

# **Universidade de São Paulo - USP**

Instituto de Ciências Matemáticas e de Computação

SSC0124 - Análise e Projeto Orientados a Objetos

---

## **Projeto 3: Diagramas de Comunicação Sistema de Biblioteca**

---

9167910 Carlos Alberto Schneider Junior  
8936926 Frederico de Azevedo Marques  
8937420 Lucas Kassouf Crocomo  
8936756 Roberto Pommella Alegro  
8066395 Rodrigo das Neves Bernardi

# Sumário

Introdução	2
Diagramas de Comunicação	3
inserirLivro	3
alterarDadosFuncionário	4
buscarUsuário	5
excluirFornecedor	6
Observações	7
Conclusão	8

# Introdução

Durante a primeira fase de um projeto, a de análise, alguns artefatos são estudados visando compreender quais serão os processos, os eventos e as operações que são pertinentes ao sistema e quais serão modeladas. Uma vez concluída a análise, é feita a transição para a fase de projeto, onde uma solução lógica e competente entre em cena.

Um dos principais artefatos estudados durante esta segunda fase do projeto é o diagrama de Interação, que busca descrever em formato de grafo as interações entre classes e objetos, do ponto de vista das funcionalidades do sistema.

O diagrama em questão pode ser representado na UML através do chamado Diagrama de Comunicação, cujo objetivo é modelar quais mensagens serão transmitidas, dada uma certa funcionalidade, ao longo do sistema e observar a relação entre as classes e objetos, como o nível de coesão e acoplamento. Esse diagrama, em outras palavras, também é conhecido por atribuir responsabilidades aos elementos do sistema.

Para a elaboração do diagrama é necessário partir de um Contrato de Operações, tendo em vista as descrições da funcionalidade que será modelada, suas pré e pós condições, e os parâmetros utilizados conforme descritos no contrato. A noção e a manipulação desses aspectos são necessários para modelarmos quais mensagens serão enviadas à cada elemento (classe, objeto, classe instanciada).

A fim de tornar a modelagem mais concisa, serão aplicados os conceitos de Orientação a Objeto e também alguns padrões para atribuição de responsabilidade (GRASP), ensinados em sala de aula, tais como Especialista, Criador, Coesão Alta, Acoplamento fraco e Controlador.

Foi utilizada a ferramenta Astah Professional<sup>1</sup> para criar os diagramas de comunicação.

