

TRABALHO DE PS

SISTEMA ATELIER FLOR DE MACELA

**ALONSO BASSINI, LUISA STELLET, MIGUEL
JEHLE E NICOLE MEIRELES E RAFAEL ACCETTA**

CONTEXTO:

Breve Descrição do Sistema

- Loja de produtos veganos com operação física e online.
- Propósito: Centralizar operações e oferecer uma experiência completa de integração entre loja física e online para otimizar a experiência do cliente

Motivação para Escolha

- Necessidade de automação e organização de vendas, estoque e produção.
- Crescimento do mercado vegano e importância de uma plataforma eficiente.



OBJETIVOS DA ARQUITETURA

- Desempenho: Garantir tempos de resposta rápidos, mantendo baixa latência e eficiência na comunicação entre os serviços.
- Escalabilidade: Permitir que o sistema suporte um número crescente de usuários, pedidos e produtos sem perda de desempenho.
- Segurança: Assegurar proteção contra ameaças externas, implementação de autenticação robusta e conformidade com a LGPD.
- Disponibilidade: Garantir alta disponibilidade do sistema, evitando downtime e falhas críticas.

OBJETIVOS DA ARQUITETURA

- Manutenibilidade: Implementar um design modular que facilite atualizações, correções de erros e aprimoramentos contínuos.
- Usabilidade: Fornecer uma interface intuitiva que ofereça uma experiência fluida para usuários finais e administradores.
- Portabilidade: Assegurar compatibilidade entre diferentes dispositivos e navegadores sem perda de funcionalidade.
- Confiabilidade: Garantir que o sistema funcione corretamente sob diferentes condições e minimize falhas operacionais.
- Testabilidade: Desenvolver componentes que possam ser facilmente testados e validados para garantir qualidade contínua.

RESTRIÇÕES DA ARQUITETURA

- **Dependência de terceiros:** APIs de pagamento devem garantir conformidade com LGPD.
- **Tempo de resposta:** O site deve carregar em menos de 3 segundos para evitar altas taxas de desistência.
- **Compatibilidade:** Suporte total a dispositivos móveis e diferentes navegadores.
- **Custos:** A escolha de tecnologias deve considerar o orçamento disponível para desenvolvimento e manutenção.
- **Manutenção e Suporte:** A infraestrutura deve permitir atualizações e correções sem interrupções prolongadas.

DECISÕES ARQUITETURAIS

- Arquitetura em camadas
- MVC (model-view-controller)
- Cliente-Servidor

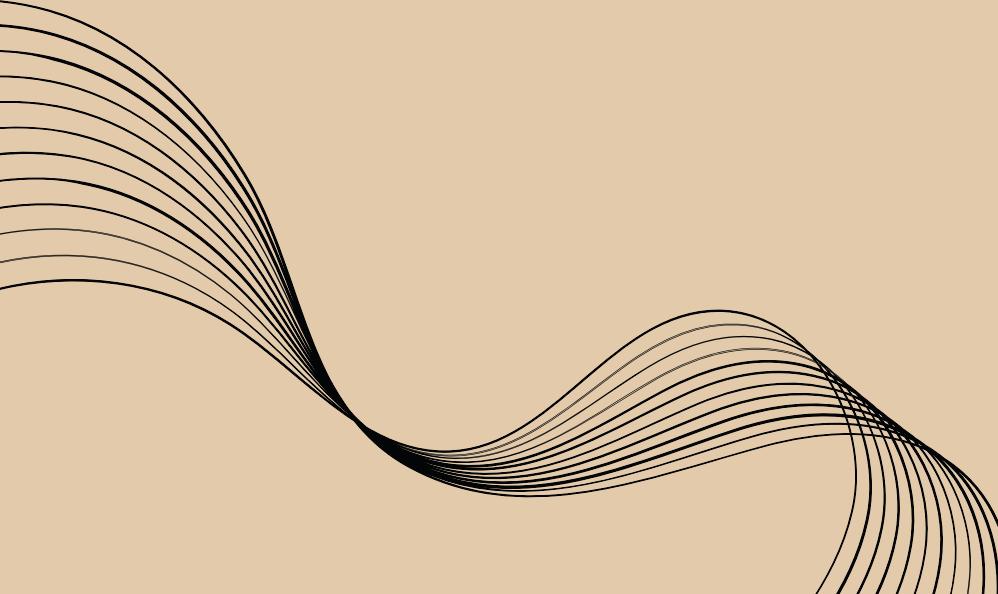
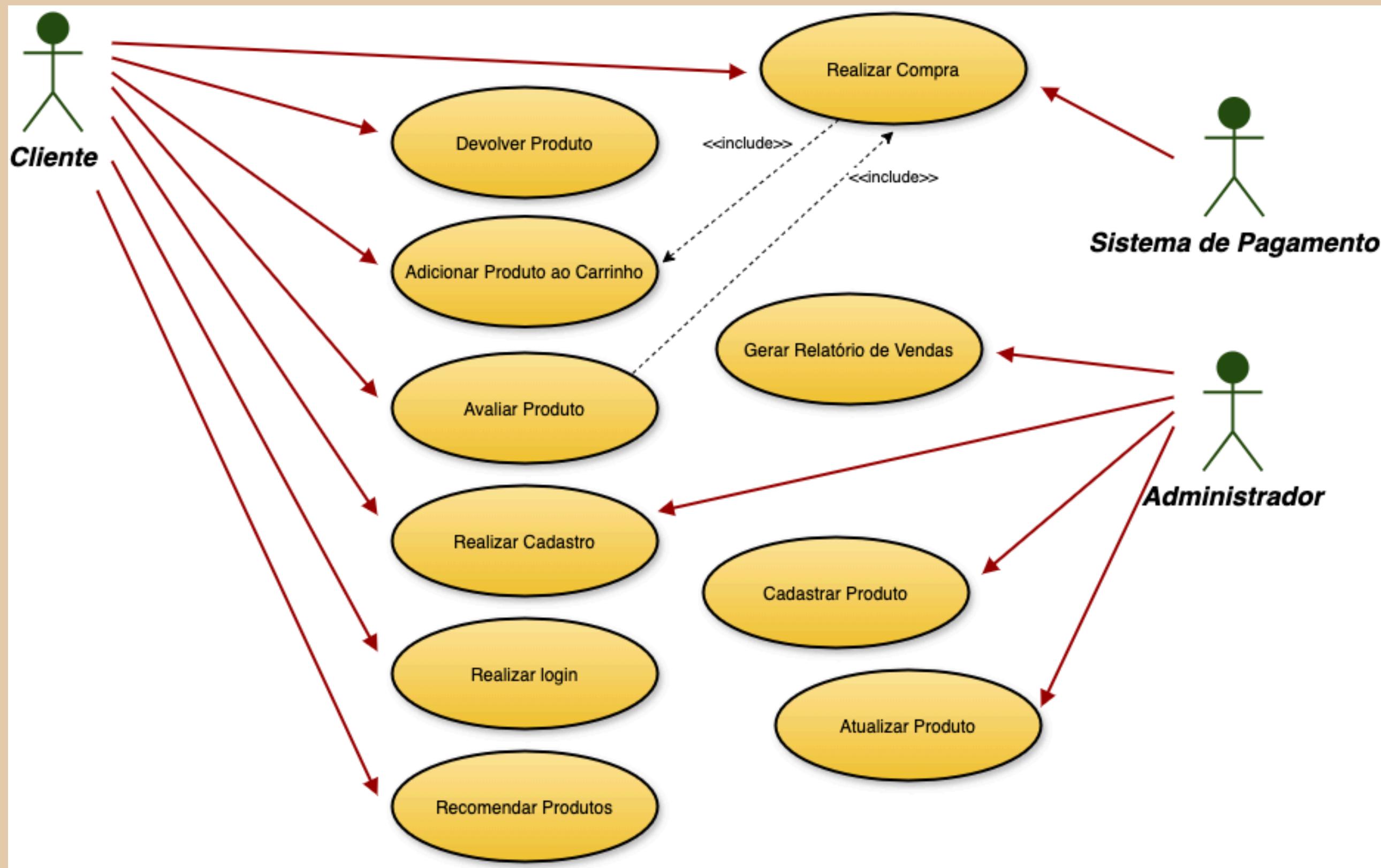


