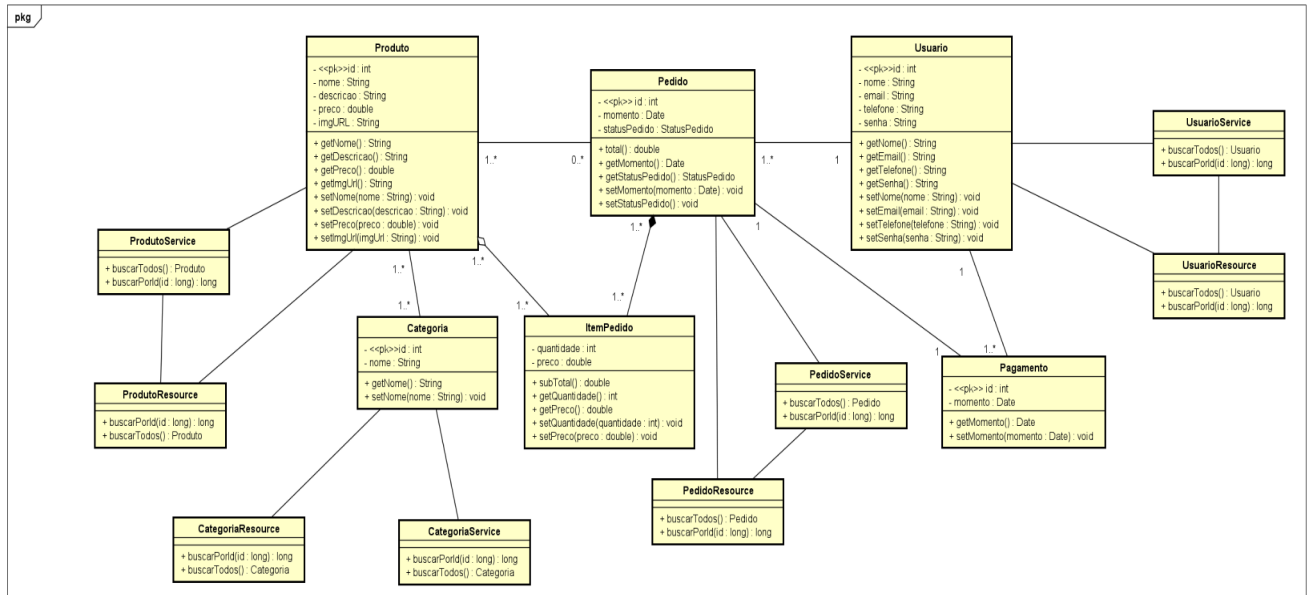


Instituto Federal da Paraíba - IFPB
Campus - Esperança

KLEBSON AMARANTE DA SILVA
MATEUS TOMAZ RIBEIRO
GUILHERME SANTOS

CONSISTÊNCIA DE DIAGRAMAS

Avaliação da modelagem do diagrama de classes



1. Estrutura Geral

- ✓ O diagrama possui classes? Se houver elementos desnecessários ou vazios, devem ser removidos.
- ✓ Todas as classes possuem atributos e/ou métodos? Classes vazias sem justificativas devem ser revisadas.
- ✓ Cada classe possui um nome distinto? Não devem existir classes duplicadas com o mesmo identificador.

2. Validação de classes abstratas e concretas

- ✓ Se houver classes abstratas no diagrama, elas devem conter pelo menos um método abstrato, seja próprio ou herdado. (Neste diagrama, não há classes abstratas.)

✓ Nenhuma classe concreta deve possuir métodos abstratos, seja próprio ou herdado sem sobrescrita

3. Atributos e Métodos

✓ Nenhuma classe deve conter atributos repetidos, seja dentro dela mesma ou herdados sem necessidade.

✓ Métodos com o mesmo nome e mesma lista de parâmetros só devem existir se caracterizarem sobrecarga (múltiplas versões do método) ou sobrescrita (redefinição de um método herdado)..

✓ Todos os atributos, parâmetros e retornos de métodos devem ter um tipo claramente especificado.

