

Sistema de Gestão de Fábrica de Roupas

- Etapa 1 - Modelagem estrutural e dinâmica em alto nível de abstração
 - Diagrama de classes, casos de uso e visão geral de interação
- Etapa 2 - Refinamento estrutural
- Etapa 3 - Refinamento de casos de uso
 - Diagrama de atividades (cenário principal) e de sequência
- Etapa 4 - Modelagem de estados associada à classe
 - Diagrama de máquinas de estado
- Etapa 5 - Introdução de elementos do domínio da solução computacional
- Etapa 6 - Modelagem de algoritmos de métodos
 - Diagrama de atividades ,exemplo específicos
- Etapa 7 - Geração de código e o desenvolvimento iterativo

Qual a função do sistema ?

Sistema composto por 11 classes, dentre essas classes os principais agentes são “Cliente” e “Supervisor” , responsáveis por realizar as principais funções do sistema, sendo elas solicitar uma peça de roupa e gerir o funcionamento da fábrica em geral.



Etapa 1 - Diagrama de Classes

- Elementos do domínio do problema
- Apenas classes e seus relacionamentos
- Ausência atributos e métodos

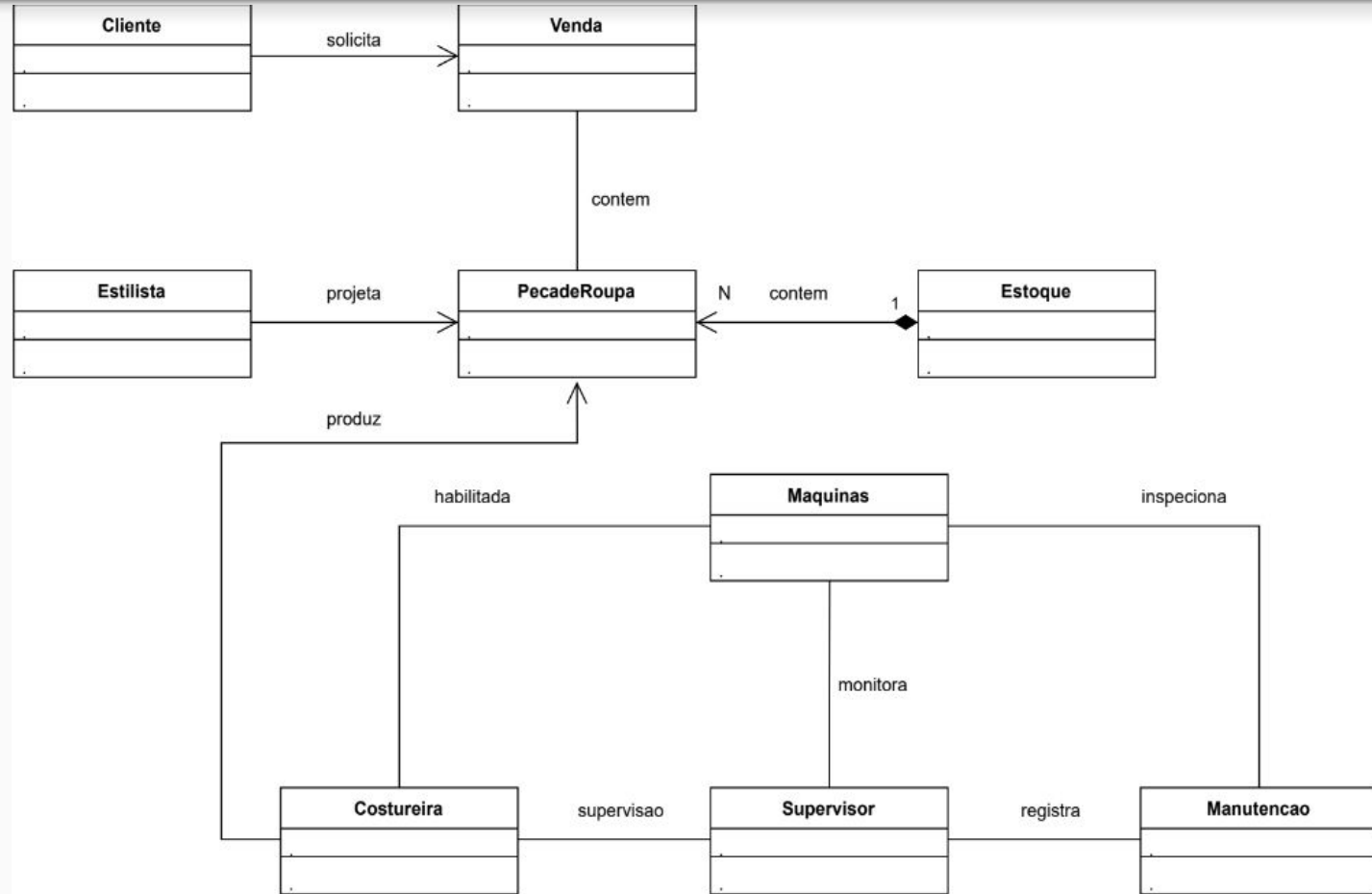


Diagrama de Casos de Uso

- Identificação de todas funcionalidades
- fatores externos que influenciam no sistema
- utilização de inclusão e extensão
 - caso auxiliar pode ocorrer ou deve ocorrer
- Ator Supervisor possui visibilidade sobre as peças de roupa, máquinas e costureiras