DIAGRAMA DE CASO DE USO



MODELO CONCEITUAL

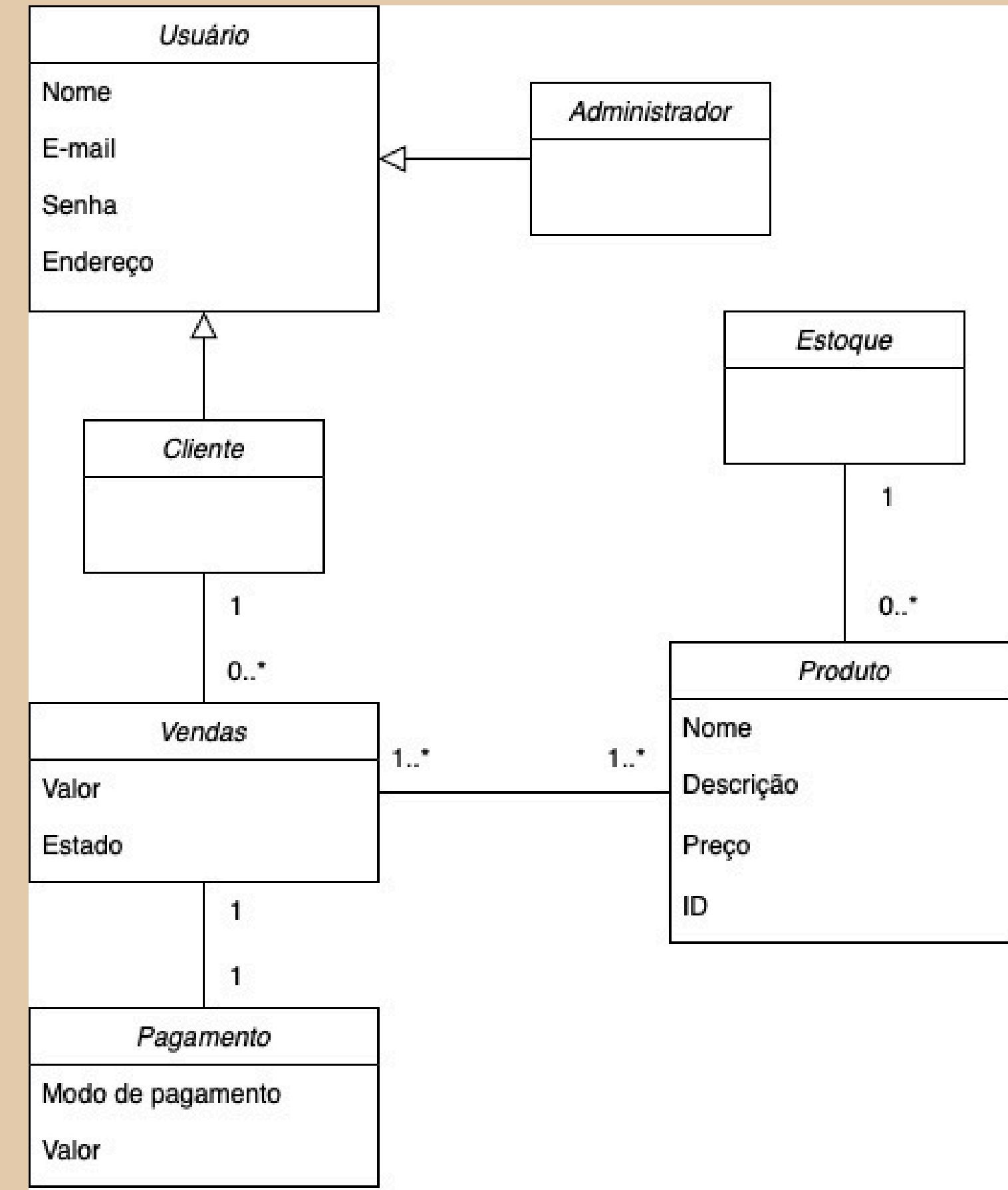
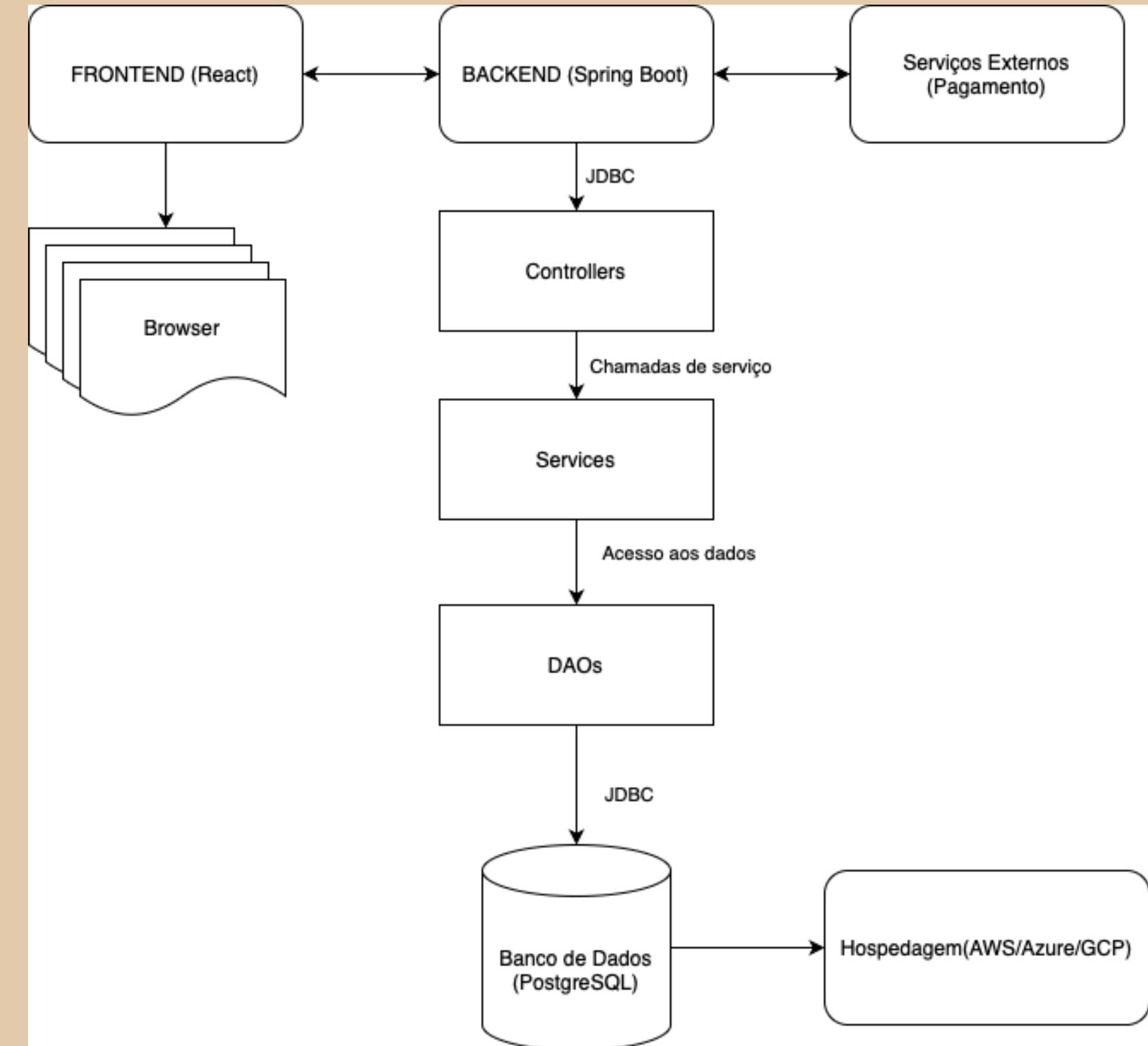


DIAGRAMA COM VISÃO GERAL DE ARQUITETURA DO SISTEMA



JUSTIFICATIVAS

- **Arquitetura em Camadas:** O sistema foi estruturado em camadas distintas (frontend, backend e banco de dados) para garantir uma separação clara de responsabilidades. Essa abordagem facilita a manutenção, permite a evolução modular do projeto e isola possíveis falhas, assegurando que mudanças em uma camada não afetem as outras desnecessariamente. Além disso, essa organização promove um desenvolvimento mais organizado e escalável.



JUSTIFICATIVAS

- **Modelo MVC (Model-View-Controller):** A aplicação adota o padrão MVC para definir responsabilidades específicas em cada componente. O frontend (React) atua como a View, responsável pela interface do usuário, enquanto o backend (Spring Boot) implementa os Controllers, que gerenciam as requisições e respostas. O Model, representado pelas entidades e DAOs, lida com a estrutura dos dados e a comunicação com o banco. Essa divisão melhora a reusabilidade do código e simplifica a colaboração entre equipes.

JUSTIFICATIVAS

Cliente-Servidor: A arquitetura cliente-servidor foi escolhida para separar claramente o frontend (cliente) do backend (servidor), permitindo que cada parte seja desenvolvida e escalada independentemente. O cliente consome APIs fornecidas pelo servidor, garantindo flexibilidade para futuras integrações ou migrações de hospedagem (como AWS, Azure ou GCP). Essa abordagem também favorece a segurança e o desempenho, distribuindo adequadamente as cargas de processamento.