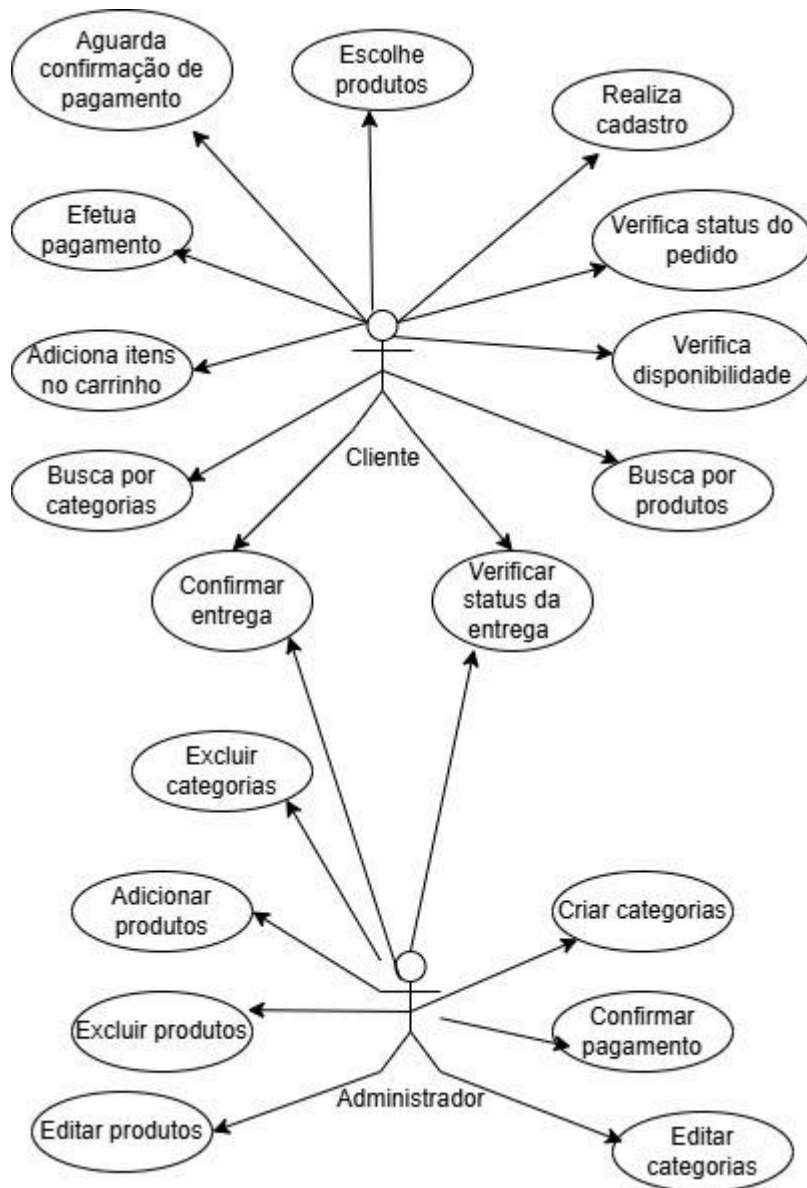
4. Relacionamentos e Herança

✓ A hierarquia de herança deve ser linear, sem ciclos que tornem a estrutura inconsistente.

✓ Se uma classe implementa uma interface, deve fornecer implementação para todos os métodos declarados. (Não se aplica a este diagrama.)

✓ Métodos abstratos herdados devem ser sobrescritos no nível mais baixo da hierarquia, a menos que seja um framework orientado a objetos. (Não se aplica a este diagrama.)

Avaliação da Modelagem do Diagrama de Caso de Uso



1. Correspondência com Requisitos Funcionais

✓ Cada caso de uso deve representar uma funcionalidade esperada do sistema, ainda que não haja uma correspondência exata um para um com os requisitos documentados.

