

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Rodrigo Castro Vieira Gomes, Lucas Ricardo João Otávio Manieri e Rodrigo Martins
12021BSI234, 11511BSI274, 12021BSI263, 32011BSI030

Relatório Do Projeto Final: Sistema para Prestadora de Serviços de Limpeza

Uberlândia-MG

2023

Rodrigo Castro Vieira Gomes, Lucas Ricardo João Otávio Manieri e Rodrigo Martins

12021BSI234, 11511BSI274, 12021BSI263, 32011BSI030

/

Relatório Do Projeto Final: Sistema de Gestão de Fábrica de Roupas

Trabalho apresentado no curso de Bacharel
de Sistemas de Informação na Matéria
Modelagem de Software da Universidade
Federal de Uberlândia

Professor Murillo Guimarães Carneiro

Sistema para Prestadora de Serviços de Limpeza

1- Apresentação:.....	4
2- Resumo das etapas:	5
1. Modelagem estrutural e dinâmica em alto nível de abstração:.....	5
3. Refinamento de casos de uso	7
4. Modelagem de estados associada à classe	7
5. Introdução de elementos do domínio da solução computacional;.....	7
7. Modelagem de algoritmos de métodos;	7
8. Geração de código e o desenvolvimento iterativo.....	7
3- Etapa 1. Modelagem estrutural e dinâmica em alto nível de abstração:	8
Diagrama de Classes:	8
Diagrama de casos de uso:	9
4- Etapa 2. Refinamento estrutural:	10
5- Etapa 3. Refinamento de casos de uso:.....	11
Diagrama de atividades:	11
6- Etapa 4. Modelagem de estados associada à classe.	13
Diagrama de máquina de estados:	14
7- Etapa 5. Introdução de elementos do domínio da solução computacional	14
8- Etapa 7. Modelagem de algoritmos de métodos	16
Diagramas de atividades:.....	17
9- Etapa 8. Geração de código e o desenvolvimento iterativo.	20
10- BIBLIOGRAFIA	21

1- Apresentação:

A firma "K. H. Doméstica" é uma prestadora de serviços de limpeza que busca um sistema automatizado para auxiliar na alocação de empregados aos pedidos de serviço e lidar com possíveis restrições durante a alocação.

No processo de atendimento, a atendente cadastra o cliente, solicitando informações como CPF/CNPJ, nome, endereço e telefone. Em seguida, um pedido de serviço é registrado, incluindo informações como nome do cliente, data de abertura, data de realização dos serviços, local e lista de serviços desejados com suas respectivas metragens quadradas.

A atendente consulta uma tabela de serviços, contendo códigos, descrições, valores por metro quadrado e duração por metro quadrado, para preencher o pedido com informações sobre valor total e duração máxima dos serviços.

Todos os empregados possuem habilidades para executar determinados tipos de serviço, mas nem todos podem realizar todos os serviços. Caso a empresa não possua um empregado habilitado para um serviço solicitado, é providenciado um treinamento específico antes de alocar um empregado ao pedido.

Durante a alocação dos empregados, é verificado se eles já estão alocados para outros pedidos na mesma data e se possuem habilidades adequadas para os serviços solicitados. A alocação é considerada completa quando pelo menos um empregado habilitado está alocado para cada tipo de serviço solicitado.

O sistema gera relatórios, incluindo:

1-Pedidos com alocação incompleta, listando o número do pedido, código do cliente, nome/razão social do cliente, data de realização e lista de serviços exigidos com suas metragens quadradas.

2-Lista de empregados habilitados em um determinado tipo de serviço e que não estão alocados em um pedido em uma determinada data.

3-Agenda de um determinado empregado em um período específico, mostrando os horários alocados para cada dia, juntamente com o tipo de serviço alocado.

4-Esses relatórios fornecem informações valiosas para o gerenciamento e acompanhamento dos serviços de limpeza realizados pela empresa.

Com a implementação desse sistema, a firma "K. H. Doméstica" poderá otimizar a alocação de empregados, melhorar a eficiência operacional e proporcionar um melhor atendimento aos clientes.

2- Resumo das etapas:

1. Modelagem estrutural e dinâmica em alto nível de abstração:

Na etapa 1 foi modelado o diagrama de classes das seguintes classes:

- > Funcionário
- > Atendente
- > Limpador
- > Sistema
- > Equipe
- > Cliente
- > Cliente Pessoa Jurídica
- > Cliente Pessoa Física

Lembrando que a modelagem de classes nesse primeiro esforço apresenta somente o nome da classe e seus relacionamento com as demais (se existente).

No diagrama de casos de uso, é necessário identificar as funcionalidades do sistema e os atores em que são atribuídas essas funcionalidades.

Sendo assim, foram identificadas os seguintes casos de uso e atores.

Atores:

- > Ator Cliente
- > Ator Atendente
- > Ator Limpador
- > Ator Equipe

Lembrando que o sistema NÃO é caracterizado como um ator.

Casos de uso:

- > Solicita serviço (Cliente)
- > Realiza pagamento (Cliente)
- > Avaliar Serviços (Cliente)
- > Verifica Cadastro Cliente (Atendente)
- > Cadastra Cliente (Atendente)
- > Preenchimento Pedido (Atendente)
- > Informa Cliente sobre Pedido (Atendente)
- > Realiza Serviço (Limpador)

- > Realiza Treinamento (Limpador)

- > Aloca à um serviço

2. Refinamento estrutural:

Na segunda etapa, refinamento estrutural, o objetivo é identificar os atributos das classes, isto é, atribuir as características às classes.

Pode ser feito de duas formas, olhando para as classes e pensar nas características necessárias para desenvolver o sistema ou observando os casos de uso e localizando atributos que são necessários para que esse caso de uso.