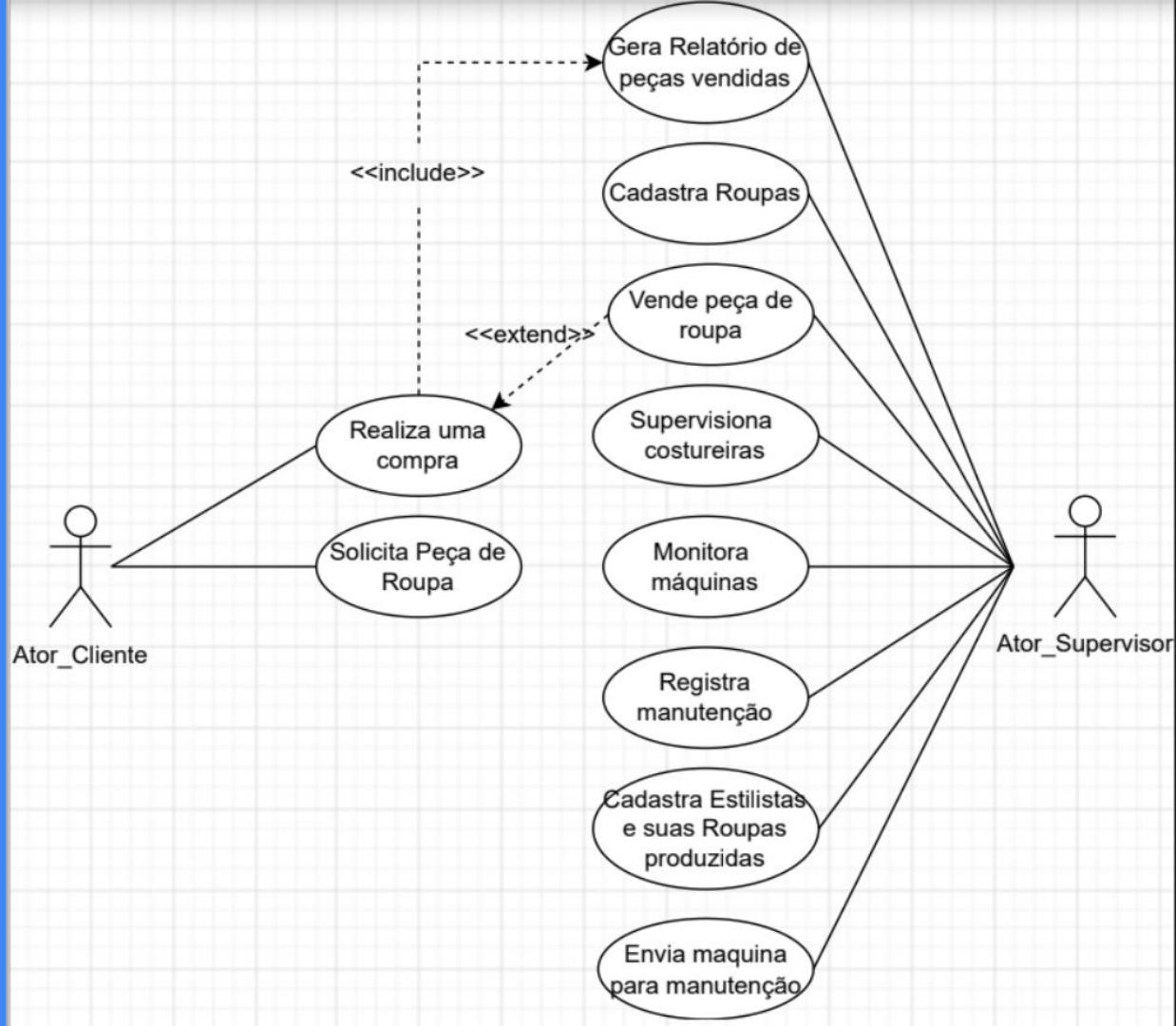
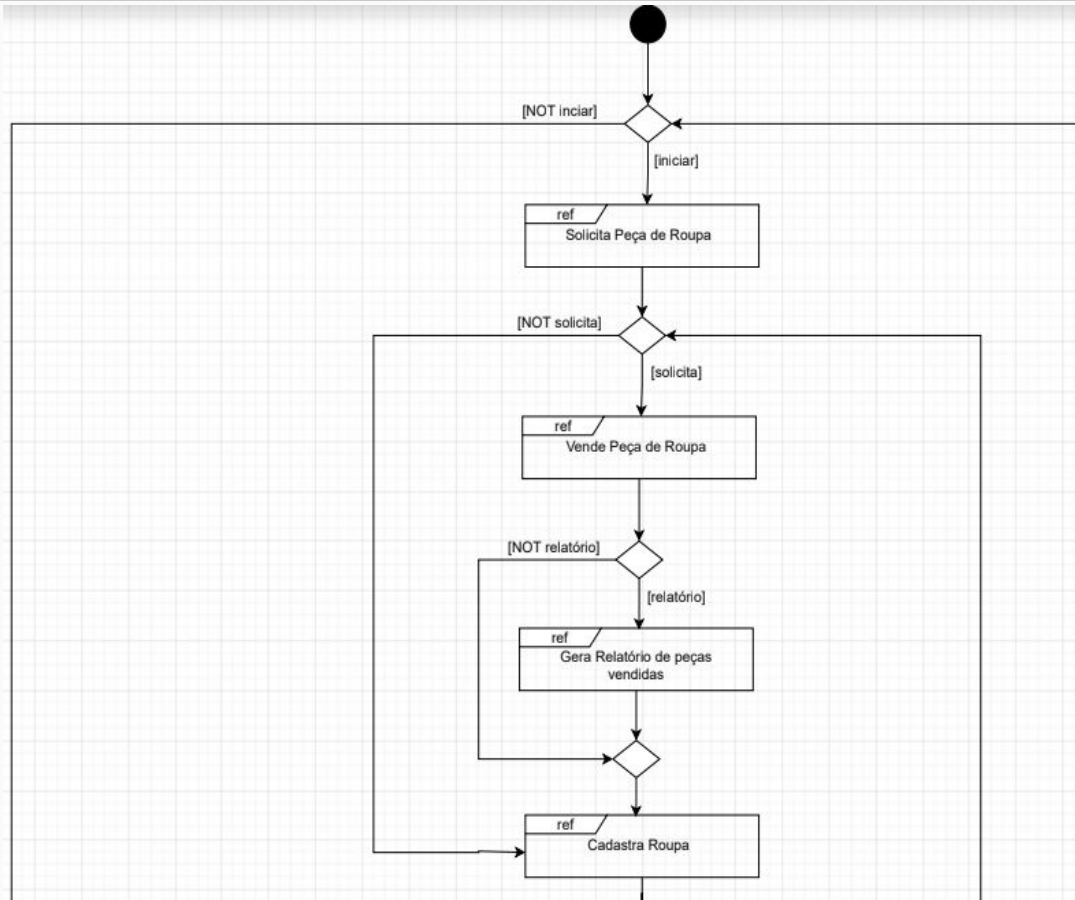