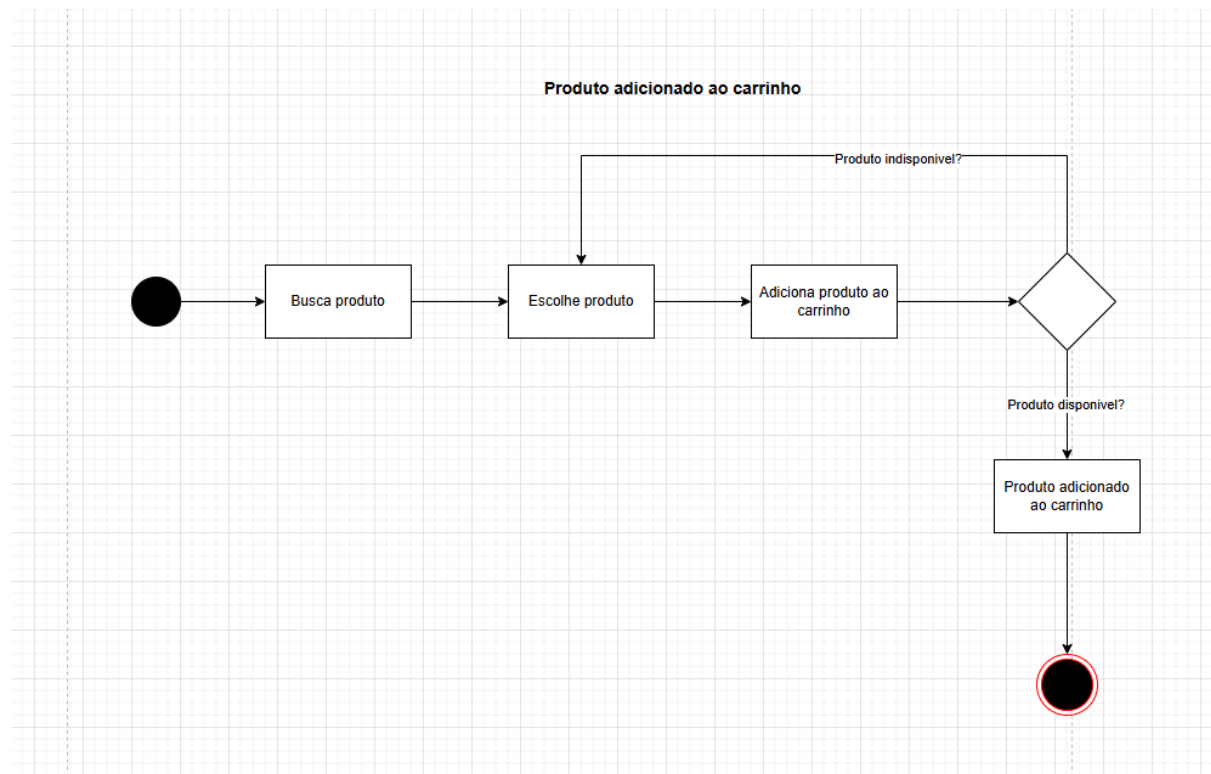
2. Validação de Casos de Uso e Atores

- ✓ Todo caso de uso deve estar ligado a pelo menos um ator, direta ou indiretamente.
- ✓ Os relacionamentos devem estar corretamente aplicados
- ✓ Nenhum ator pode estar isolado; todos devem estar conectados a pelo menos um caso de uso ou especializar um ator que esteja.