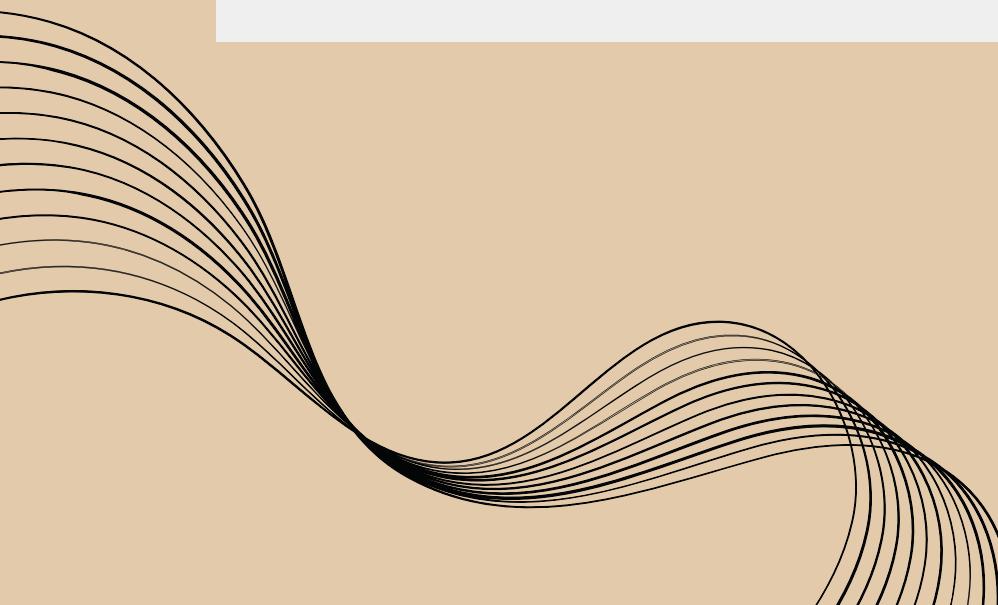
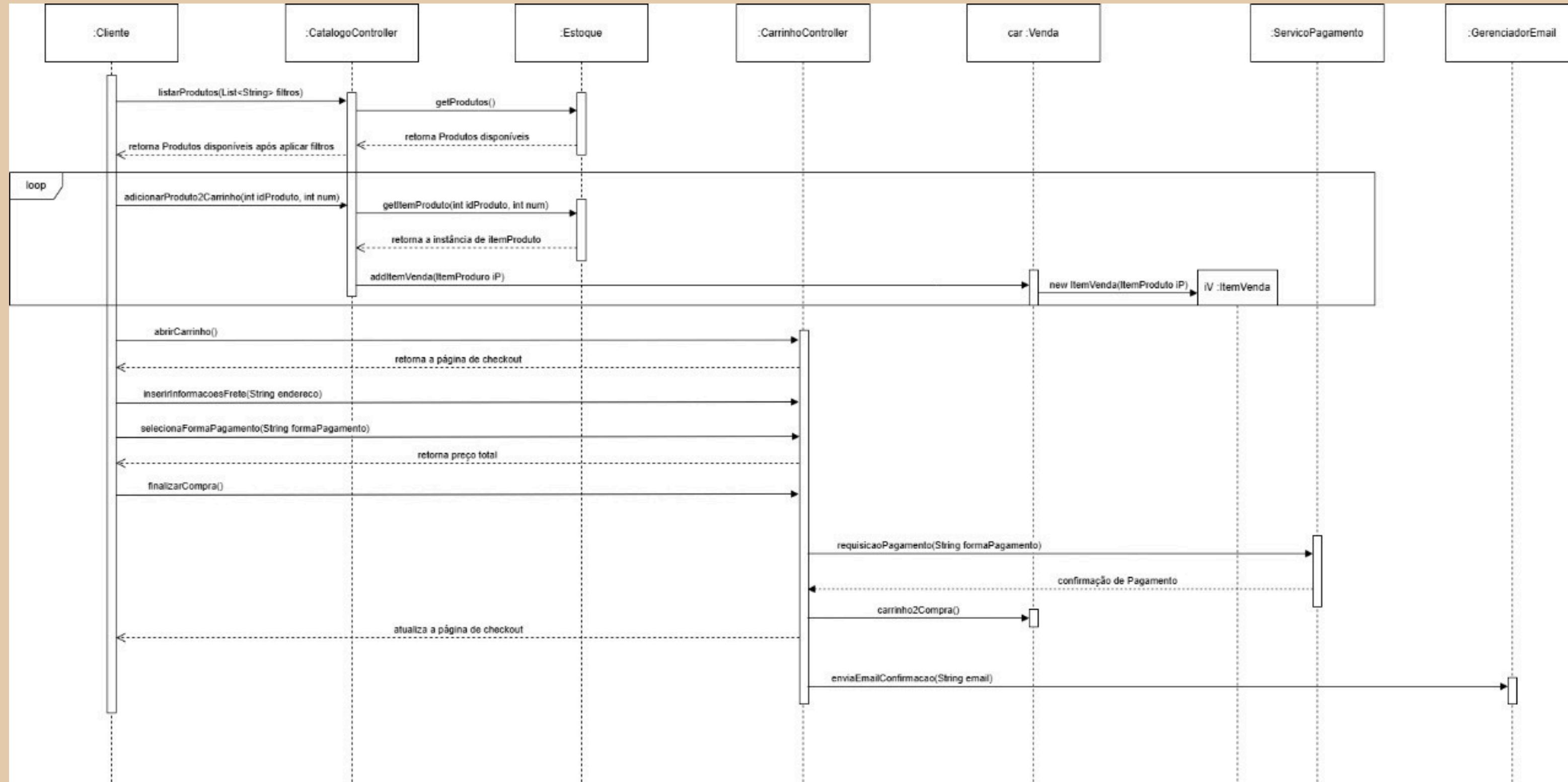