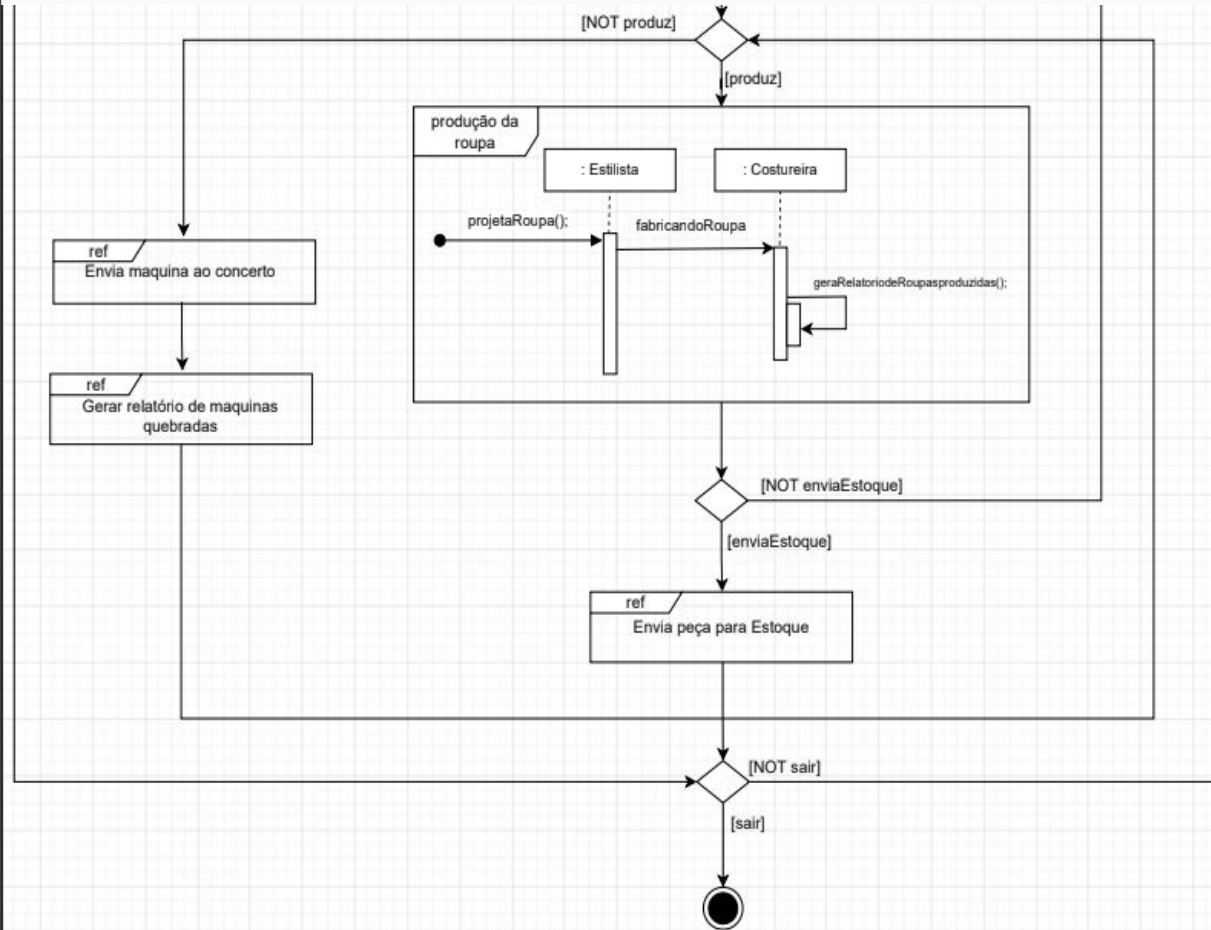
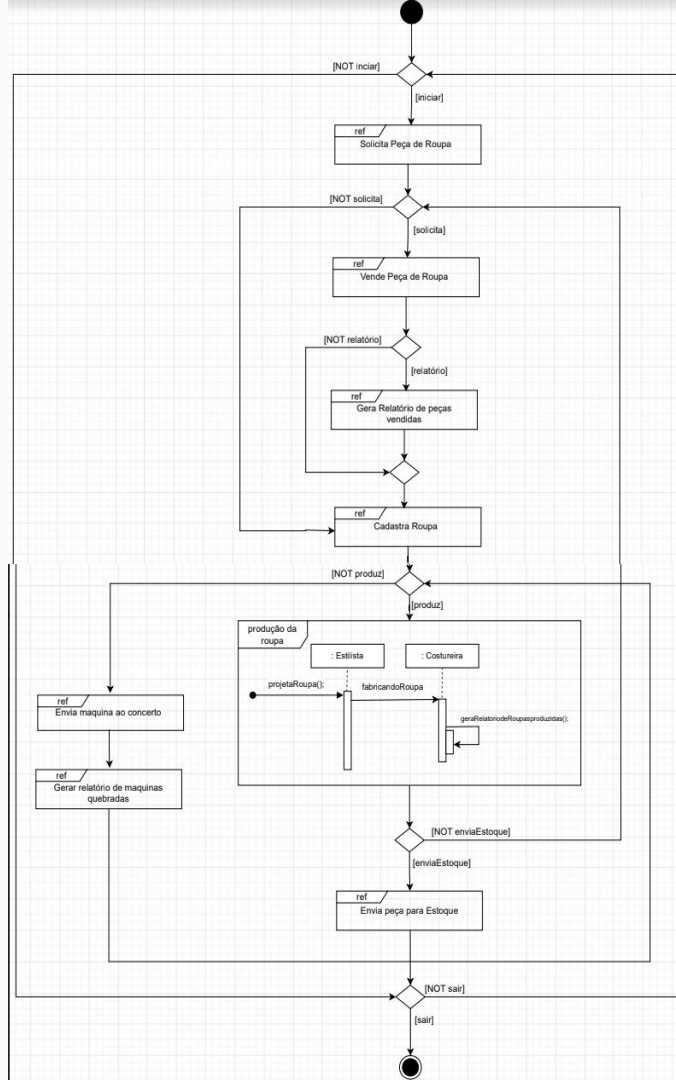


