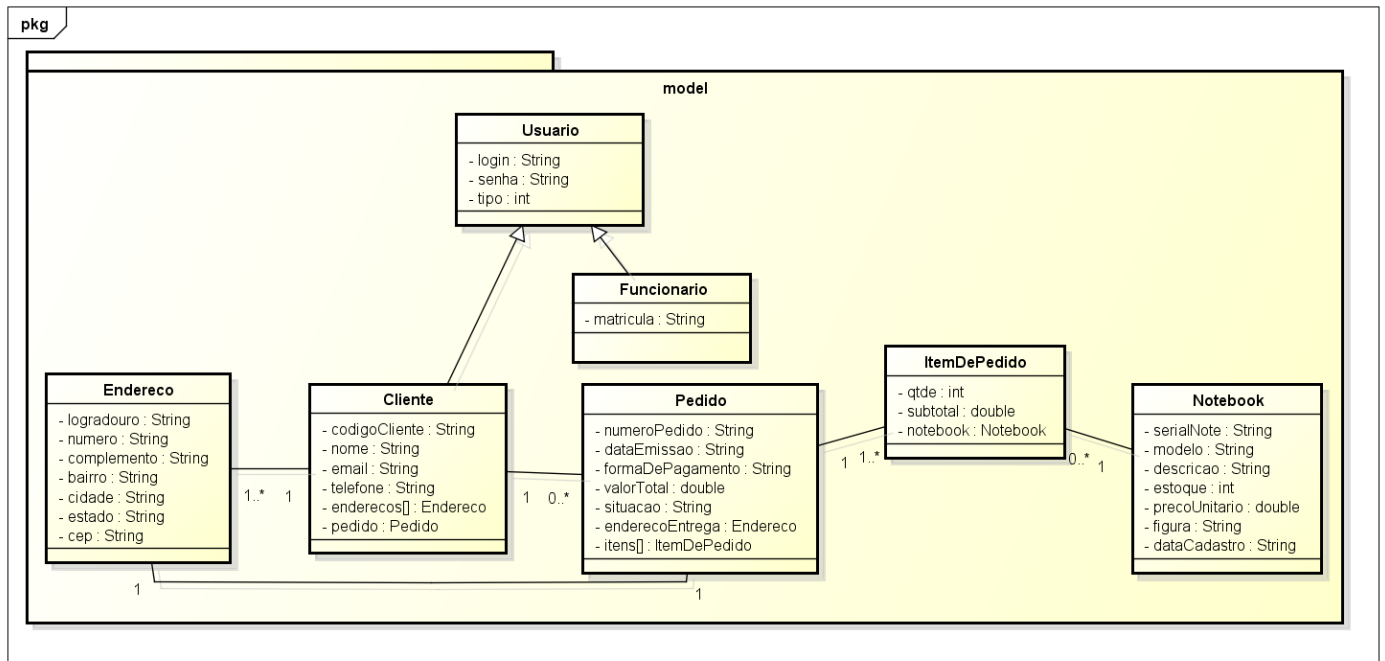


Projeto *InfoNote_06*

Execute este passo a passo:



powered by astah

- 1 – Copiar e colar o projeto *InfoNote_05* e renomear para *InfoNote_06*.
- 2 – Renomear a Classe Usuario, para UsuarioOld .
- 3 – Adaptar as Classes Cliente, Pedido e Notebook, pois houveram mudanças no diagrama acima, com relação aos anteriores.
- 4 – Criar SuperClasse Usuario com os atributos do diagrama acima.
- 5 – Criar a SubClasse Funcionario conforme diagrama acima:

```
package model;
public class Funcionario extends Usuario {

    private String matricula;

    public String getMatricula() {
        return matricula;
    }

    public void setMatricula(String matricula) {
        this.matricula = matricula;
    }

    public Funcionario() {
        super();
    }

    public Funcionario(String login, String senha, int tipo, String matricula) {
        super(login, senha, tipo);
        this.matricula = matricula;
    }
}
```