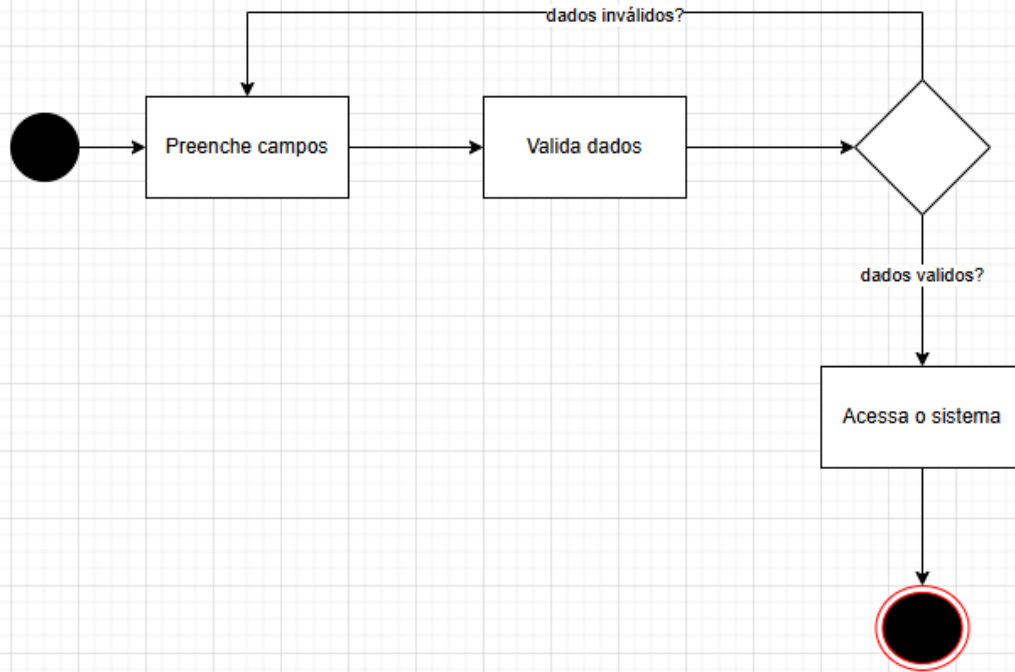
3. Consistência nos Relacionamentos

- ✓ Não devem existir ciclos em relacionamentos evitando loops infinitos na execução dos casos de uso.
- ✓ A generalização deve ser hierárquica e sem ciclos, garantindo uma estrutura lógica e compreensível.

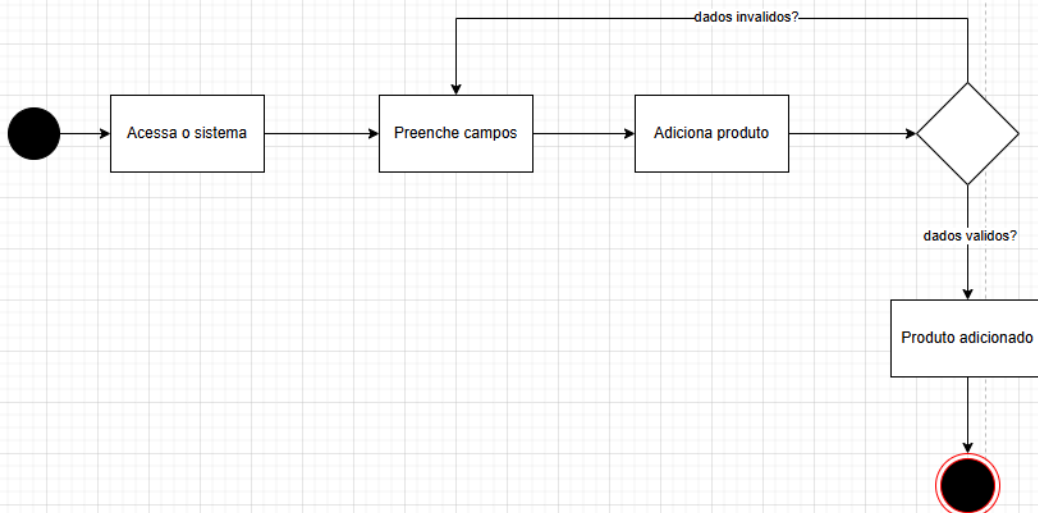
Avaliação da Modelagem em Diagrama de Atividades