DIAGRAMA DE INTERAÇÃO - REALIZAR COMPRA



DSS - REALIZAR COMPRA

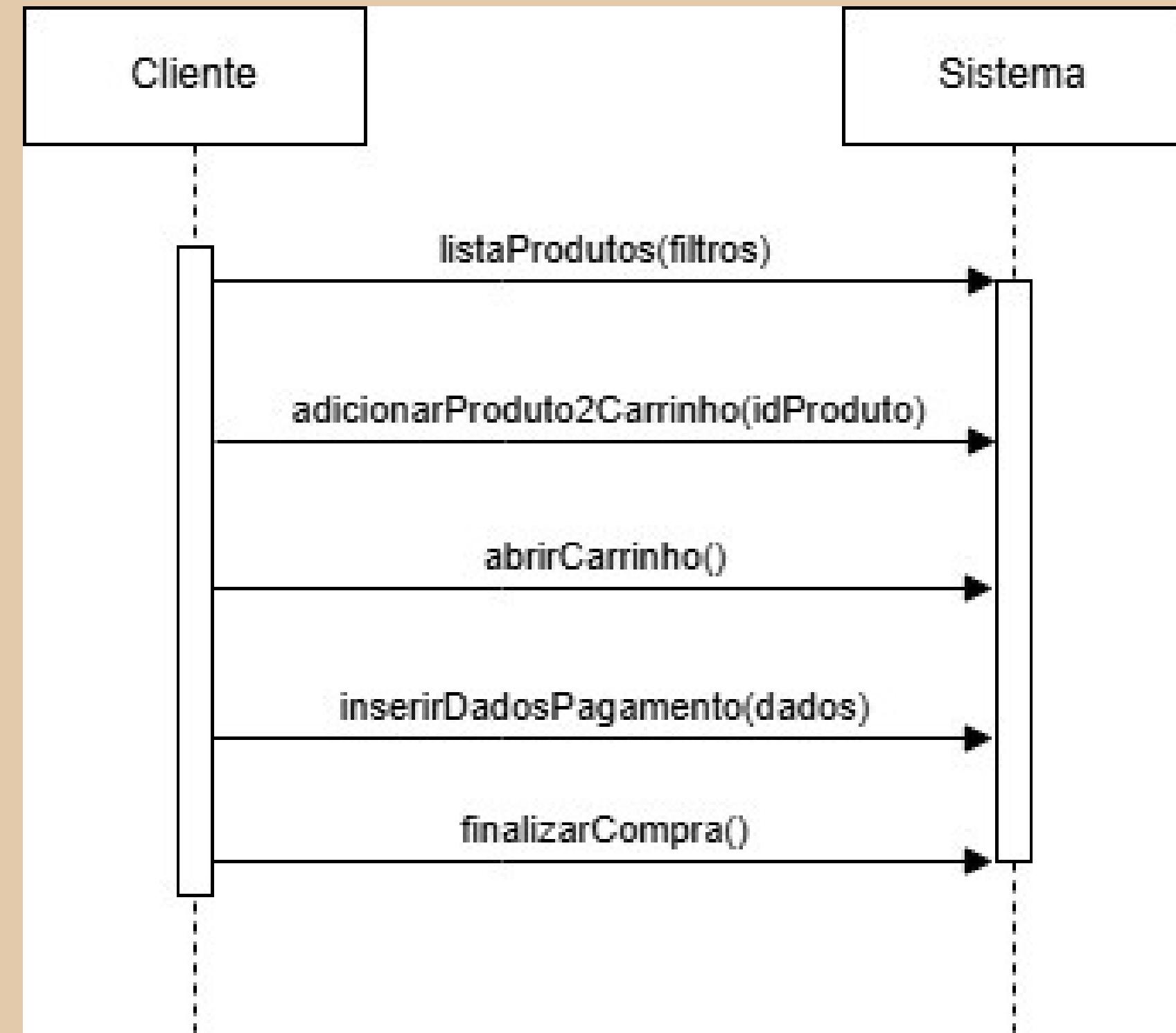
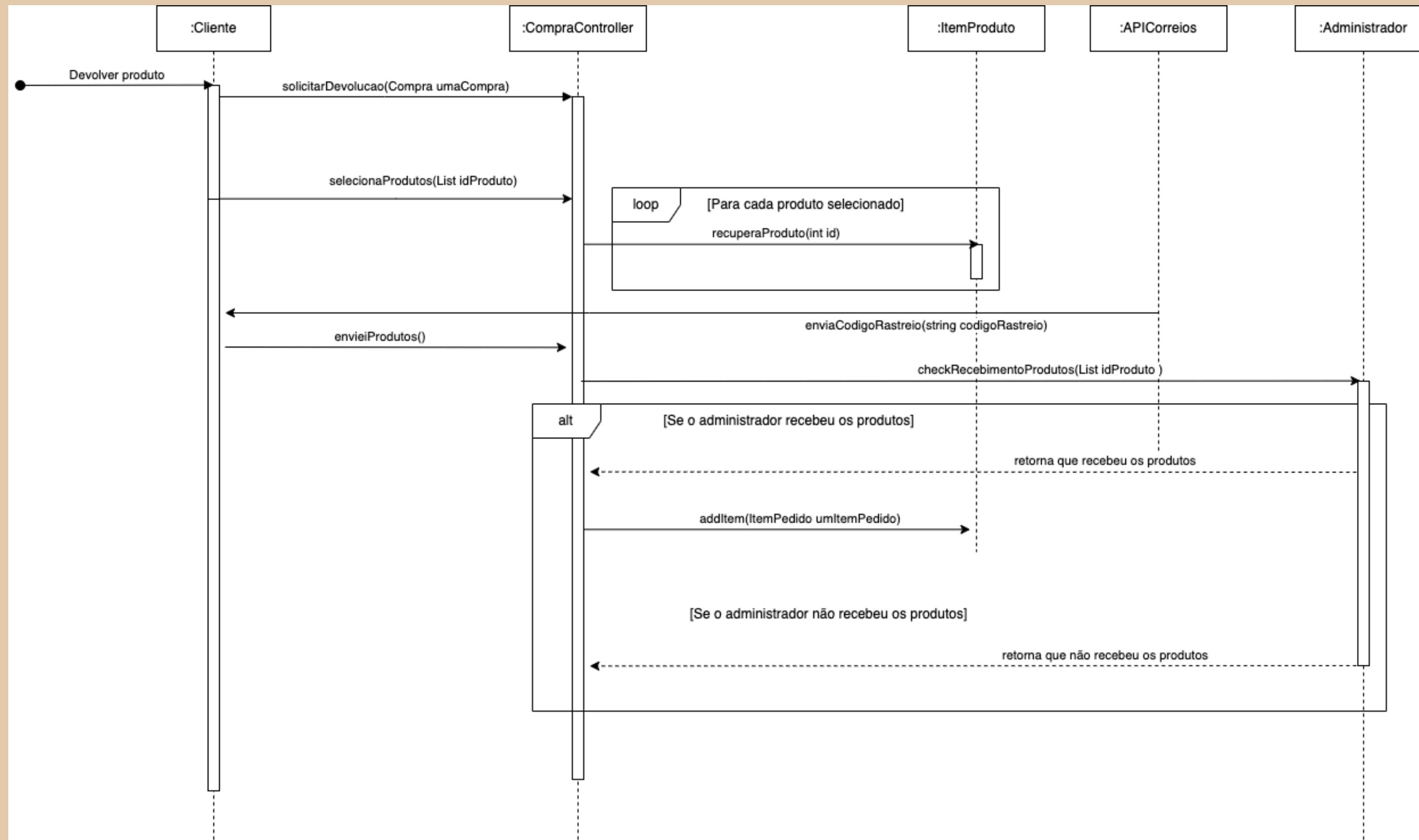


DIAGRAMA DE INTERAÇÃO - DEVOLVER PRODUTO



DSS - DEVOLVER PRODUTO

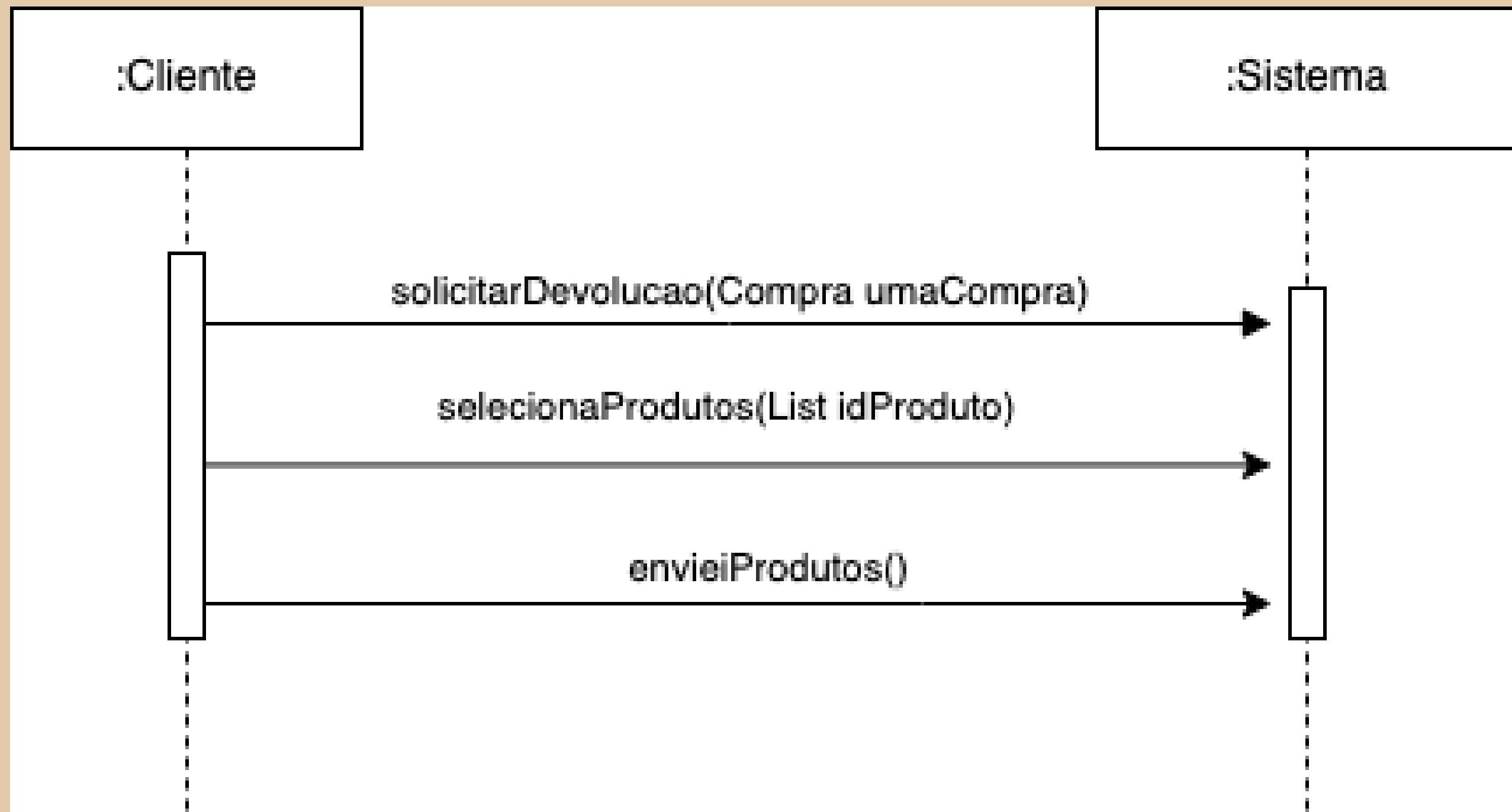
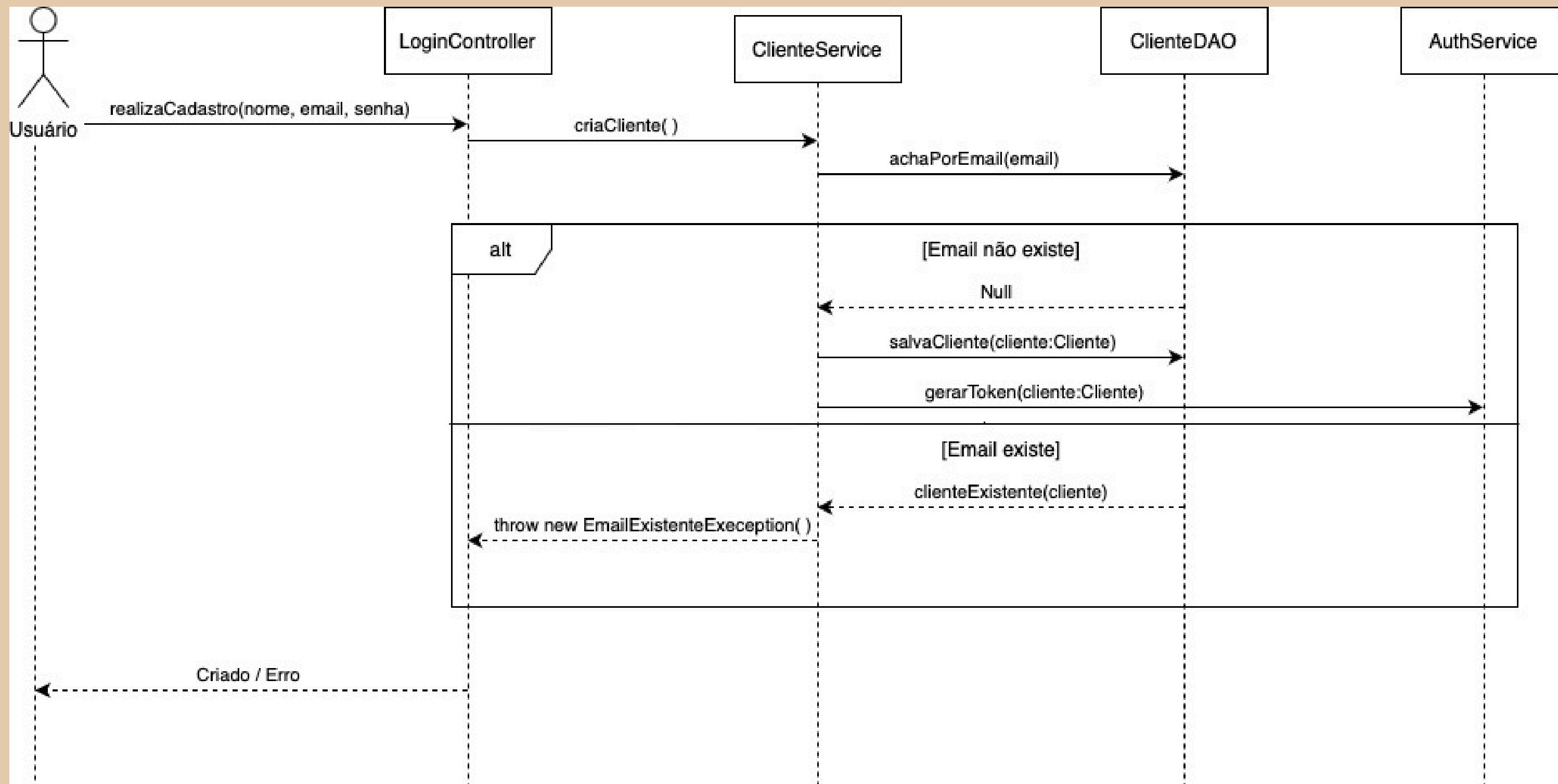