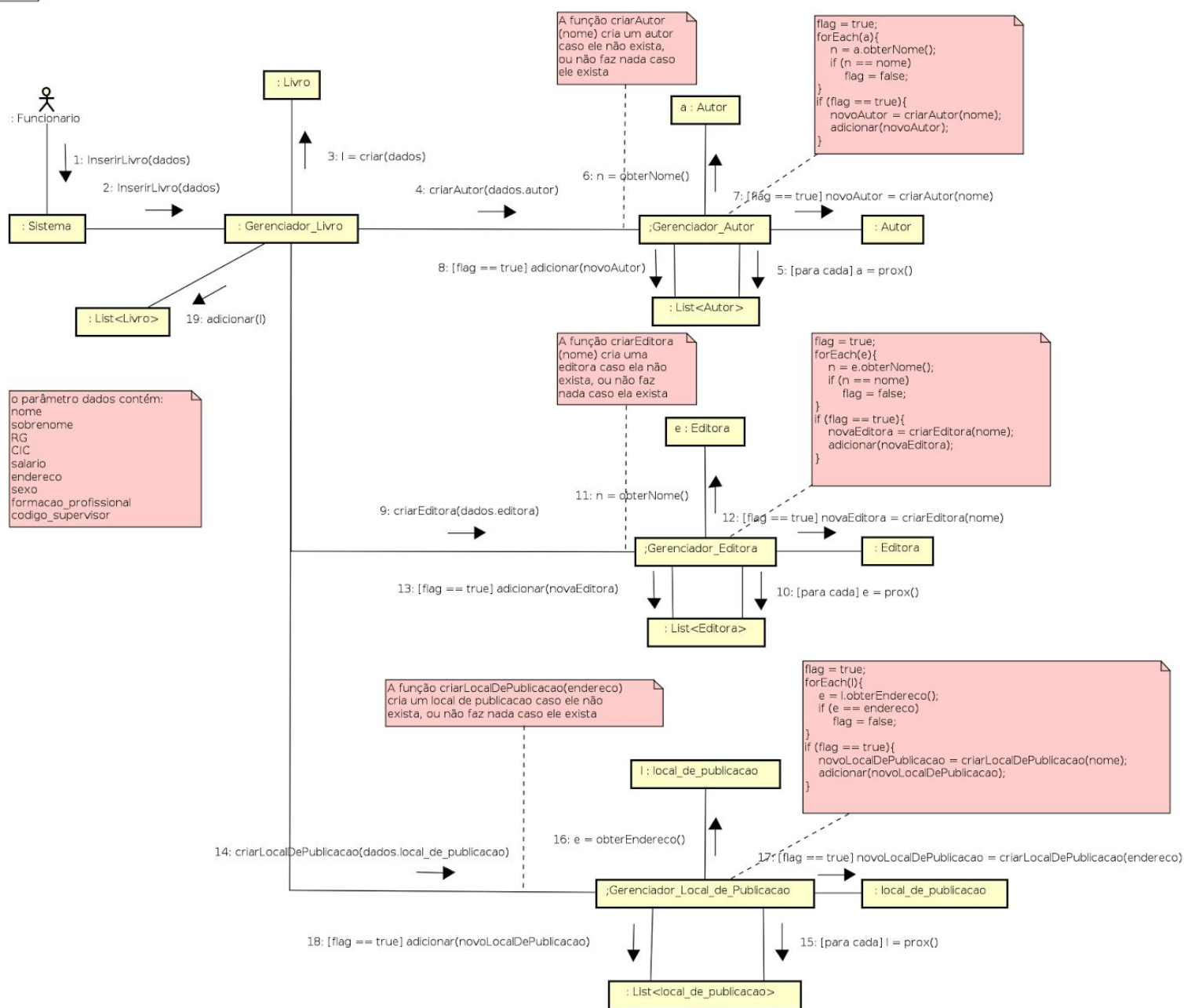
---

<sup>1</sup> <http://astah.net/>

# Diagramas de Comunicação

## inserirLivro

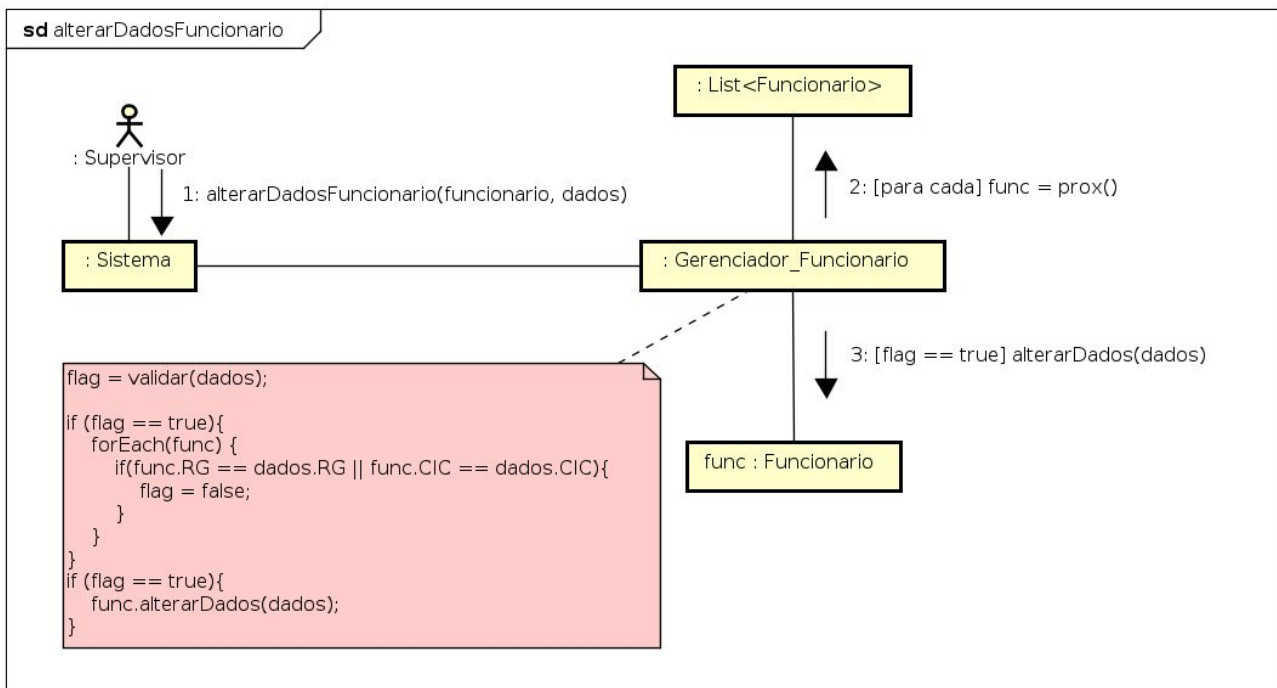
sd InserirLivro



## Padrões GRASP:

- Criador: A classe que gerencia os livros cria objetos do tipo Livro.
- Especialista: A classe de livros é aquela que conhece as informações para a criação de novos livros, por tanto é sua responsabilidade criar novos livros.
- Controlador: Representa um subsistema (livros da biblioteca)
- Acoplamento baixo: Separação da coleção de livros em: Gerenciador, Livro (indivíduo) e coleção em si.