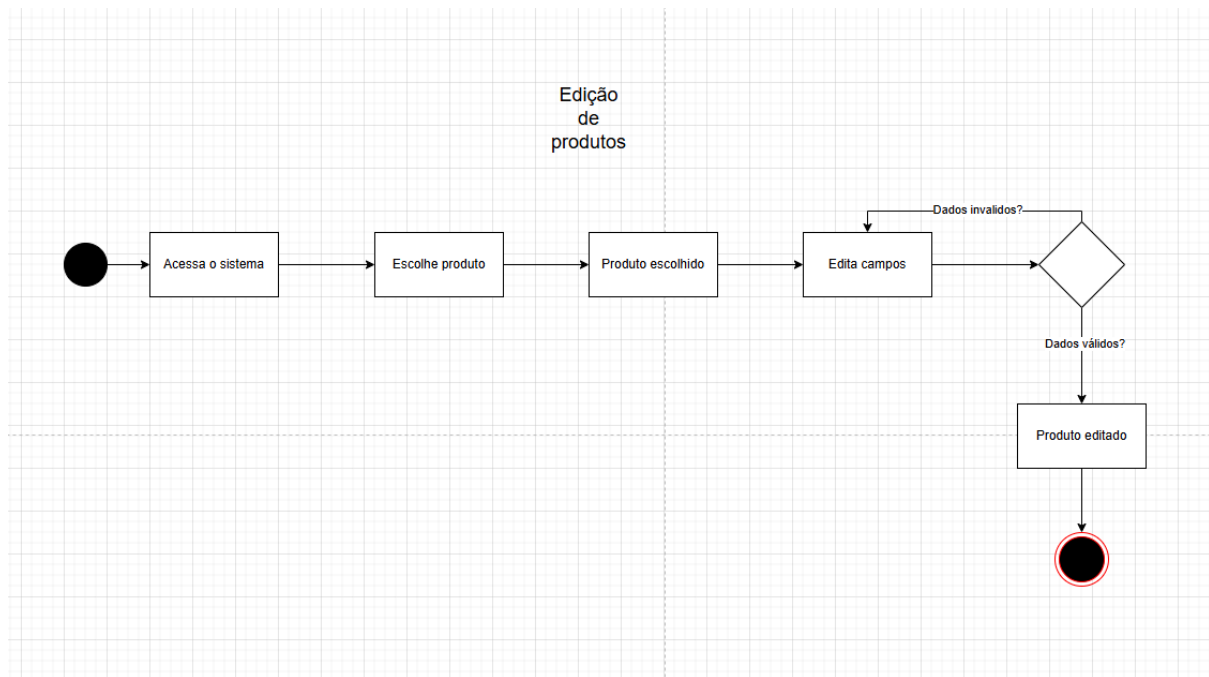


Login



Cadastro de produtos





1. Estrutura Geral

- ✓ O diagrama deve representar corretamente o fluxo das atividades dentro do processo modelado.
- ✓ Cada fluxo deve possuir um nó de início e um nó de término claramente definidos.
- ✓ Todas as atividades devem ter ao menos uma transição de entrada e uma de saída, exceto os nós inicial e final.