DIAGRAMA DE INTERAÇÃO - REALIZAR CADASTRO



DSS - REALIZAR CADASTRO

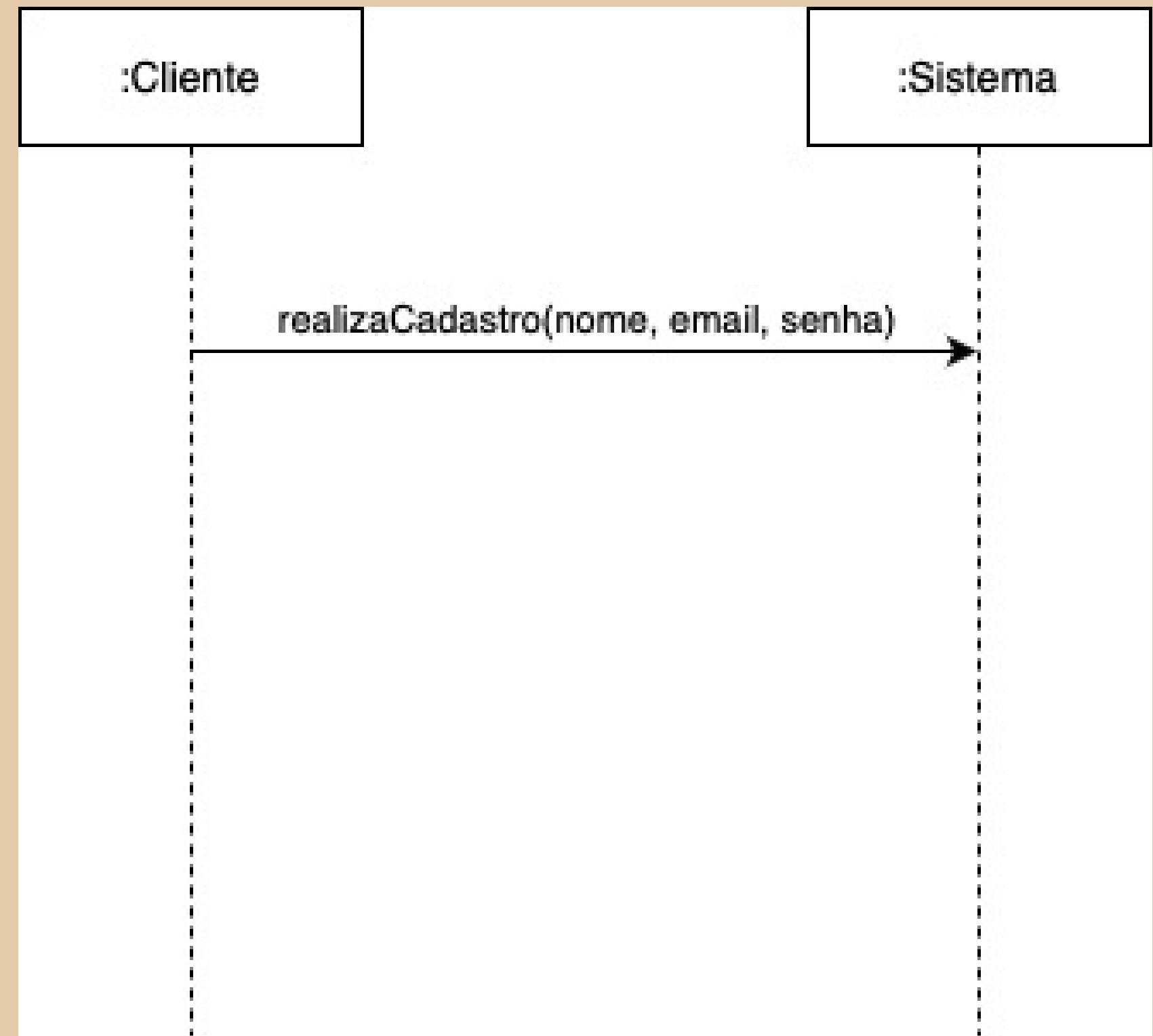
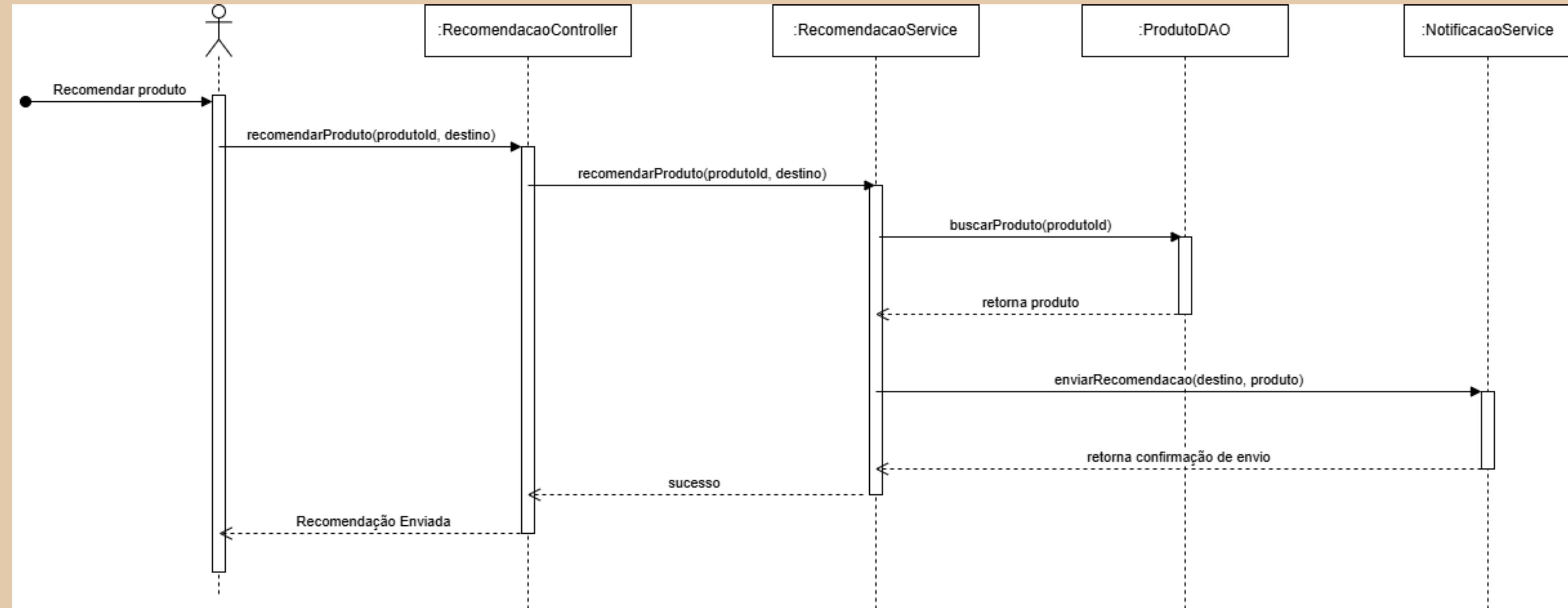