Diagrama de Visão Geral de Interação



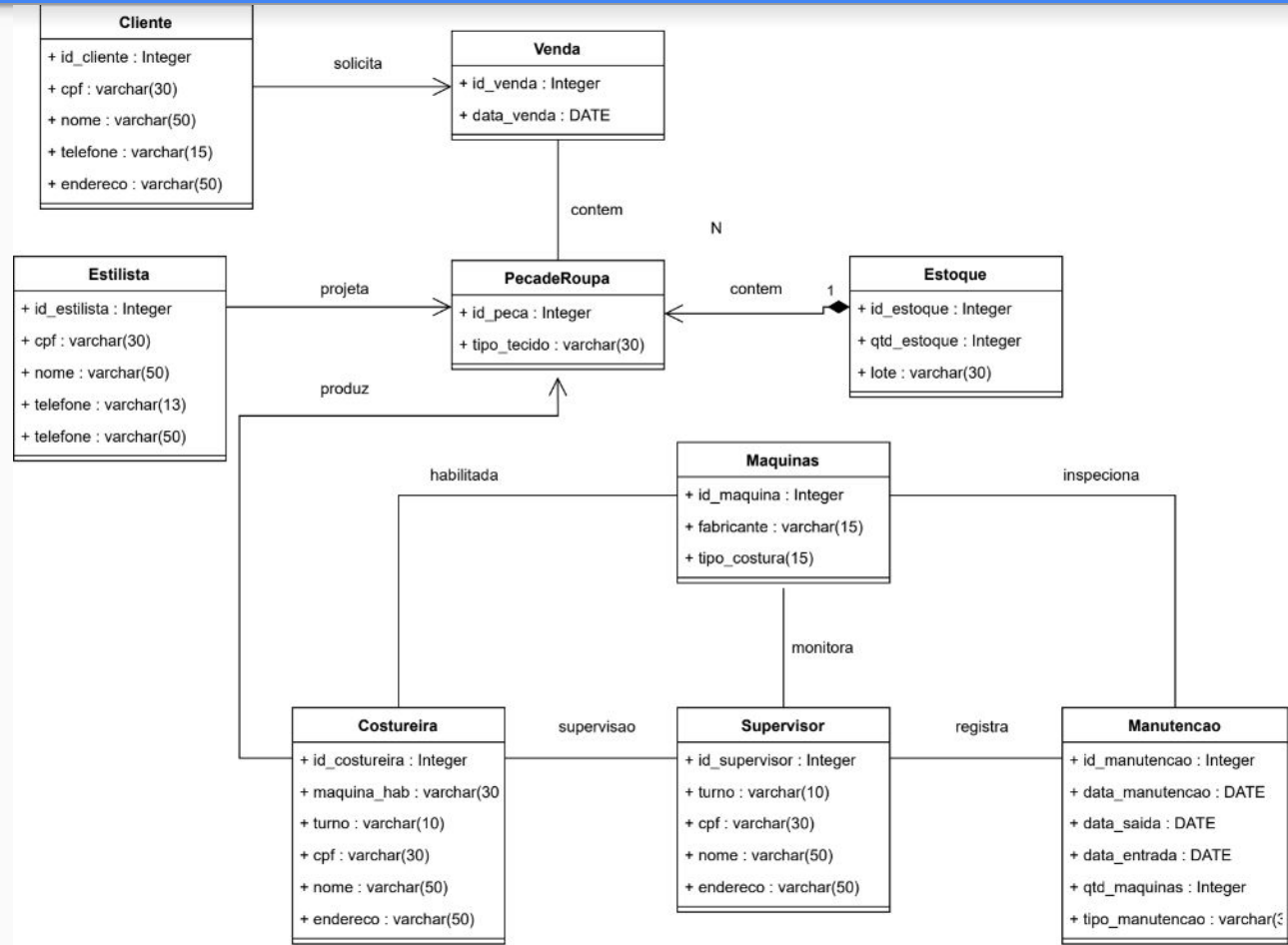
- Representação visual do fluxo geral de interação entre os diferentes elementos do sistema
- Mostra as possíveis tomadas de decisões durante a utilização
- Não apresenta detalhes de implementação, métodos ou atributos
- Se tornam muito complexos quando o projeto possui diversos componentes



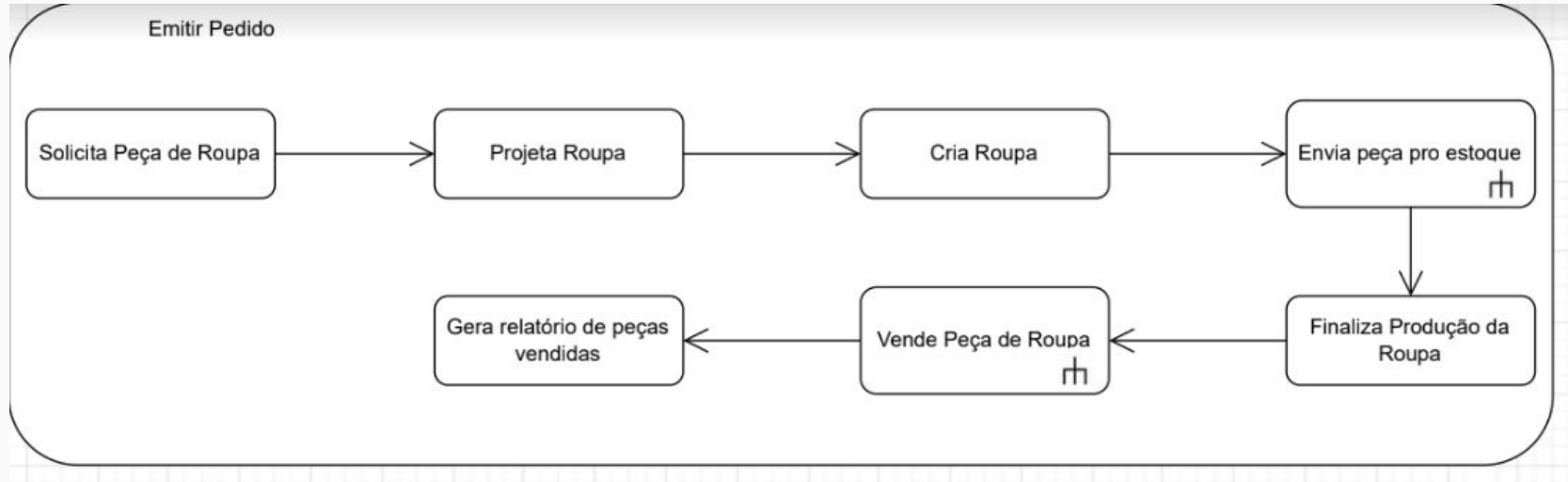


Etapa 2 - Diagrama de Classes (Refinamento Estrutural)

- Identificação de atributos, com a ausência dos métodos
- Tem como base de criação as classes e o diagrama de casos de uso