Foram identificados então os seguintes atributos para as seguintes classes:

- > Cliente:

- > Nome

- > Telefone

- > Endereço

- > Cliente Pessoa Jurídica:

- > CGC

- > Razão Social

- > Cliente Pessoa Física:

- > Email

- > Funcionário:

- > Nome

- > Telefone

- > Email

- > Salario

- > Atendente:

- > feedBack Atendimento:

- > Limpador:

- > comissão: int

- > Serviço:

- > Nome

- > Descrição

- > Valor por Metro²

- > Duração por Metro²

> Pedido:

> Duração

> Valor Unitário

> Cliente

> Atendente

> Valor Total

3. Refinamento de casos de uso

Na terceira etapa, refinamento de casos de uso, é necessário identificar os métodos atribuídos às classes, utilizando o diagrama de atividades (cenário principal) e de sequência.

Foram identificados então os métodos:

Classe Atendente: + cadastrarCliente();+ verificaDadosCliente();+ consultaTabelaServiços();+preenchePedido();+ informaCliente();

Classe Limpador: + realizaTreinamento();+ realizaServiço();+calculaSalario();

Classe Cliente: +realizarPagamento(pedido Pedido);

4. Modelagem de estados associada à classe

Na quarta etapa: Modelamos o estado de um serviço no nosso sistema e suas transições.

5. Introdução de elementos do domínio da solução computacional;

7. Modelagem de algoritmos de métodos;

8. Geração de código e o desenvolvimento iterativo

3- Etapa 1. Modelagem estrutural e dinâmica em alto nível de abstração:

A modelagem estrutural e dinâmica em alto nível de abstração é uma abordagem utilizada para descrever e analisar sistemas complexos de maneira simplificada, fornecendo uma visão ampla do sistema em questão. Essa abordagem visa capturar os principais elementos e interações do sistema, sem entrar em detalhes específicos ou implementações técnicas.

A modelagem estrutural em alto nível de abstração se concentra na representação dos componentes principais do sistema e nas relações entre eles. Ela envolve a criação de diagramas e representações gráficas que mostram como os diferentes elementos se relacionam e interagem entre si. Esses diagramas podem incluir blocos, setas, linhas e outros símbolos para representar os componentes e suas conexões.

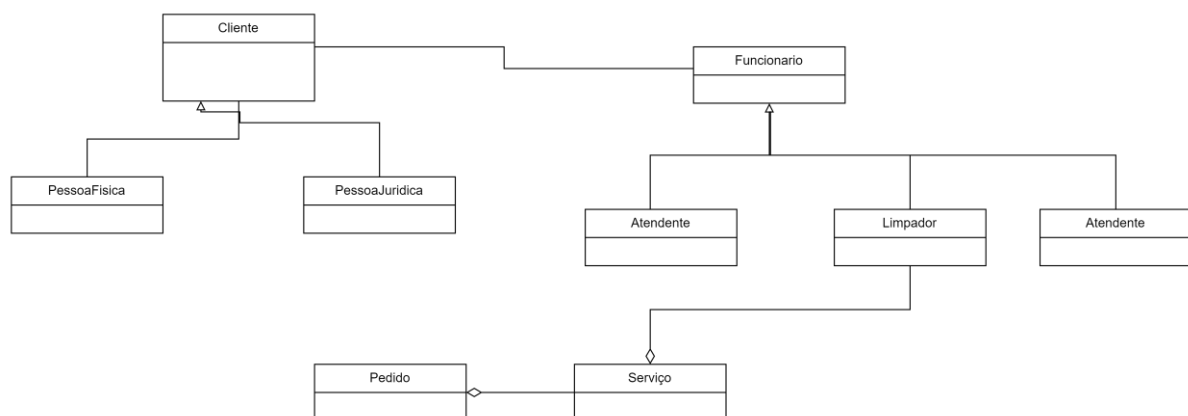
A modelagem dinâmica em alto nível de abstração, por outro lado, lida com a representação do comportamento e das interações entre os componentes do sistema ao longo do tempo. Ela envolve a descrição de como o sistema responde a estímulos externos e como os componentes internos interagem entre si para produzir resultados específicos. Essa modelagem pode ser feita por meio de diagramas de fluxo, simulações ou outras técnicas que ajudam a visualizar o comportamento dinâmico do sistema.

A modelagem estrutural e dinâmica em alto nível de abstração é útil em várias áreas, como engenharia de software, engenharia de sistemas, arquitetura de computadores e design de sistemas complexos. Ela permite que os projetistas e engenheiros compreendam melhor a arquitetura e o comportamento dos sistemas, identifiquem possíveis problemas ou pontos de melhoria e testem diferentes cenários antes de investir tempo e recursos na implementação completa.

Ao usar essa abordagem, é importante ter em mente que a modelagem em alto nível de abstração é uma simplificação do sistema real e pode não capturar todos os detalhes ou nuances. No entanto, ela fornece uma representação concisa e acessível, permitindo uma compreensão geral do sistema e auxiliando na tomada de decisões e na comunicação eficaz entre os membros da equipe.

Na primeira etapa foram construídos os seguintes diagramas:

Diagrama de Classes:



As classes Pessoa Física e Jurídica herdam Cliente, pois são dois tipos diferentes de clientes no sistema.