DIAGRAMA DE INTERAÇÃO - RECOMENDAR PRODUTO



DSS - RECOMENDAR PRODUTO

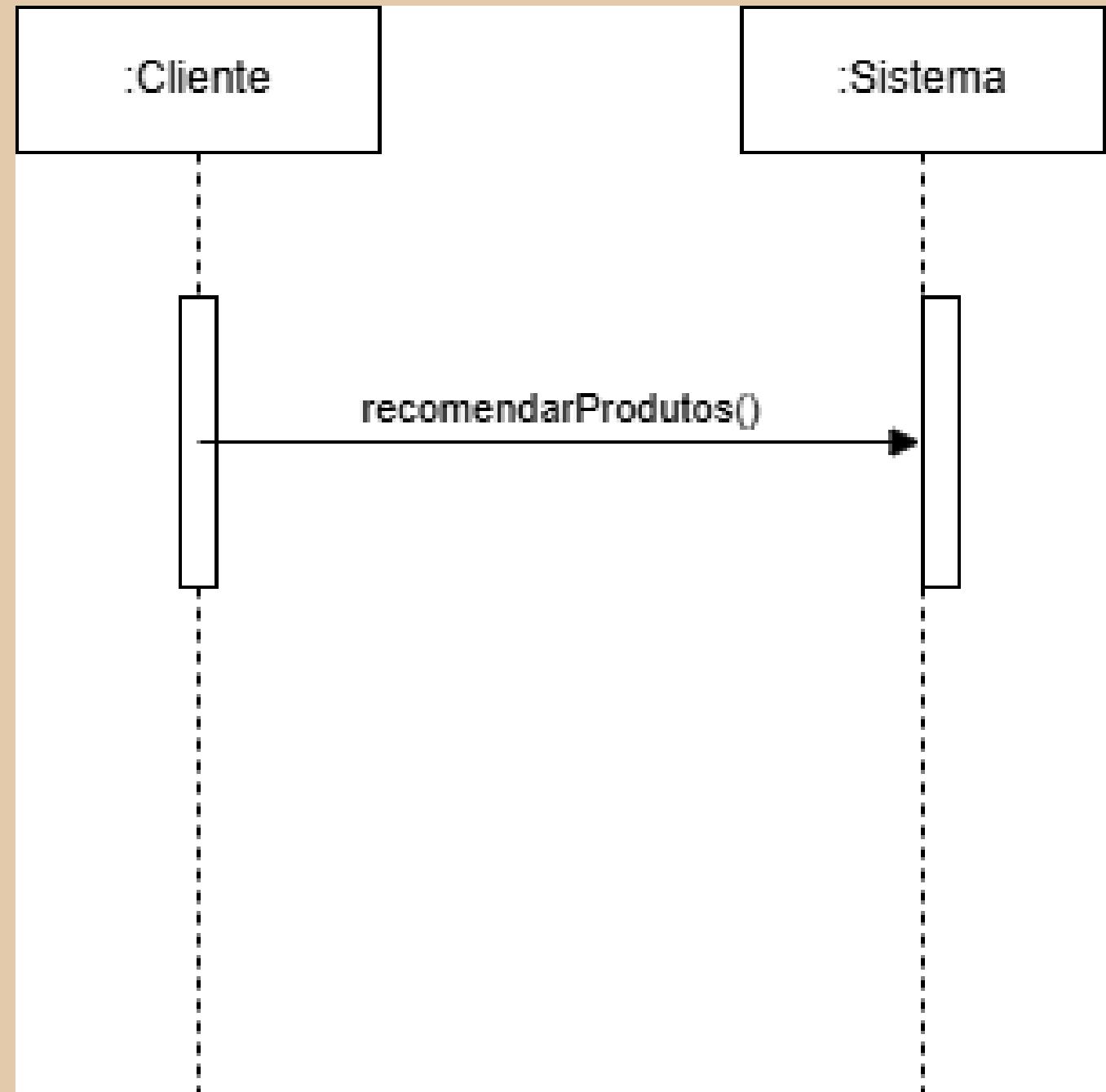
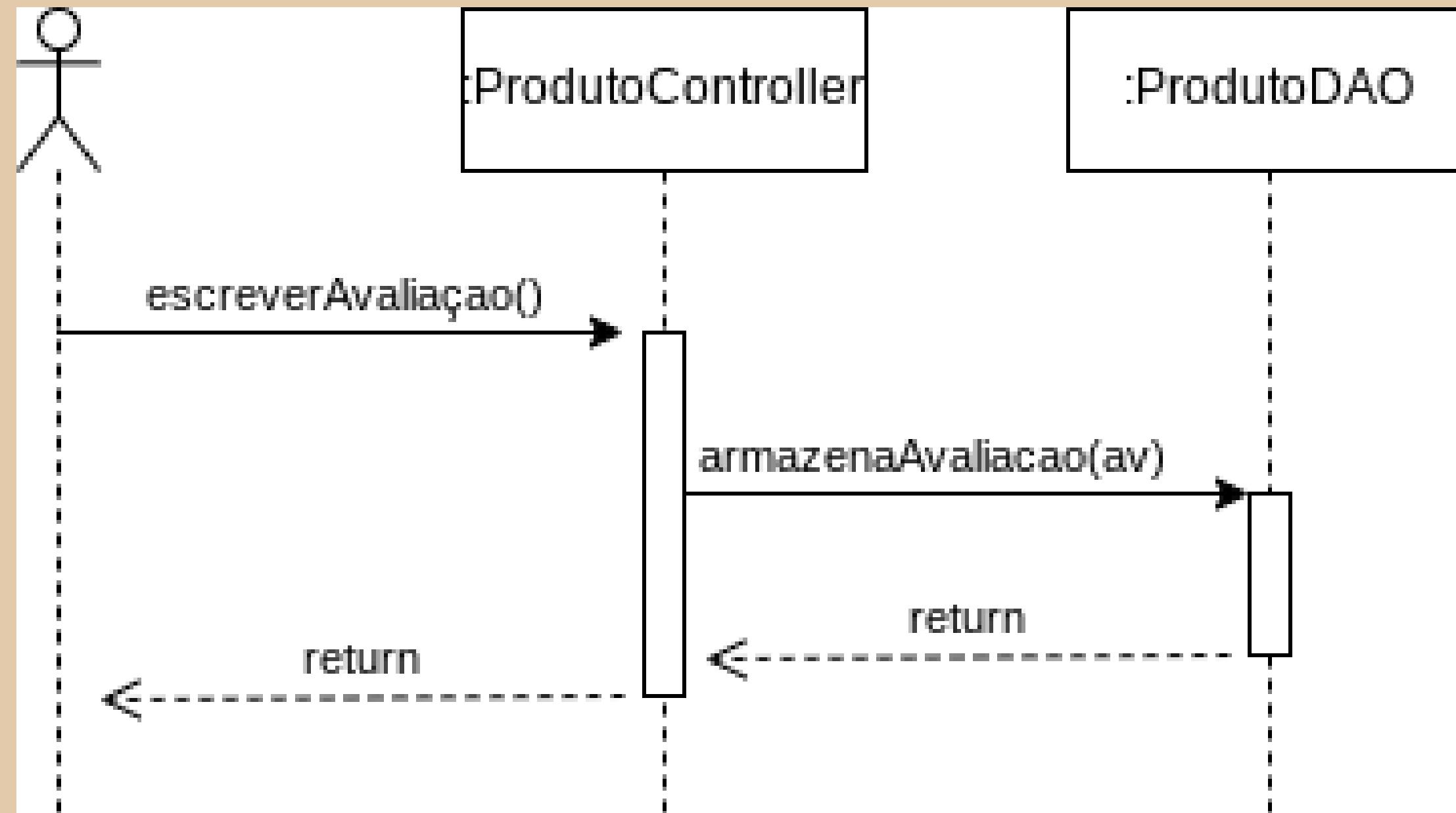