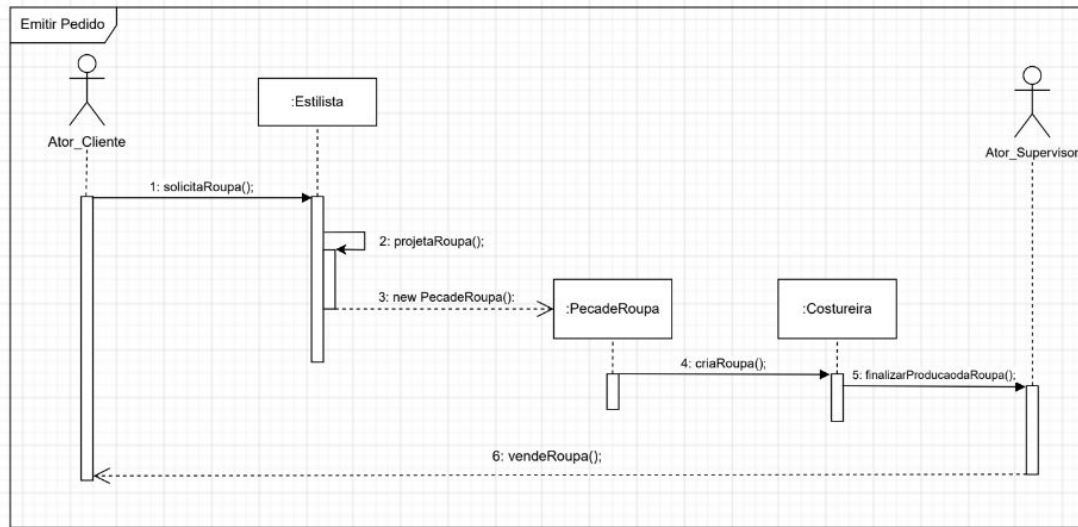
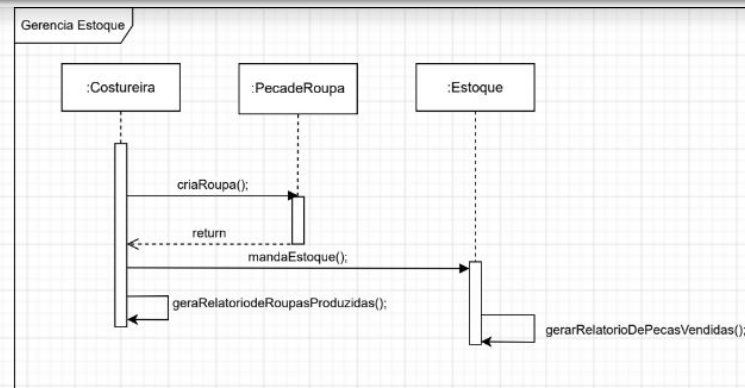
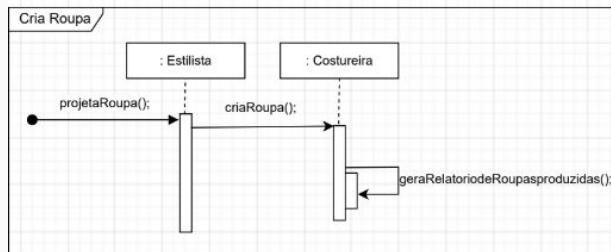
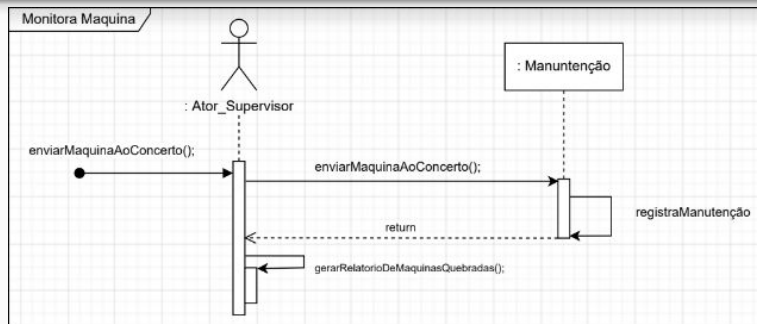


Etapa 3 - Diagrama de Atividades (cenário principal)



- Mostra o fluxo do trabalho no sistema utilizando os nós , como as tarefas são realizadas e como elas estão relacionadas.
 - Facilita visualização , identificação de problemas e comunicação de informações sobre processos para diferentes pessoas.