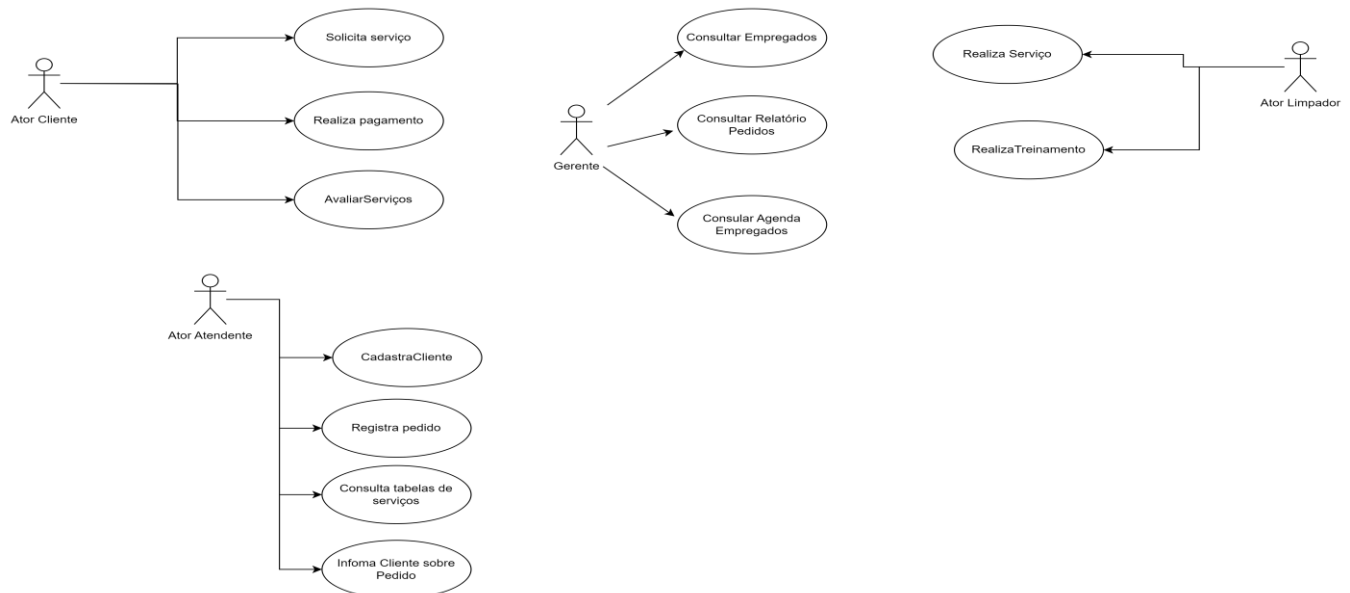
As classes atendente, gerente e limpador herdam de funcionário e se diferenciam nas suas funções:

O atendente é responsável pela maior parte de contato com o sistema, o cadastro do cliente até a realização do serviço. O limpador realiza os serviços e os treinamentos necessários. O gerente é responsável por fazer análise dos diversos relatórios e gerir a empresa da melhor maneira.

Cada limpador pode realizar um ou mais serviços (agregação, pois limpador não depende do serviço para existir), e um pedido pode receber um ou mais tipos de serviços

Exemplo: um pedido contendo dois serviços, limpeza de piscina e aspirar ambiente interno, possuindo dois limpadores.

Diagrama de casos de uso:



Foram identificados os atores:

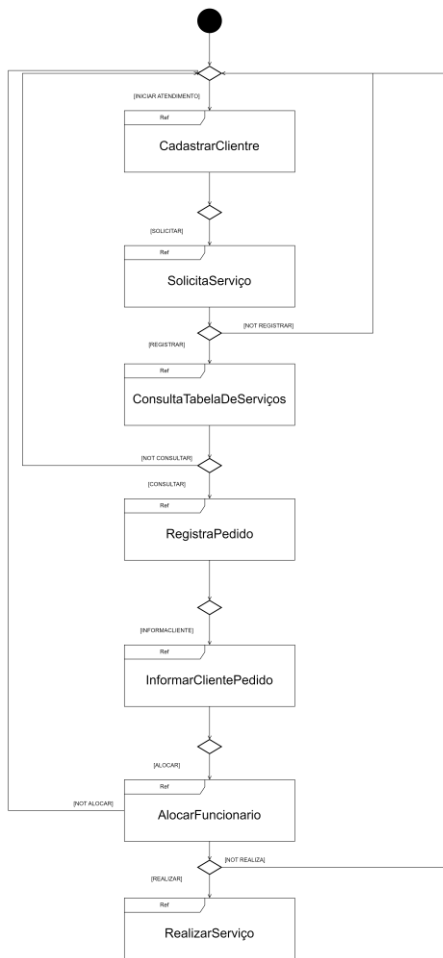
Cliente: possui os casos de uso de solicitar um serviço (telefonema para a empresa), realizar pagamento (por meio do envio de link de pagamento do sistema).

Atendente: cadastrar cliente (no sistema), consultar tabelas de serviço (sistema), registrar pedido (sistema), informar cliente sobre pedido (envio de e-mail avisando sobre link de pagamento).

Gerente: Consultar empregados e exibir todas as informações, consultar relatório de todos os pedidos e seus respectivos status, consultar agenda empregados (ver qual funcionário que está disponível para realizar determinado serviço)

Limpador: Realiza o serviço logo após ser alocado a um pedido e realiza treinamento de um determinado serviço quando nenhum funcionário está disponível ou habilitado a realizar.

Diagrama de visão geral de interação:

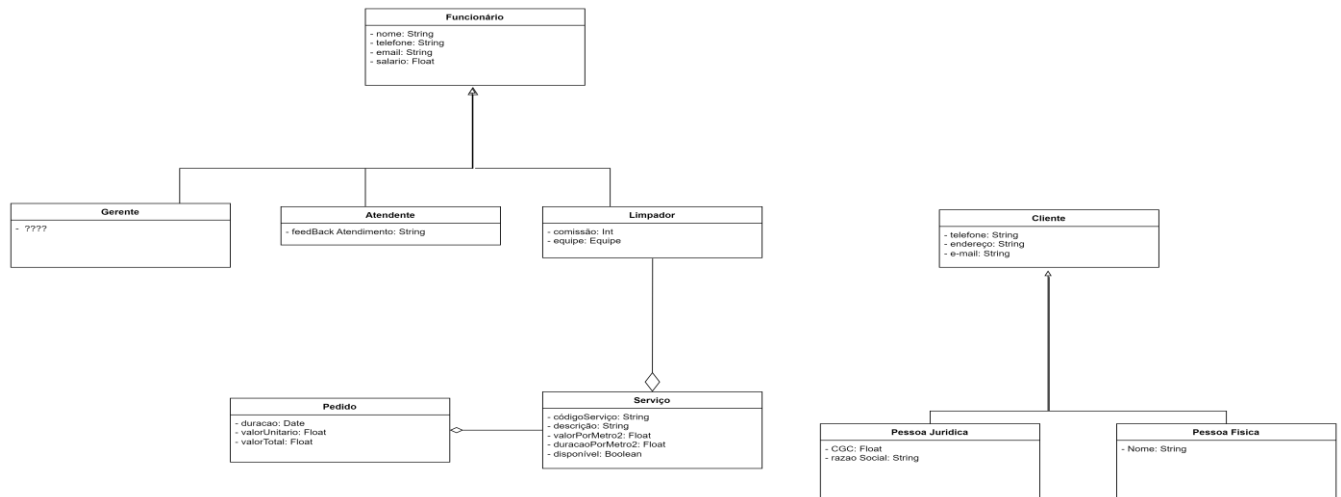


4- Etapa 2. Refinamento estrutural:

Na Etapa 2, o objetivo é identificar as características de uma classe, ou seja, seus atributos.

Para que isso seja possível, olhamos para cada classe e nos perguntamos, quais as características dessa classe que é crucial para o sistema funcionar. Como por exemplo, o CPF de um cliente pessoa física é necessário para cadastrar um Cliente, ou o valor de um pedido, para realizar o pagamento,

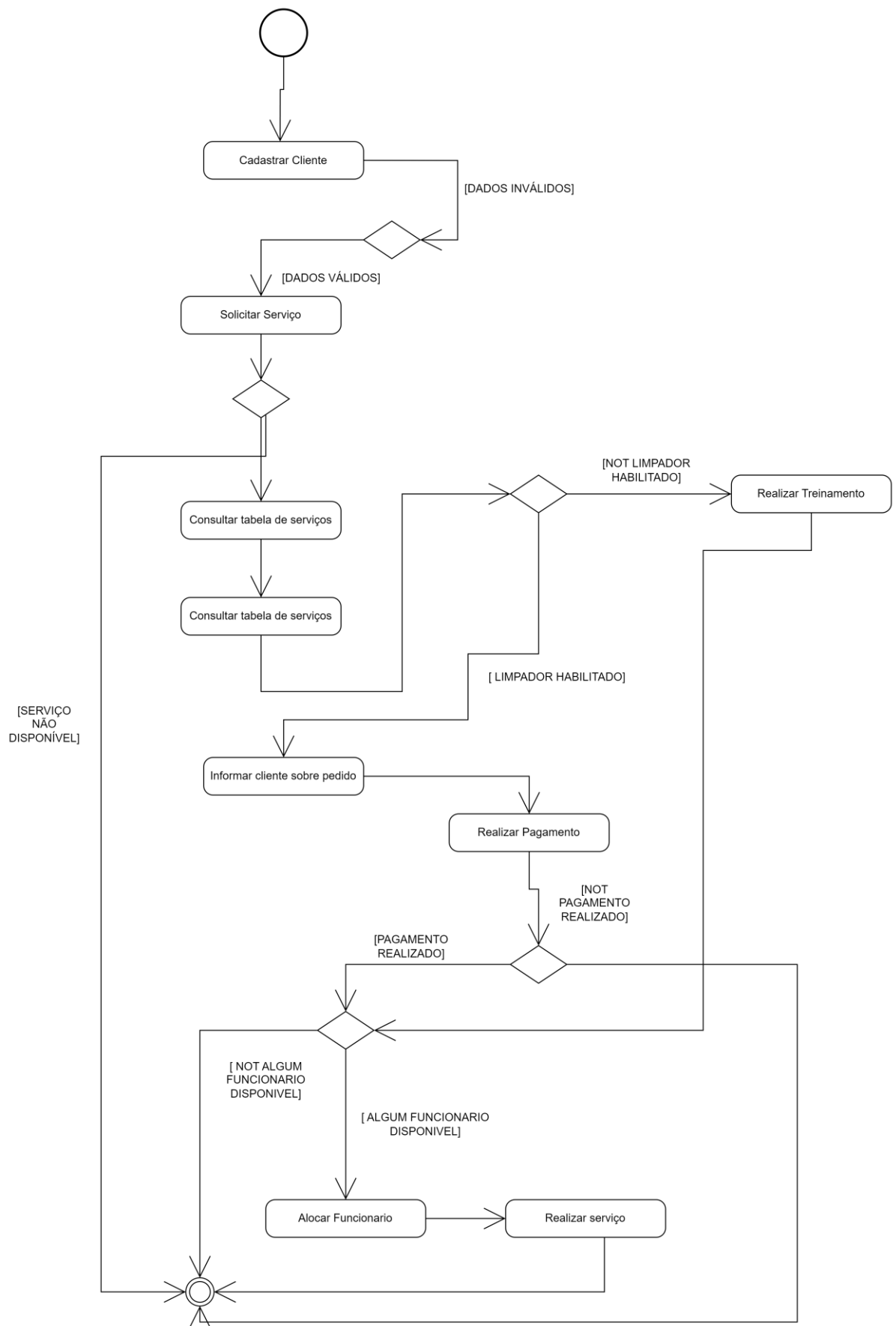
etc. Sendo assim foi construído o seguinte diagrama (ainda faltando coisas):