DIAGRAMA DE INTERAÇÃO - AVALIAR PRODUTO



DSS - AVALIAR PRODUTO

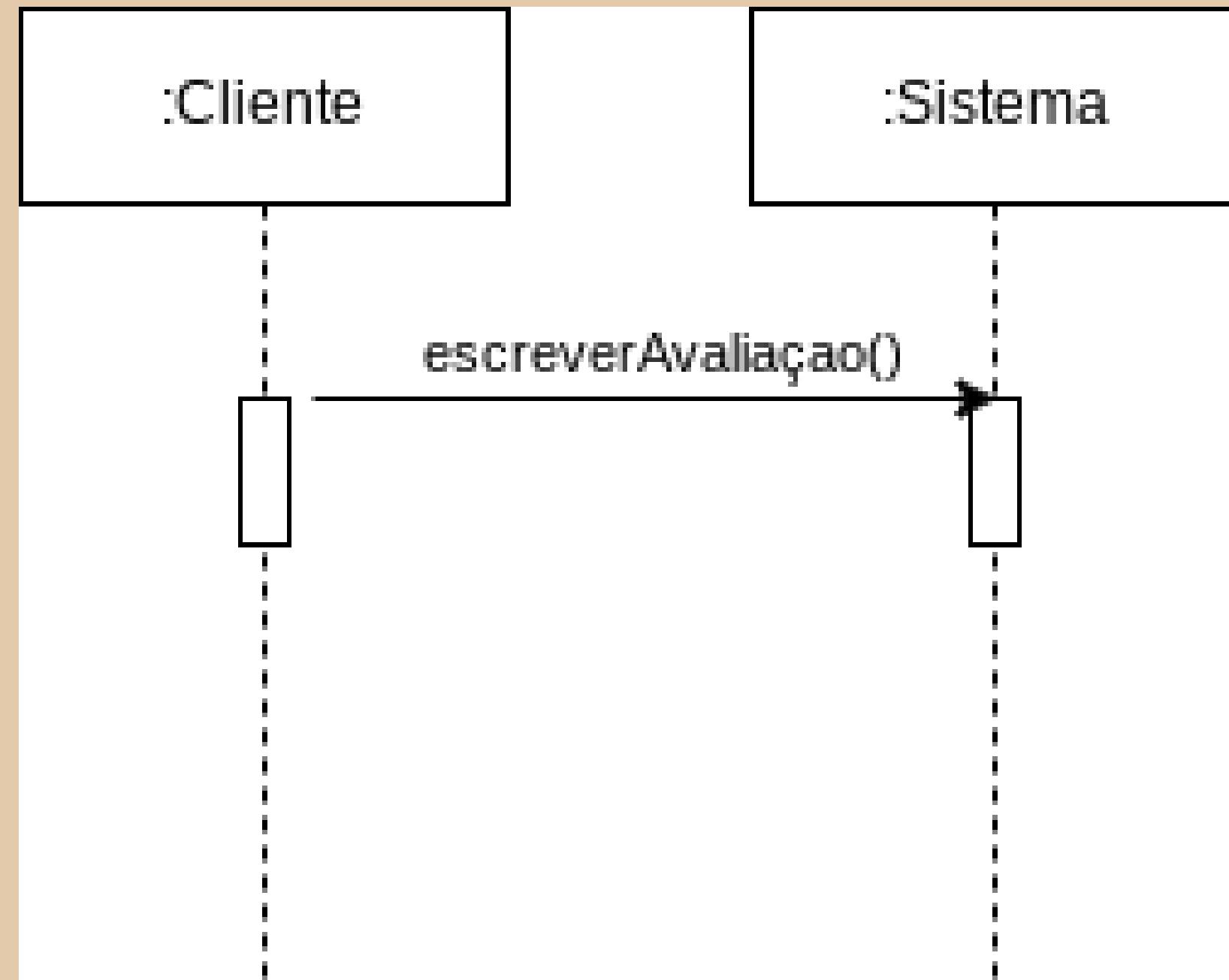
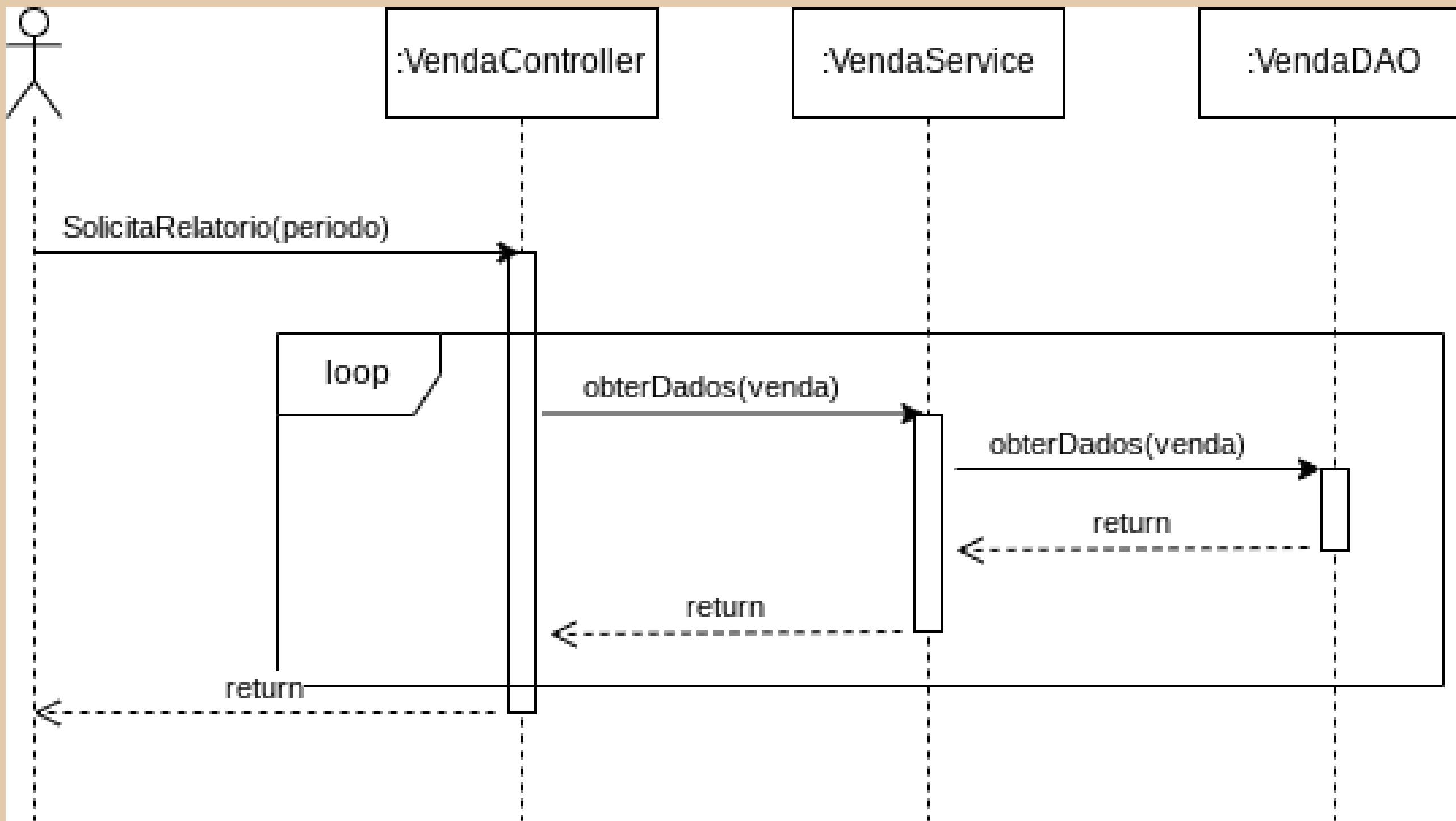