6 – Criar a SubClasse Cliente conforme diagrama acima, não esquecendo de suas associações:

```
package model;

public class Cliente extends Usuario{
    private String codigoCliente;
    private String nome;
    private String email;
    private String telefone;
    private Endereco enderecos[] = new Endereco[10];
    private Pedido pedidos[] = new Pedido[10];

    public String getCodigoCliente() {
        return codigoCliente;
    }
    public void setCodigoCliente(String codigoCliente) {
        this.codigoCliente = codigoCliente;
    }
    public String getNome() {
        return nome;
    }
    public void setNome(String nome) {
        this.nome = nome;
    }
    public String getEmail() {
        return email;
    }
    public void setEmail(String email) {
        this.email = email;
    }
    public String getTelefone() {
        return telefone;
    }
    public void setTelefone(String telefone) {
        this.telefone = telefone;
    }
    public Endereco[] getEnderecos() {
        return enderecos;
    }
    public void setEnderecos(Endereco[] enderecos) {
        this.enderecos = enderecos;
    }
    public Pedido[] getPedidos() {
        return pedidos;
    }
    public void setPedidos(Pedido[] pedidos) {
        this.pedidos = pedidos;
    }
    public Cliente() {
        super();
    }
    public Cliente(String login, String senha, int tipo, String codigoCliente,
        String nome, String email, String telefone, Endereco endereco) {
        super(login, senha, tipo);
        this.codigoCliente = codigoCliente;
        this.nome = nome;
        this.email = email;
        this.telefone = telefone;
        //insere na posição zero do vetor o endereço residencial
        inserirEndereco(endereco);
    }
}
```

```

//inserir o pedido no vetor
public boolean inserirPedido(Pedido pedido){
    for (int i = 0; i < pedidos.length; i++){
        if (pedidos[i] == null) {
            pedidos[i] = pedido;
            return true;
        }
    }
    return false;
}

//inserir o endereço no vetor
public boolean inserirEndereco (Endereco end){
    for (int i = 0; i < enderecos.length; i++){
        if (enderecos[i] == null){
            enderecos[i] = end;
            return true;
        }
    }
    return false;
}

```

7 – Criar o método toString nas Classes Cliente e Usuario (Este método é utilizado para demonstrar o objeto em formato texto quando é solicitada a impressão do objeto. Ex: `System.out.println(cliente);`)

Exemplo Classe Cliente.

```

@Override
public String toString(){
    final String ENTER = "\n";
    String retValue = "";

    retValue = "Informações sobre o cliente:" + ENTER +
    super.toString()+
    "Código do Cliente: " + codigoCliente + ENTER +
    "Nome: " + nome + ENTER +
    "E-mail: " + email + ENTER +
    "Telefone: " + telefone + ENTER +
    "Endereco: " + enderecos[0];

    return retValue;
}

```

8 – Alterar o construtor da classe Pedido adicionando o endereço de entrega:

```

public Pedido(String numeroPedido, String dataEmissao, String formaDePagamento,
    double valorTotal, String situacao, Endereco enderecoEntrega){
    super();
    this.numeroPedido = numeroPedido;
    this.dataEmissao = dataEmissao;
    this.formaDePagamento = formaDePagamento;
    this.valorTotal = valorTotal;
    this.situacao = situacao;
    this.enderecoEntrega = enderecoEntrega;
}

```

9 – Substituir na Classe Pedido o método mostrar() pelo método toString():

```
@Override
public String toString(){
    final String ENTER = "\n";
    String retValue = "";

    retValue = "Informações sobre Pedido:" + ENTER +
    "Número do Pedido: " + numeroPedido + ENTER +
    "Data de Emissão: " + dataEmissao + ENTER +
    "Forma de pagamento: " + formaDePagamento + ENTER +
    "Valor Total: " + valorTotal + ENTER +
    "Situação: " + situacao + ENTER +
    "Endereço de Entrega: " + ENTER +
    "Itens: ";
    for (int i = 0; i < itens.length; i++){
        retValue += itens[i] + ENTER;
    }
    return retValue;
}
```

10 – Substituir nas demais Classes o método mostrar() pelo método toString().

11 – Apagar as Classes Teste (no pacote controller) e UsuarioOld (no pacote model).

12 – Desafio: Adaptar a Classe InfoNote (no pacote controller) para que fique funcional (sem erros).