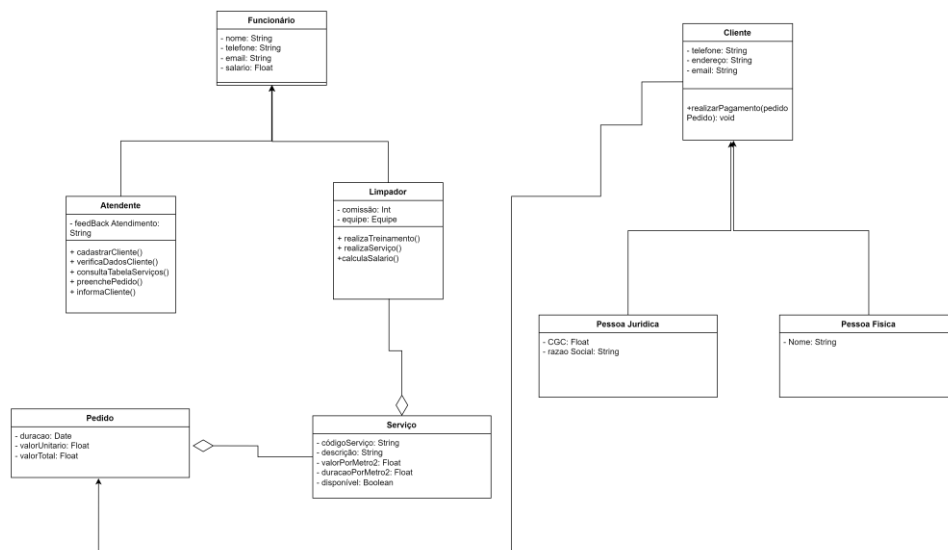
5- Etapa 3. Refinamento de casos de uso:

Na etapa 3, é utilizado o diagrama de atividades (Cenário Principal) com a finalidade de refinar os casos de usos e localizar os métodos que cada classe possui e a ordem que esses casos de uso ocorrem. Foram identificados, além dos métodos já possuintes na ideia principal do sistema nos casos de uso, o método `alocarFuncionários(funcionário Funcionario, servic Serviço)` da classe `Gerente` e o atributo `preço` do serviço para saber o preço total do pedido.

Diagrama de atividades:



Métodos identificados no diagrama de classes:



Métodos como o `realizarPagamento()` que identificamos depois e localizamos sua ordem no nosso sistema.

6- Etapa 4. Modelagem de estados associada à classe.

Na etapa 4 é necessário modelar os estados de cada classe, isto é, uma técnica utilizada para descrever o comportamento de um sistema de software através da identificação e representação dos diferentes estados que ele pode assumir. Os estados são definidos pelas condições internas e externas do sistema, e as transições entre os estados são determinadas pelas ações e eventos que ocorrem. Essa modelagem é útil para compreender e visualizar o fluxo de controle e a lógica do software, facilitando o processo de análise, design e implementação do sistema. o segundo diagrama de estado e sobre o caso de uso realizar serviço aonde e identificado todo o caminho para a realização de um serviço desde a solicitação feita pelo cliente até o ato de realizar o serviço, foi uma decisão de projeto colocar o estado pagamento realizado antes de realizar o serviço, do mesmo modo que escolhemos não no aprofundar no modo de pagamento deixando o diagrama só no caso em que o pagamento é realizado, também não no aprofundamos no estado no limpado pois não era o objetivo do diagrama, o estado informar cliente sobre o pedido faz parte do caminho até a realização do serviço. Caso o serviço não esteja disponível o estado de serviço termina.

Modelamos os estados do nosso sistema em específico olhando para o estado de um serviço, isto é, todas as fases em que um serviço passa em sua criação até destruição.

Diagrama de máquina de estados limpador:

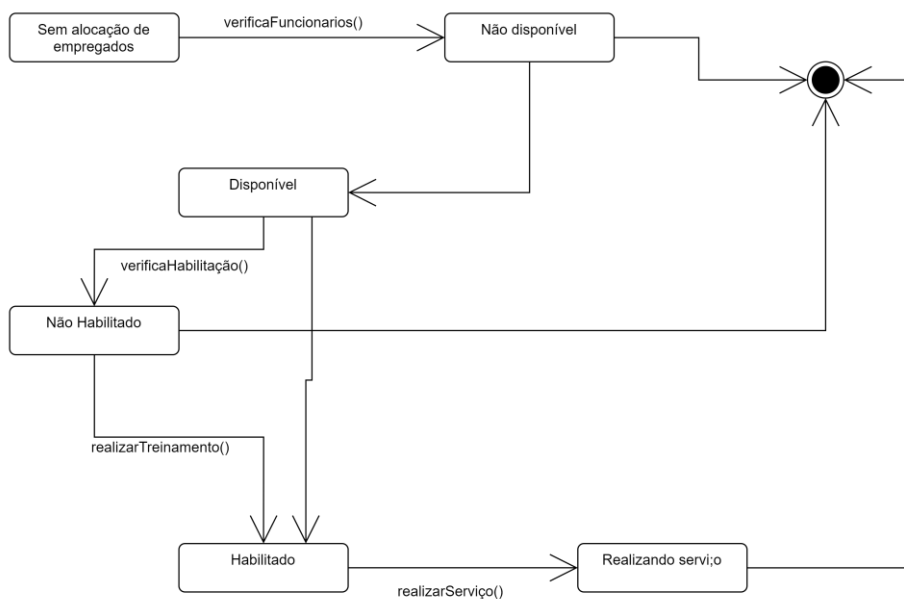


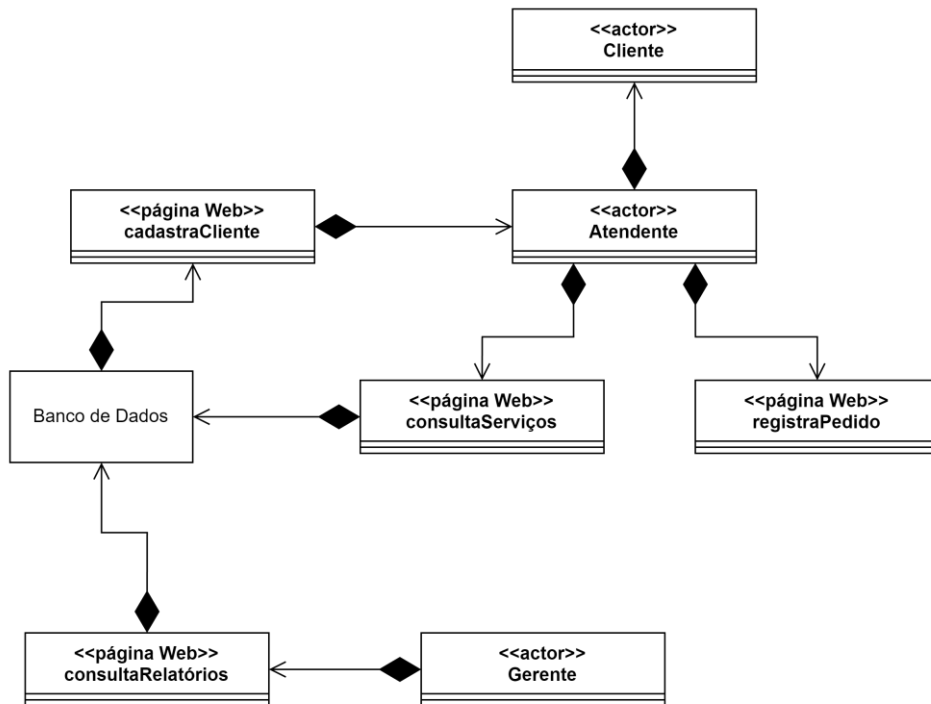
Diagrama de máquina de estado serviço:

7- Etapa 5. Introdução de elementos do domínio da solução computacional

Comportamento é uma etapa da modelagem de software em que os elementos relevantes do problema são representados. Isso inclui conceitos, entidades e regras de negócio. Através de diagramas e modelos, esses elementos são descritos para entender o contexto do problema e identificar requisitos e funcionalidades necessárias para a solução. Essa modelagem ajuda a definir a arquitetura e guiar o desenvolvimento do software de maneira mais eficiente.

É a etapa que itera sobre as demais etapas já realizadas e pensa no que vai ser importante pensando em questões computacionais e ver o que é de melhor aproveitamento do sistema.

Diagrama de classes do sistema pensando na solução computacional e analisando os atributos e principalmente métodos atribuídos à cada classe:



Não foi feito o diagrama de atividades ou até mesmo de sequência, mas sabemos da importância desses diagramas no projeto de um software.

8- Etapa 7. Modelagem de algoritmos de métodos

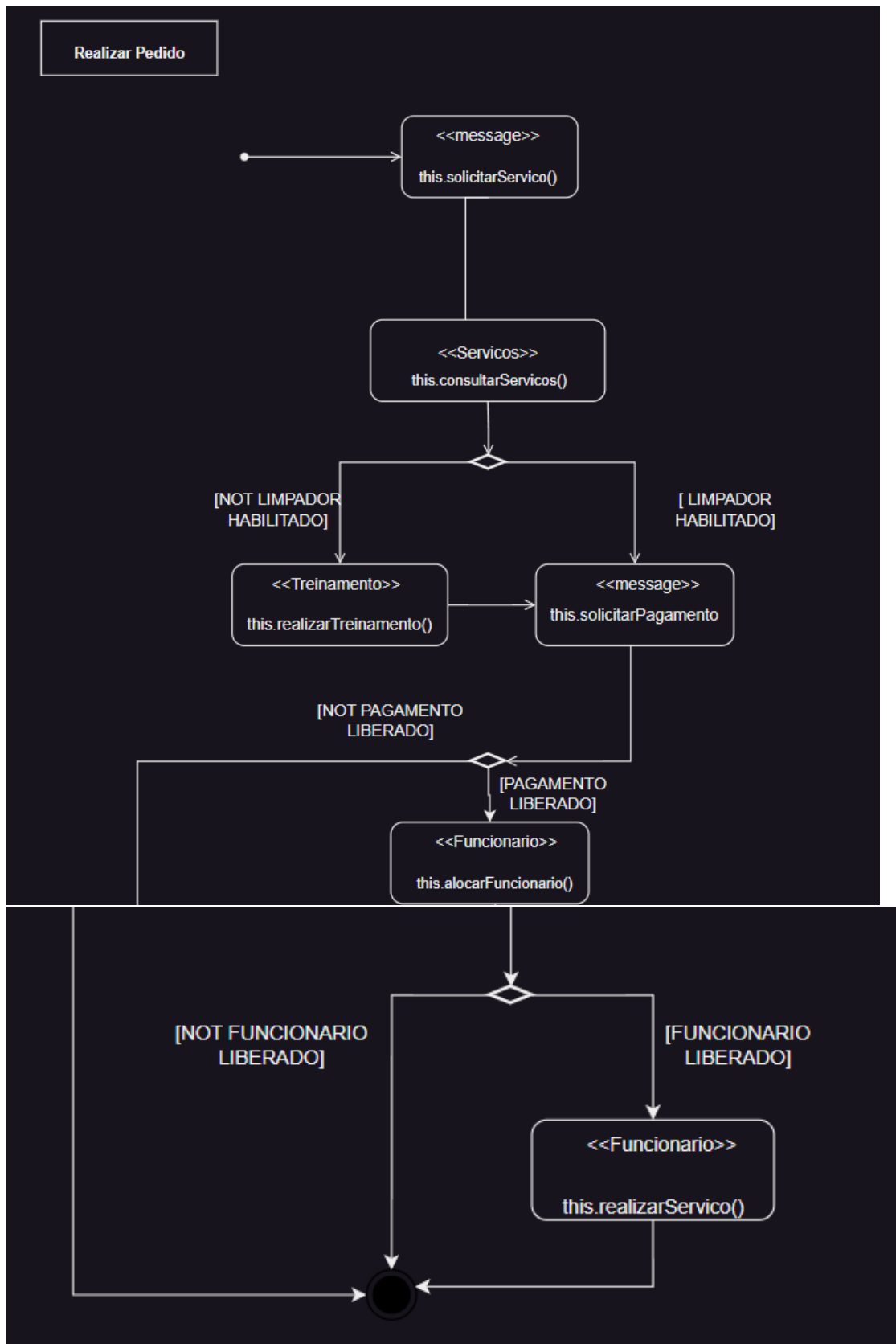
A Etapa 7 - Modelagem de algoritmos de métodos é uma parte essencial da modelagem de um sistema. Nessa etapa, são definidas as funções e procedimentos que compõem o sistema, descrevendo como cada uma delas opera e se relaciona com as outras.