DIAGRAMA DE INTERAÇÃO - GERAR RELATÓRIO



DSS - GERAR RELATÓRIO

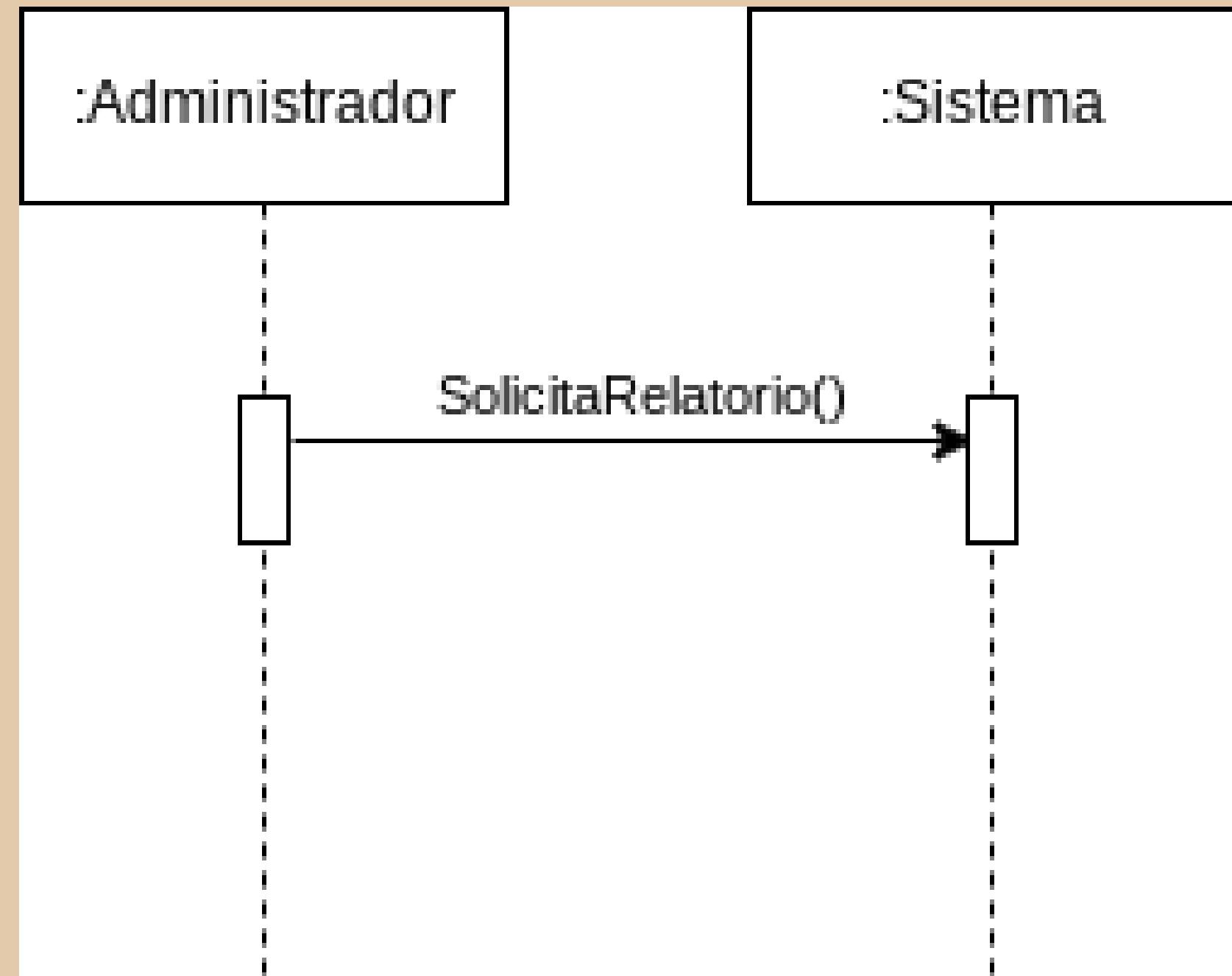
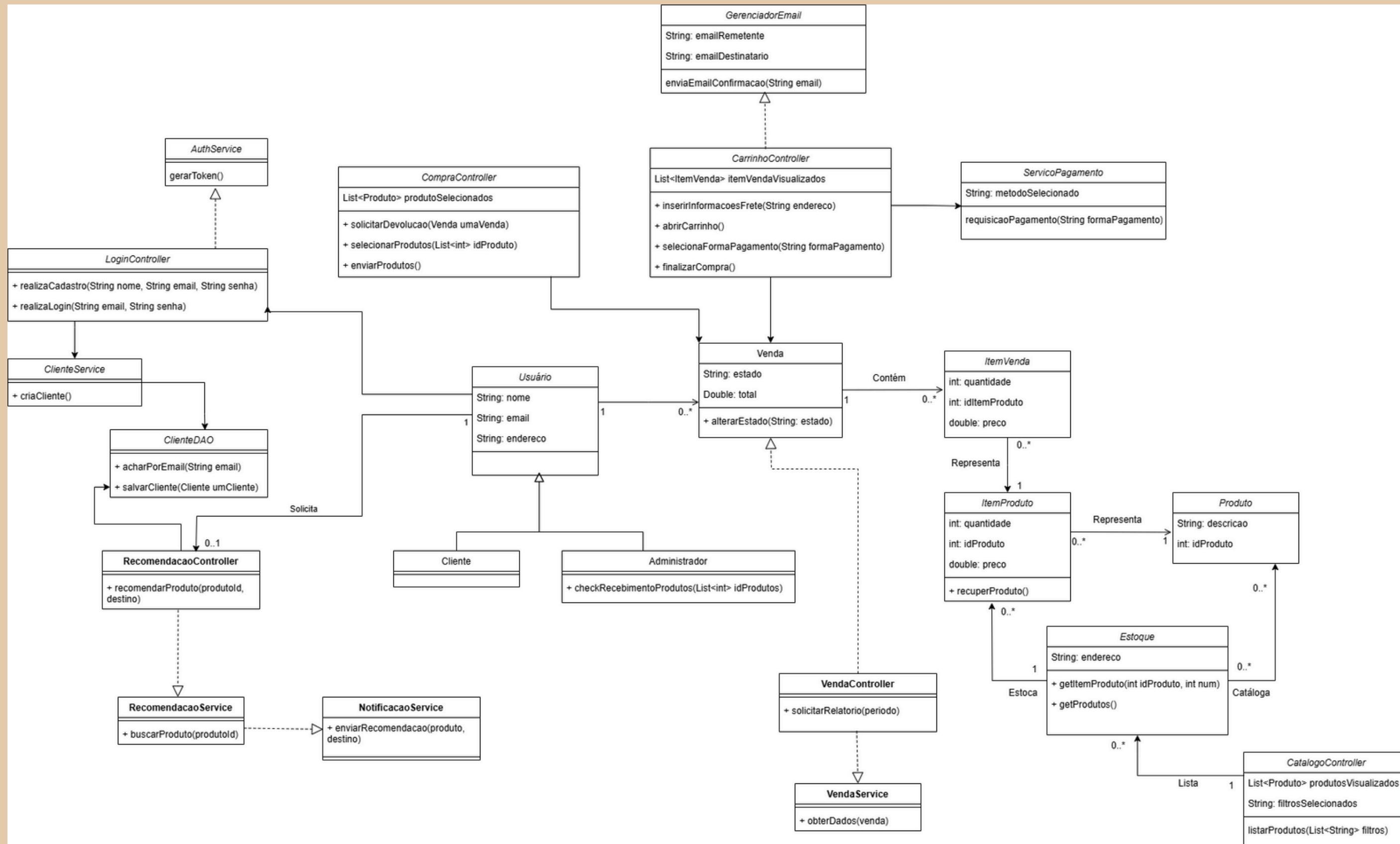


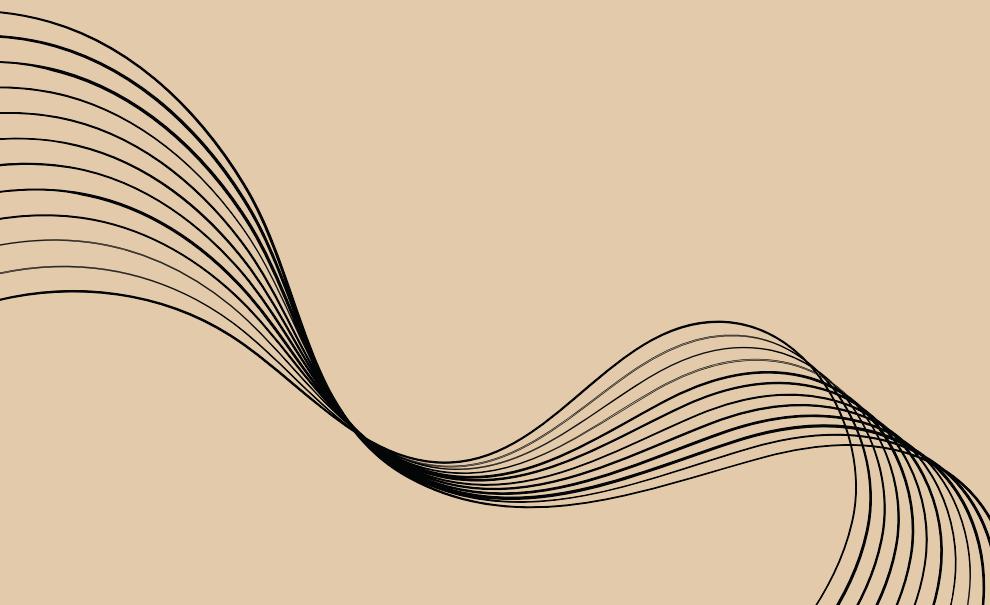
DIAGRAMA DE CLASSE DETALHADO



DIFICULDADES ENCONTRADAS E DECISÕES TOMADAS

Dificuldades encontradas ao logo do trabalho:

- Tivemos dificuldade em decidir o que é complexo o bastante para fazer um diagrama ou não.
- Dificuldade em alguns momentos de balancear funcionalidades essenciais e funcionalidades extras.
- Foi difícil entender como alinhar as arquiteturas escolhidas com as classes que aparecem nos diagramas.
- Não sabíamos a melhor forma exatamente de fazer o diagrama com visão geral da arquitetura do sistema



OBRIGADO
PELA
ATENÇÃO!

