return this.nome;
    }
    String getCpf() {
        return this.cpf;
    }
    Endereco getEndereco() {
        return this.endereco;
    }
    void setNome(String nome) {
        this.nome = nome;
    }
    void setCpf(String cpf) {
        this.cpf = cpf;
    }
    void setEndereco(Endereco endereco) {
        this.endereco = endereco;
    }
}
```

Classe Endereço

```

package rangonaweb.entity;

public class Endereco {
    private String rua, cep, cidade, bairro, estado, numero;

    public String getRua() {
        return rua;
    }
    public void setRua(String rua) {
        this.rua = rua;
    }
    public String getCep() {
        return cep;
    }
    public void setCep(String cep) {
        this.cep = cep;
    }
    public String getCidade() {
        return cidade;
    }
    public void setCidade(String cidade) {
        this.cidade = cidade;
    }
    public String getBairro() {
        return bairro;
    }
    public void setBairro(String bairro) {
        this.bairro = bairro;
    }
    public String getEstado() {
        return estado;
    }
    public void setEstado(String estado) {
        this.estado = estado;
    }
}

```

```

    public String getNumero() {
        return numero;
    }
    public void setNumero(String numero) {
        this.numero = numero;
    }

    public Endereco() {
    }
    public Endereco(String rua, String cep, String cidade, String bairro, String estado, String numero) {
        setBairro(bairro);
        setCep(cep);
        setCidade(cidade);
        setEstado(estado);
        setNumero(numero);
        setRua(rua);
    }
}

```

Classe Pedido

```

package rangonaweb.entity;

import java.time.LocalDateTime;
import rangonaweb.enumerations.MenuBebidas;
import rangonaweb.enumerations.MenuQuentinha;

public class Pedido {
    private Cliente cliente;
    private MenuQuentinha quentinha;
    private MenuBebidas bebida;
    private LocalDateTime horaEntrega;
    private Integer quantidade;

    public Cliente getCliente() {
        return cliente;
    }
    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }
    public MenuQuentinha getQuentinha() {
        return quentinha;
    }
    public void setQuentinha(MenuQuentinha quentinha) {
        this.quentinha = quentinha;
    }
    public MenuBebidas getBebida() {
        return bebida;
    }
    public void setBebida(MenuBebidas bebida) {
        this.bebida = bebida;
    }
    public LocalDateTime getHoraEntrega() {
        return horaEntrega;
    }
    public void setHoraEntrega(LocalDateTime horaEntrega) {
        this.horaEntrega = horaEntrega;
    }
}

```

```

public Pedido() {
}
public Pedido(Cliente cliente, MenuQuentinha quentinha, MenuBebidas bebida, LocalDateTime horaEntrega, Integer quantidade) {
    setBebida(bebida);
    setCliente(cliente);
    setHoraEntrega(horaEntrega);
    setQuentinha(quentinha);
    setQuantidade(null); //erro proposital
}

public Integer getQuantidade() {
    return quantidade;
}
public void setQuantidade(Integer quantidade) {
    this.quantidade = quantidade;
}

@Override
public String toString() {
    return "
        getCliente().getNome() + "\n" +
        getCliente().getCpf() + "\n" +
        getCliente().getEndereco().getRua() + ", " + getCliente().getEndereco().getBairro() + ", " + getCliente().getEndereco().getCidade() +
        ", " + getCliente().getEndereco().getNumero() + "\n" +
        getQuentinha() + "\n" +
        getBebida() + "\n" +
        getHoraEntrega().format(DateTimeFormatter.ISO_LOCAL_DATE) + "\n" +
        getQuantidade();
}
}

```

Classe Cardápio(Bebidas|Quentinha)