A função da Etapa 7 é detalhar a lógica e o fluxo de execução dos métodos que serão implementados no sistema. Ela permite uma compreensão mais clara e precisa de como as operações serão realizadas, quais dados serão manipulados, quais cálculos serão feitos e quais são os resultados esperados.

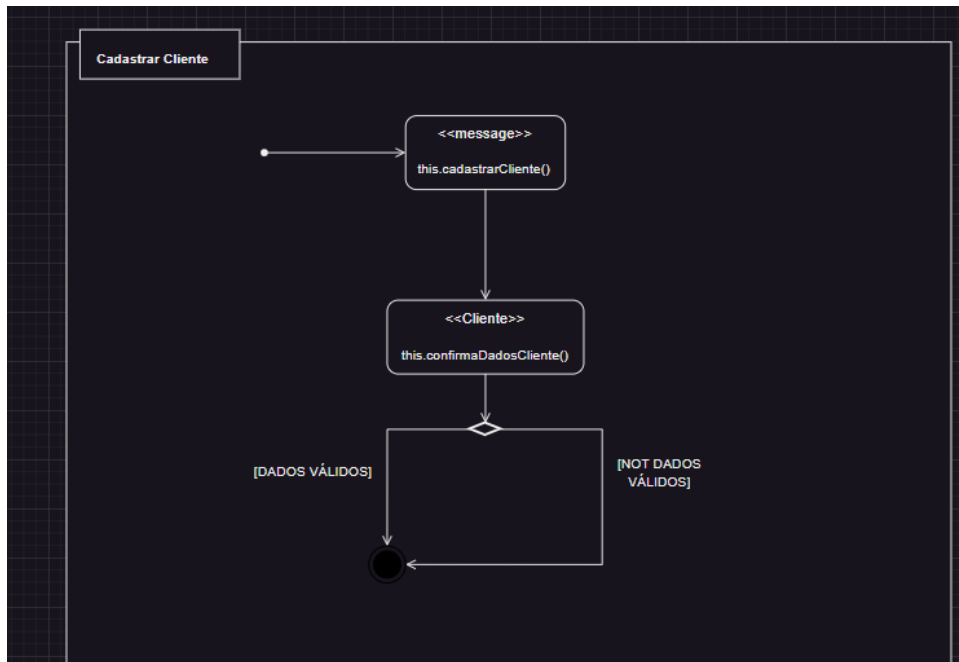
Durante a modelagem de algoritmos de métodos, são definidos os parâmetros de entrada e saída de cada função, bem como as operações que serão executadas dentro delas. Essas funções são representadas por meio de pseudocódigo, fluxogramas ou outras formas de representação que facilitem a compreensão.

Ao concluir a Etapa 7, o modelo de sistema terá uma descrição detalhada de cada método, fornecendo uma base sólida para a implementação do sistema em uma linguagem de programação específica. A modelagem dos algoritmos de métodos permite que os desenvolvedores tenham uma visão clara das tarefas a serem executadas e das interações entre os componentes do sistema, facilitando o processo de implementação e garantindo um desenvolvimento mais eficiente e sem ambiguidades.

Diagramas de atividades:

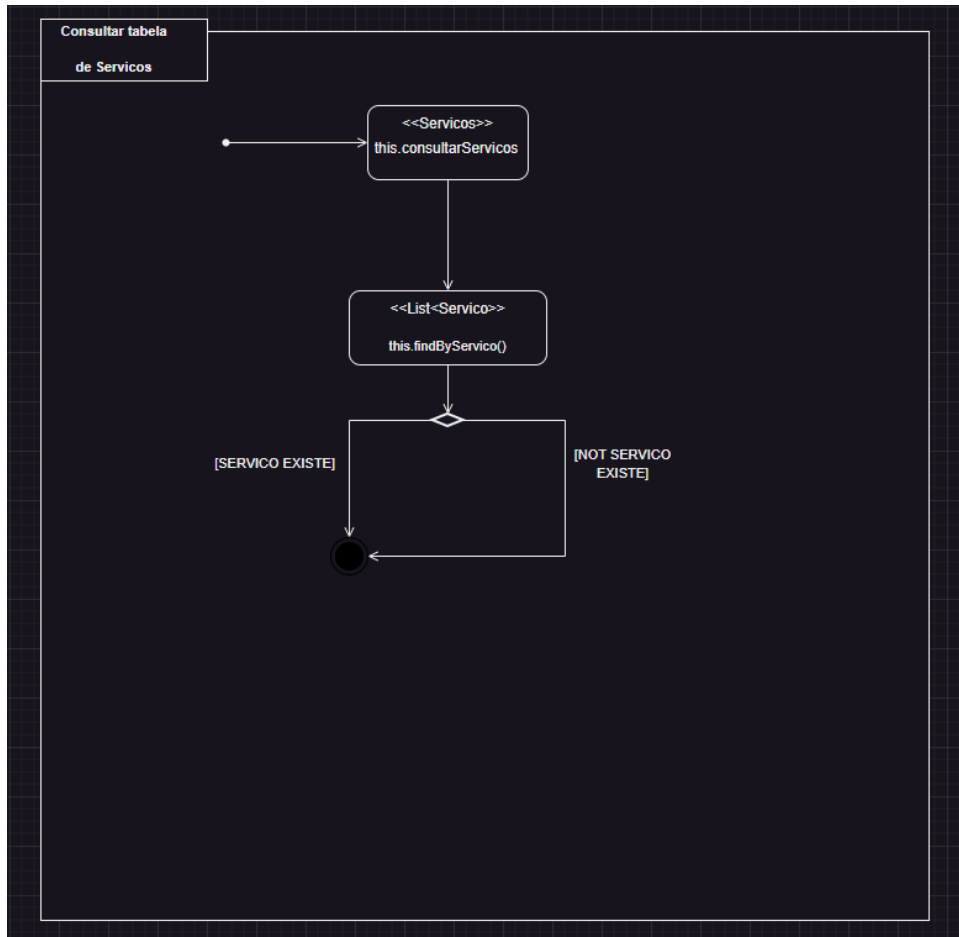


No exemplo acima vemos o método solicitarServico, que verifica se há um limpador habilitado para aquele serviço, caso não tenha realiza o treinamento para seguir o fluxo. Após essa confirmação solicita o pagamento para o cliente, que após liberado aloca o funcionário e realiza o serviço, mostrando assim o fluxo desde o início até a finalização do serviço solicitado.