Diagrama de Sequência

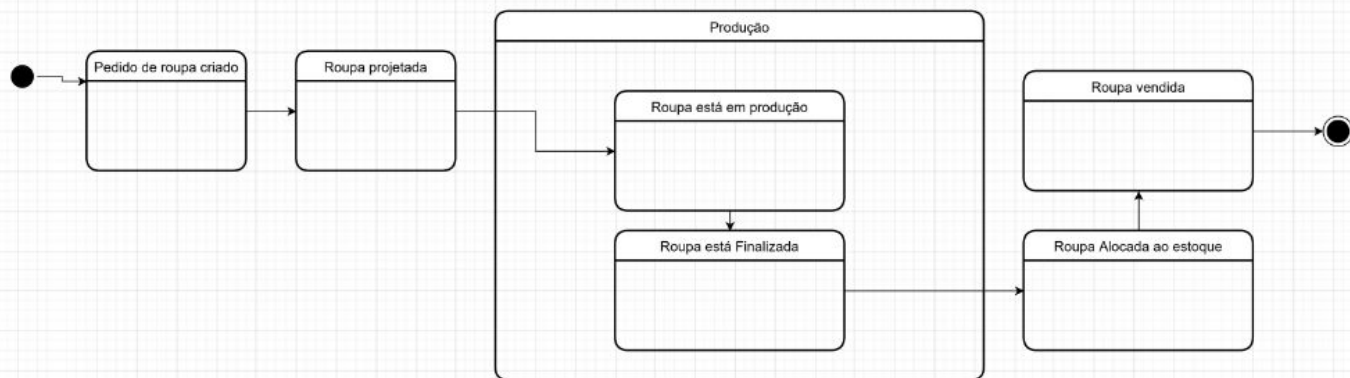
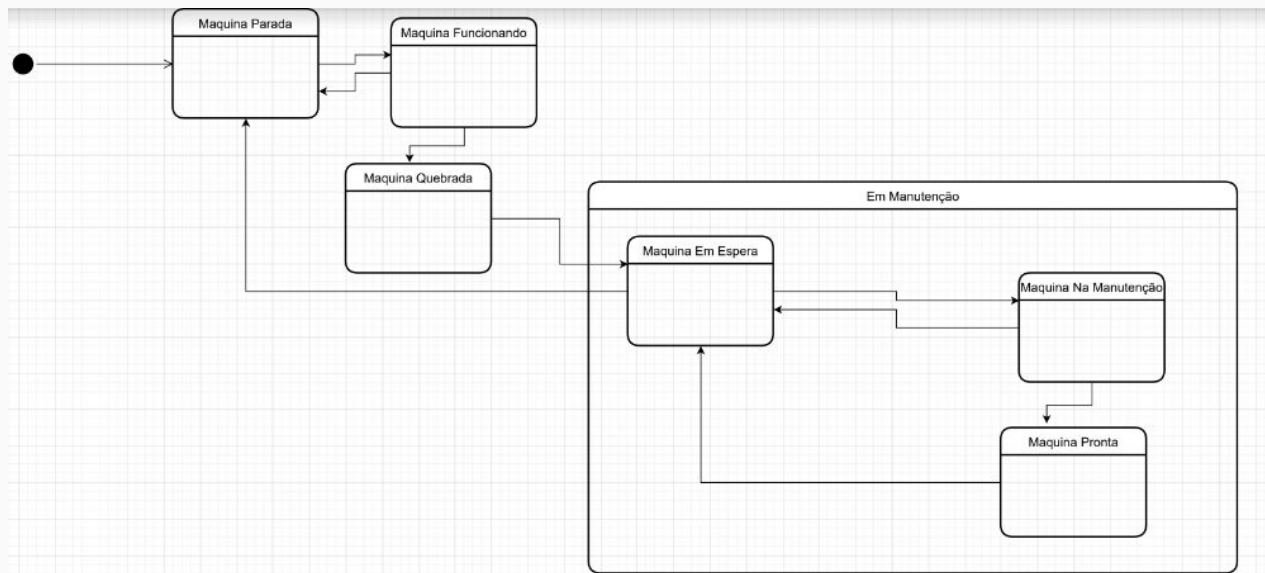


Contribuições de um Diagrama de Sequência

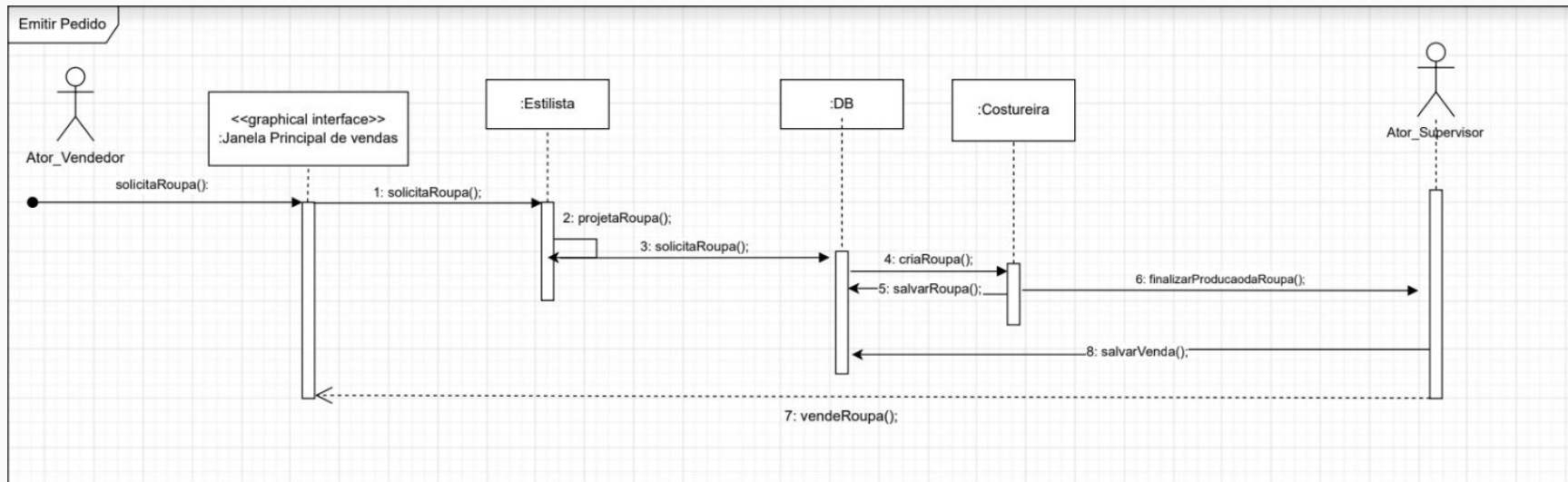
- Modelagem dos comportamentos seguindo o padrão de um programa orientado a objetos
 - interação em tempo de execução, criação de objetos, invocação de métodos
- Não apresenta a estrutura dos objetos apenas as mensagens trocadas entre eles
- Auxilia a identificar problemas de design e implementação como dependências cíclicas ou mensagens que não são tratadas