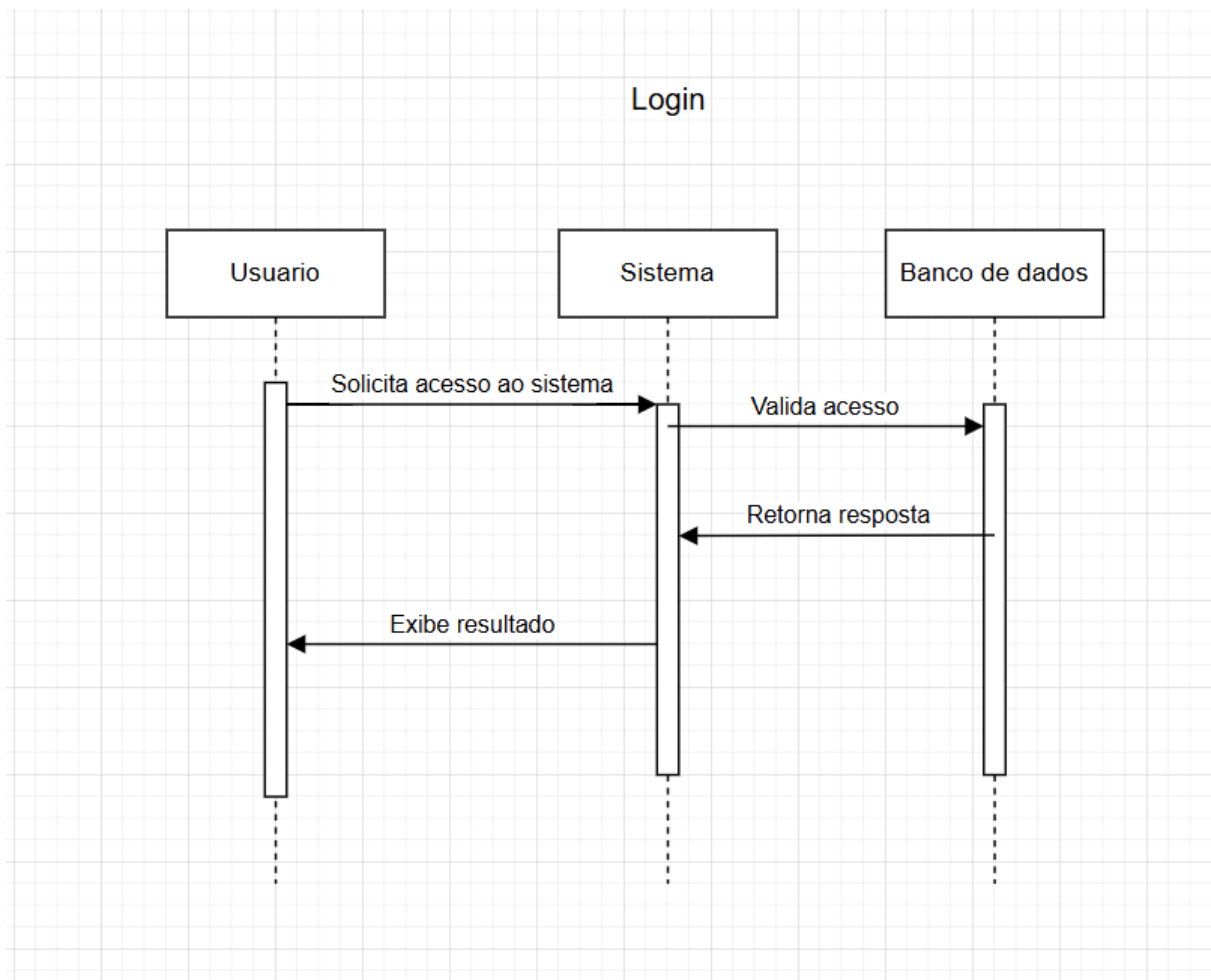
2. Decisões e Conectores

- ✓ Todas as decisões devem ter condições bem definidas e mutuamente excludentes para evitar ambiguidades no fluxo.
- ✓ Não devem existir ciclos infinitos sem uma condição de saída clara.
- ✓ As transições entre atividades devem estar corretamente definidas e seguir uma lógica coerente dentro do processo.