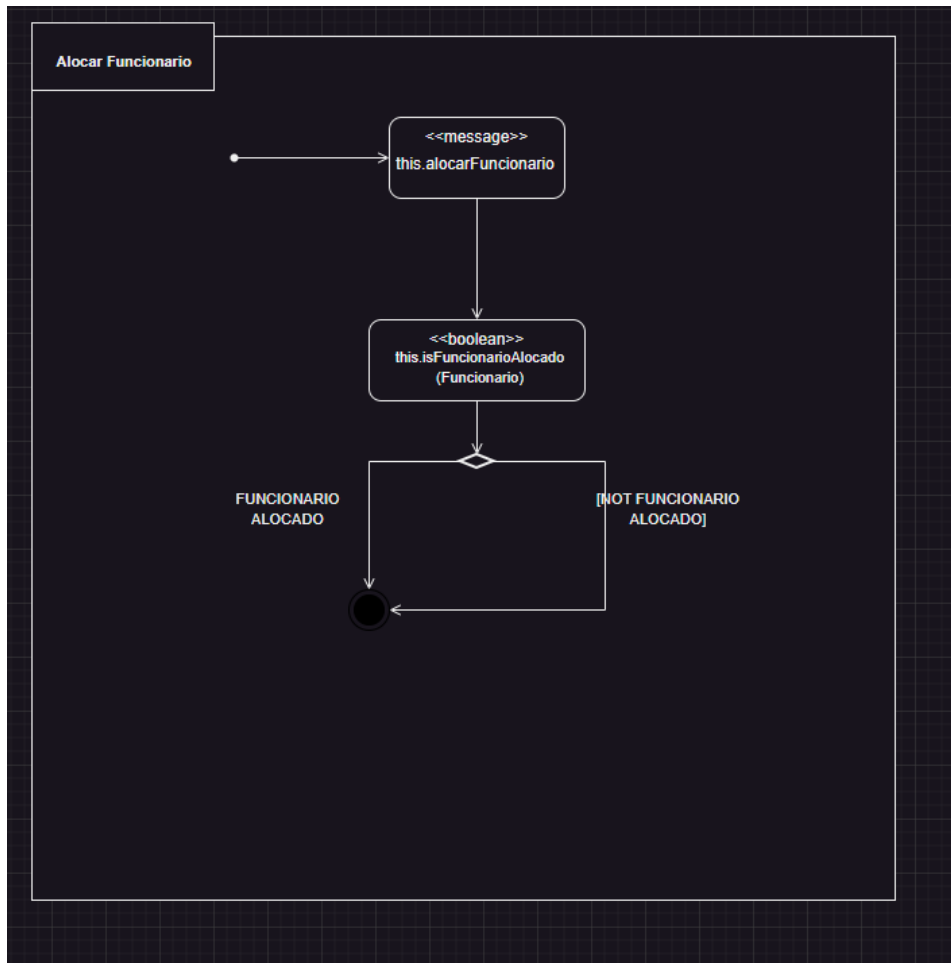


Já aqui temos um método mais simples, de cadastra o cliente, que é um método que apenas pega os dados do cliente para cadastrar e confirma esses dados fazendo as verificações necessárias.

Abaixo temos outro método, uma simples busca por Serviço no banco de dados para verificar se aquele serviço está disponível ou não, se existe ou não, e o restantes dos dados relacionados ao serviço.



Logo abaixo temos o método alocar funcionário, que faz a verificação se o funcionário já está alocado ou não, caso não estiver aloca para o serviço solicitado.



9- Etapa 8. Geração de código e o desenvolvimento iterativo.

Na etapa 8 é realizado a geração de código em cima do diagrama de classes, em nosso trabalho decidimos fazer manualmente por ser apenas uma ideia simples de código , mas normalmente é utilizado programas auxiliares para facilitar a criação do código em sistemas mais complexos.Utilizamos a linguagem Java para a implementação do código , criamos as classes, seus relacionamentos, atributos e métodos.

Como por exemplo das classes Funcionário,Atendente e Limpador, Funcionário é a super classe e Limpador e Funcionário são as subclasses e herdam todos os métodos e atributos de Funcionário.

Nessa etapa também tivemos o cuidado da correspondência entre código e projeto.

10- BIBLIOGRAFIA

Slides disponibilizados pelo professor

Material disponibilizados pelo Prof. Ricardo Pereira e Silva (UFSC)

Booch, G.; Jacobson, I. e Rumbauch, J. UML: Guia do Usuário. Campus, 2006.

Silva, R. P. UML 2 em modelagem orientada a objetos. Visual Books, 2007.

Pereira e Silva, R. Aula 01 Introdução à modelagem OO (2014). Disponível em:
<https://youtu.be/C1ff8ul06BE>.

SILVA, Gizele. O que é arquitetura MVC?. coodesh.() Disponível em:
[https://coodesh.com/blog/dicionario/o-que-e-arquitetura-mvc/#:~:text=Basicamente%2C%20a%20arquitetura%20MVC%20%C3%A9,camada%20de%20control%20\(controller\)](https://coodesh.com/blog/dicionario/o-que-e-arquitetura-mvc/#:~:text=Basicamente%2C%20a%20arquitetura%20MVC%20%C3%A9,camada%20de%20control%20(controller).). Acesso em: 08/06/2023