Etapa 4 - Diagrama de Máquina de Estado

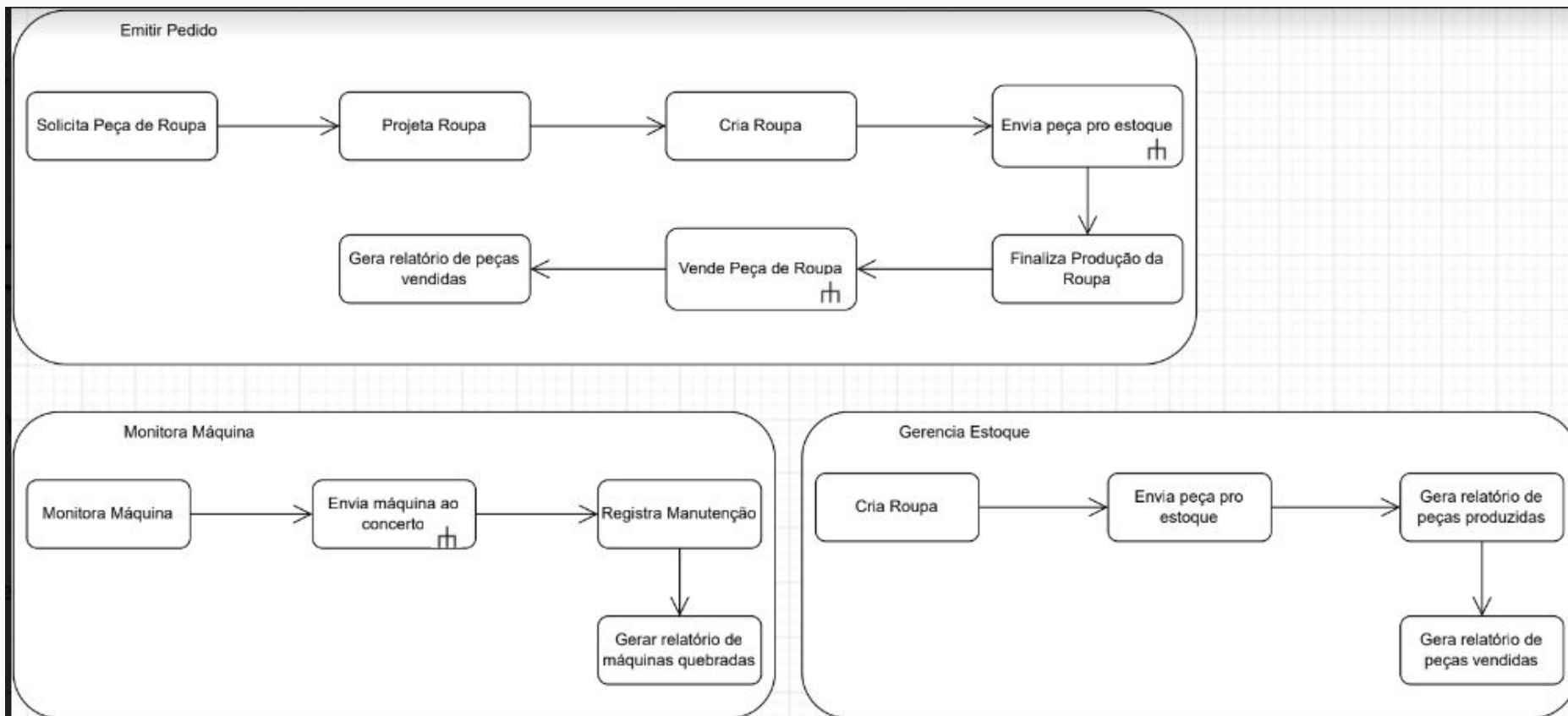
- Mostra interação entre objetos e componentes
- Demonstra o fluxo de mensagem entre objetos
- Mostra de forma bem visual processos de transição entre os estados do sistema



Etapa 5 - Introdução de elementos do domínio.



Etapa 7 - Modelagem de algoritmo de método (Diagrama de Atividades)



Etapa 8 - Geração de código

- Cliente
- Costureira
- Estilista
- Estoque
- Main
- Manutencao
- MaquinaDeCostura
- Roupa
- Supervisor
- TipoDeCostura
- TipoDeTecido
- Venda

```
Guilherme Dias Cardoso *
@Data
@AllArgsConstructor

public class Cliente {
    private int id;
    private String cpf;
    private String nome;
    private String endereco;
    private String telefone;
    private List<Venda> compras = new ArrayList<>();

    Guilherme Dias Cardoso
    public Venda solicitarRoupa(String roupa, Integer quantidade) {
        System.out.println("Solicitando a roupa: " + roupa);
        Venda venda = new Venda( cliente: this, quantidade);
        compras.add(venda);
        return venda;
    }
}
```