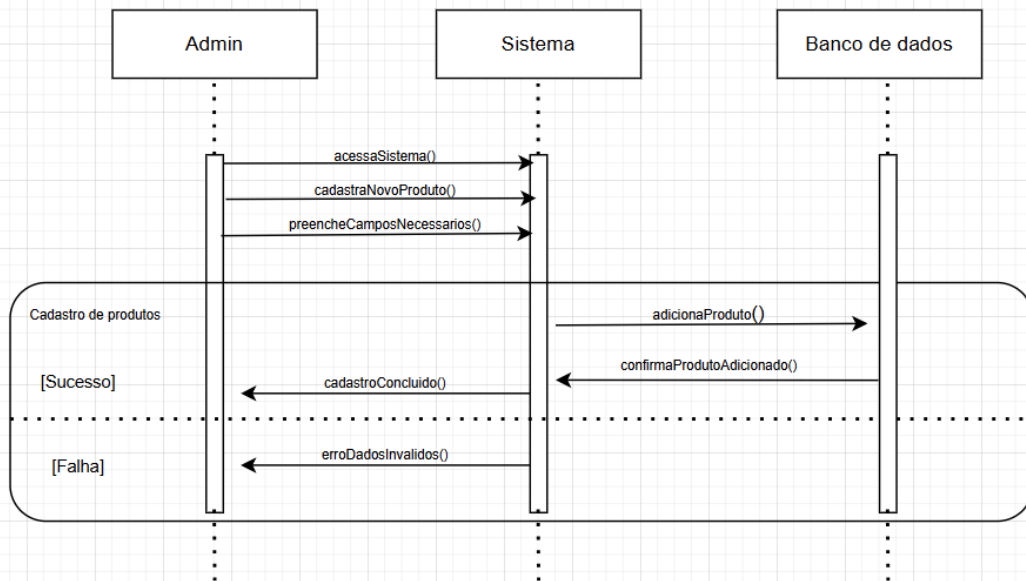
3. Objetos e Fluxo de Dados

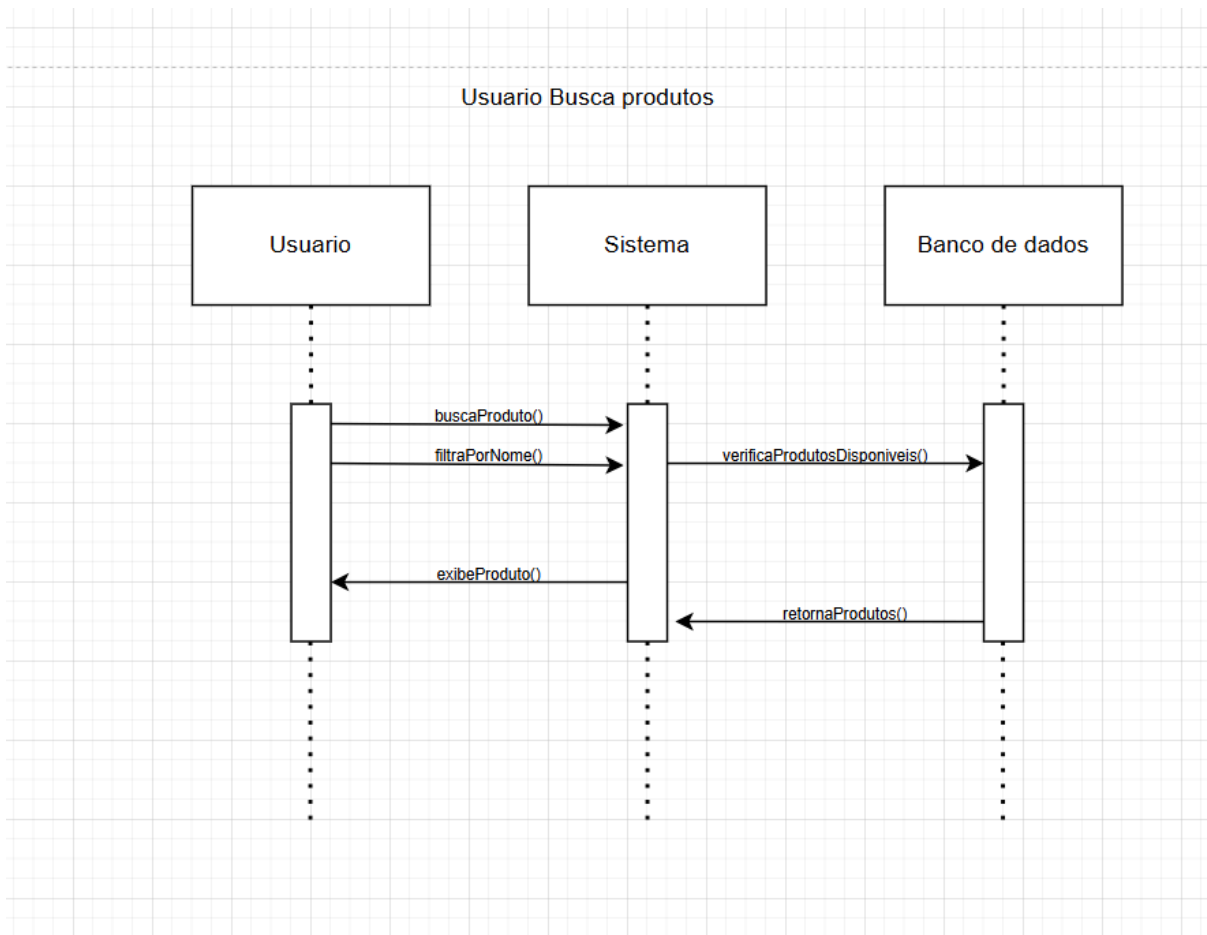
- ✓ Se o diagrama incluir objetos no fluxo, eles devem estar corretamente representados e conectados às atividades correspondentes.
- ✓ O fluxo de dados entre atividades deve ser consistente, garantindo que as informações fluam corretamente dentro do processo.

Avaliação da Modelagem do Diagrama de Sequência



Cadastro de produtos





1. Estrutura e Elementos

- ✓ Todos os objetos envolvidos na interação estão corretamente representados.
- ✓ As mensagens entre os objetos estão corretamente identificadas e seguem a ordem de execução.
- ✓ Os nomes das mensagens são coerentes com as ações realizadas no sistema.

2. Validação de Fluxo

- ✓ O fluxo de interação entre os objetos está coerente e representa corretamente o comportamento esperado do sistema.
- ✓ Condições e loops estão corretamente representados, garantindo um fluxo lógico nos processos.

✓ A diferenciação entre interações assíncronas e síncronas está aplicada de forma adequada.