```

package rangonaweb.enumerations;

public enum MenuBebidas {
    REFRIGERANTE,
    CERVEJA
}

```

```
package rangonaweb.enumerations;

public enum MenuQuentinha {

    PRATO_EXECUTIVO,
    PRATO_LEGISLATIVO,
    PRATO_ADMINISTRATIVO

}
```

Classe de Testes

```
package rangonaweb.junit;

import static org.junit.Assert.assertNotNull;
import static org.junit.Assert.assertEquals;
import java.time.LocalDateTime;
import org.junit.Test;
import rangonaweb.entity.Cliente;
import rangonaweb.entity.Endereco;
import rangonaweb.entity.Pedido;
import rangonaweb.enumerations.MenuBebidas;
import rangonaweb.enumerations.MenuQuentinha;

public class ProdutoTestJUnit {

    @Test
    public void testeRecibo(){

        Endereco endereco = new Endereco("Dom Jos Gaspar","34004-600", "Belo Horizonte", "Cora Eucarstico", "Minas Gerais", "115");
        Cliente cliente = new Cliente("441.397.580-40","Pietro",endereco);
        Pedido pedidoAtual = new Pedido(cliente , MenuQuentinha.PRATO_EXECUTIVO, MenuBebidas.CERVEJA, LocalDateTime.now(), 2);

        Pedido pedidoEsperado = new Pedido(cliente , MenuQuentinha.PRATO_EXECUTIVO, MenuBebidas.CERVEJA, LocalDateTime.now(), 2);
        pedidoEsperado.setQuantidade(2);

        assertEquals(pedidoEsperado.toString(), pedidoAtual.toString()); // testa o recibo dos pedidos e verifica se est correto

    }

    @Test
    public void testeClienteNaoNulo(){

        Endereco endereco = new Endereco("Dom Jos Gaspar","34004-600", "Belo Horizonte", "Cora Eucarstico", "Minas Gerais", "115");
        Cliente cliente = new Cliente("441.397.580-40","Pietro",endereco);
        Pedido pedidoAtual = new Pedido(cliente , MenuQuentinha.PRATO_EXECUTIVO, MenuBebidas.CERVEJA, LocalDateTime.now(), 2);

        assertNotNull(pedidoAtual.getCliente());

    }

}
```

```
@Test
public void testeClienteNaoNulo(){

    Endereco endereco = new Endereco("Dom Jos Gaspar","34004-600", "Belo Horizonte", "Cora Eucarstico", "Minas Gerais", "115");
    Cliente cliente = new Cliente("441.397.580-40","Pietro",endereco);
    Pedido pedidoAtual = new Pedido(cliente , MenuQuentinha.PRATO_EXECUTIVO, MenuBebidas.CERVEJA, LocalDateTime.now(), 2);

    assertNotNull(pedidoAtual.getCliente());

}

@Test
public void testQuantidade(){

    Endereco endereco = new Endereco("Dom Jos Gaspar","34004-600", "Belo Horizonte", "Cora Eucarstico", "Minas Gerais", "115");
    Cliente cliente = new Cliente("441.397.580-40","Pietro",endereco);
    Pedido pedidoAtual = new Pedido(cliente , MenuQuentinha.PRATO_EXECUTIVO, MenuBebidas.CERVEJA, LocalDateTime.now(), 2);

    assertEquals(2,pedidoAtual.getQuantidade());

}

@Test
public void testPedidoNaoNulo(){

    Endereco endereco = new Endereco("Dom Jos Gaspar","34004-600", "Belo Horizonte", "Cora Eucarstico", "Minas Gerais", "115");
    Cliente cliente = new Cliente("441.397.580-40","Pietro",endereco);
    Pedido pedidoAtual = new Pedido(cliente , MenuQuentinha.PRATO_EXECUTIVO, MenuBebidas.CERVEJA, LocalDateTime.now(), 2);

    assertNotNull(pedidoAtual);

}

}
```

Tabela de Testes

Dados	Resposta Esperada (RE)	Resposta Obtida (RO)
Teste de dados do Recibos	2	Erro: Nulo

Teste de cliente não nulo	Aceito	Aceito
Teste do valor de quantidade	2	Erro: Nulo
Teste de Pedido não Nulo	Pedido Aceito	Aceito