Classe Supervisor

- Métodos de manipulação de objetos e geração de relatórios.

```
@Data
@AllArgsConstructor
public class Supervisor {
    private String id;
    private String turno;
    private String cpf;
    private String nome;
    private String edereco;
    private List<Costureira> costureiras;
    private List<MaquinaDeCostura> maquinaDeCosturaHashMap = new ArrayList<>();

    // Guilherme Dias Cardoso
    public void gerarRelatorioDeMaquinasQuebradas() {
        System.out.println("====Gerando relatório de maquinas quebradas");
        for (MaquinaDeCostura maquinaDeCostura : maquinaDeCosturaHashMap) {
            if (maquinaDeCostura.isEstaEmManutencao()) {
                System.out.println("A maquina com o Id: " + maquinaDeCostura.getId() + " esta em manutenção!");
            }
        }
    }

    // Guilherme Dias Cardoso
    public Manutencao enviarMaquinaAoConcerto(MaquinaDeCostura maquina) {
        System.out.println("Enviando a maquina: " + maquina.getId() + " ao concerto!");
        addMaquina(maquina);
        return new Manutencao(maquina);
    }

    // Guilherme Dias Cardoso
    public Venda venderPecaAoCliente(Estoque estoque, Roupa roupa, Venda venda) {
        System.out.println("Realizando a venda ao cliente!");
        estoque.getRoupas().remove(roupa);
        venda.addRoupas(roupa);
        return venda;
    }

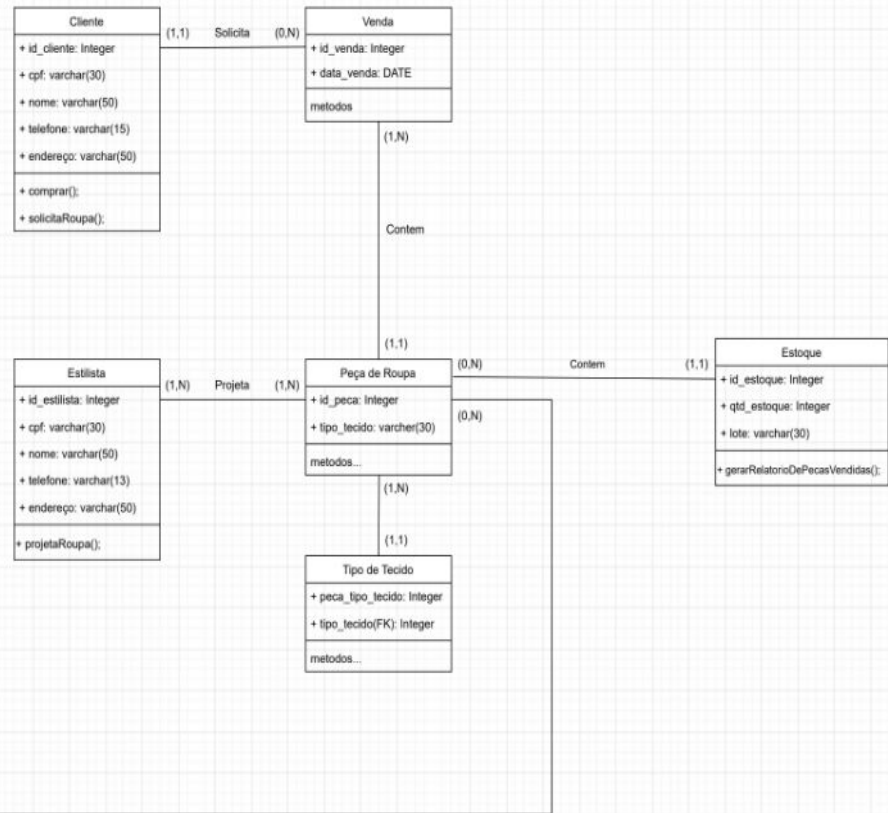
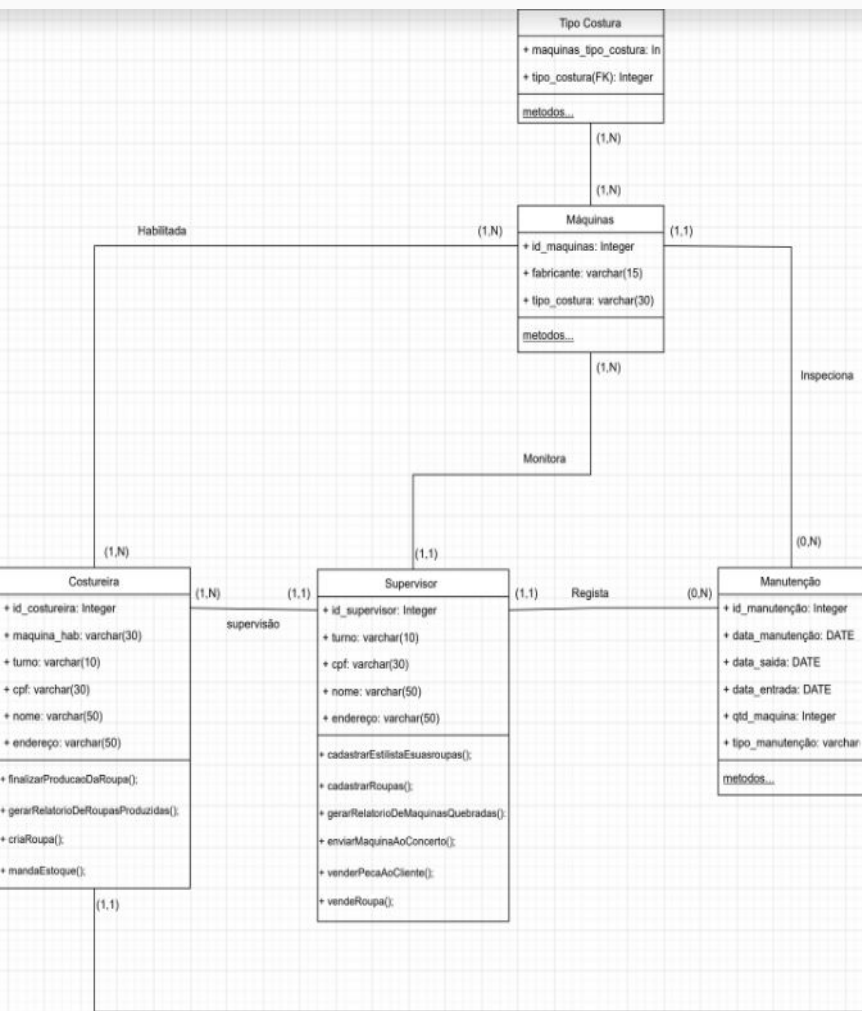
    // Cardoso, 09/01/23 22:56 + .
    // Guilherme Dias Cardoso
    public void removeDaManutencao(Manutencao manutencao, MaquinaDeCostura maquinaDeCostura) {
        System.out.println("Removendo a maquina: " + maquinaDeCostura.getId() + " da manutenção!");
        manutencao.setData_saida(new Date());
        maquinaDeCostura.setEstaEmManutencao(false);
    }
}
```

Processo de emitir o Pedido.

```
public static void emitirPedido() {  
    Cliente cliente = new Cliente( id: 1, cpf: "123456789", nome: "João", endereco: "Rua  
    Estilista estilista = new Estilista( id: 1, cpf: "123456789", telefone: "35984114444",  
    Supervisor supervisor = new Supervisor( id: "1", turno: "Manha", cpf: "123456789", n  
    MaquinaDeCostura maquinaDeCostura = new MaquinaDeCostura(1, List.of(new TipoDeTecid  
    Costureira costureira = new Costureira( id: "1", nome: "Marcia", roupaAtual: null, ma  
    Estoque estoque = new Estoque());  
  
    System.out.println("Processo de emitir Pedido");  
    Venda venda = cliente.solicitarRoupa( roupa: "Vestido Vermelho", quantidade: 2);  
    Roupa roupa = estilista.projetaRoupa( nome: "Vestido Vermelho");  
    Roupa roupaProduzida = costureira.criaRoupa(roupa);  
    costureira.finalizarProducaoDaRoupa(roupaProduzida, estoque);  
    Venda vendaFinal = supervisor.venderPecaAoCliente(estoque, roupaProduzida, venda);  
}
```

Processo de Gerenciar o Estoque

```
public static void gerenciaEstoque() {  
  
    Estilista estilista = new Estilista( id: 1, cpf: "123456789", telefo  
    Supervisor supervisor = new Supervisor( id: "1", turno: "Manha", cp  
    MaquinaDeCostura maquinaDeCostura = new MaquinaDeCostura(1, List.c  
    Costureira costureira = new Costureira( id: "1", nome: "Marcia",  
    Estoque estoque = new Estoque();  
  
    System.out.println("Processo de gerenciar o estoque");  
    Roupa roupa = estilista.projetaRoupa( nome: "Vestido Vermelho");  
    Roupa roupaProduzida = costureira.criaRoupa(roupa);  
    costureira.finalizarProducaoDaRoupa(roupaProduzida, estoque);  
    estoque.gerarRelatorioDePecasVendidas();  
    costureira.gerarRelatorioDeRoupasProduzidas();  
}
```



Obrigado!

Integrantes:

Guilherme Rafael

Guilherme Dias