- Coesão alta: A característica semântica do sistema é garantida, com um baixo nível de complexidade.

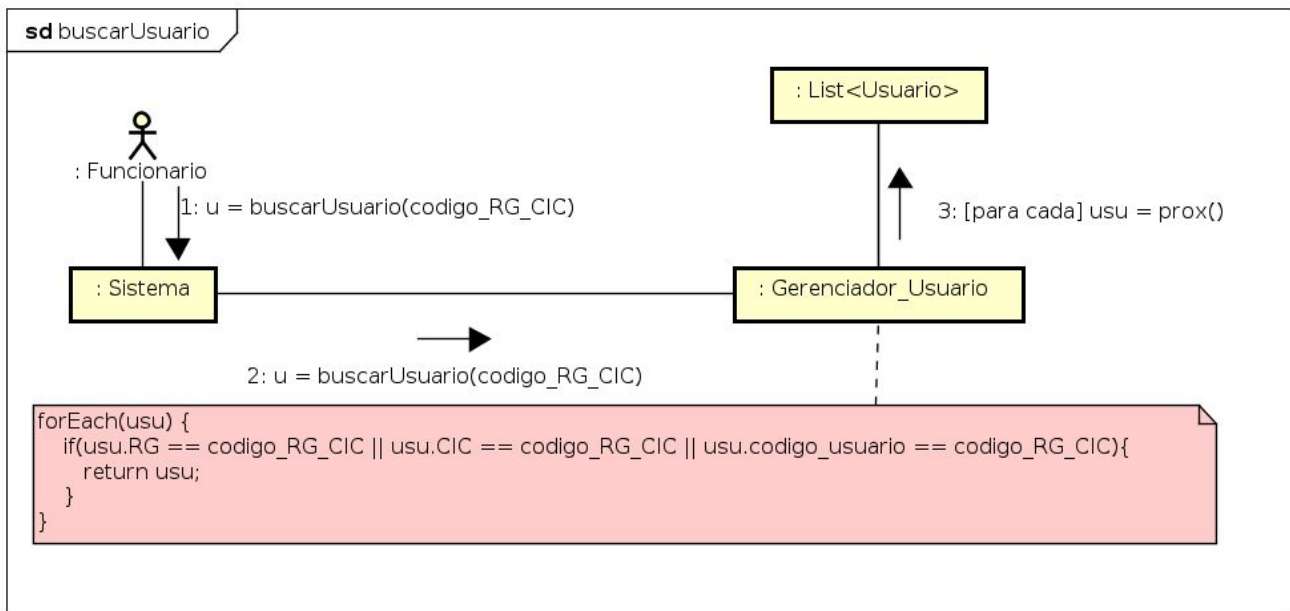
## alterarDadosFuncionario



## Padrões GRASP:

- Especialista: Dado um funcionário, ele mesmo é o responsável por suas informações, logo a alteração de seus dados deve ser providenciada por um método do objeto.
- Controlador: Representa um subsistema (funcionários da biblioteca)
- Coesão alta: A característica semântica do sistema é garantida, com um baixo nível de complexidade.

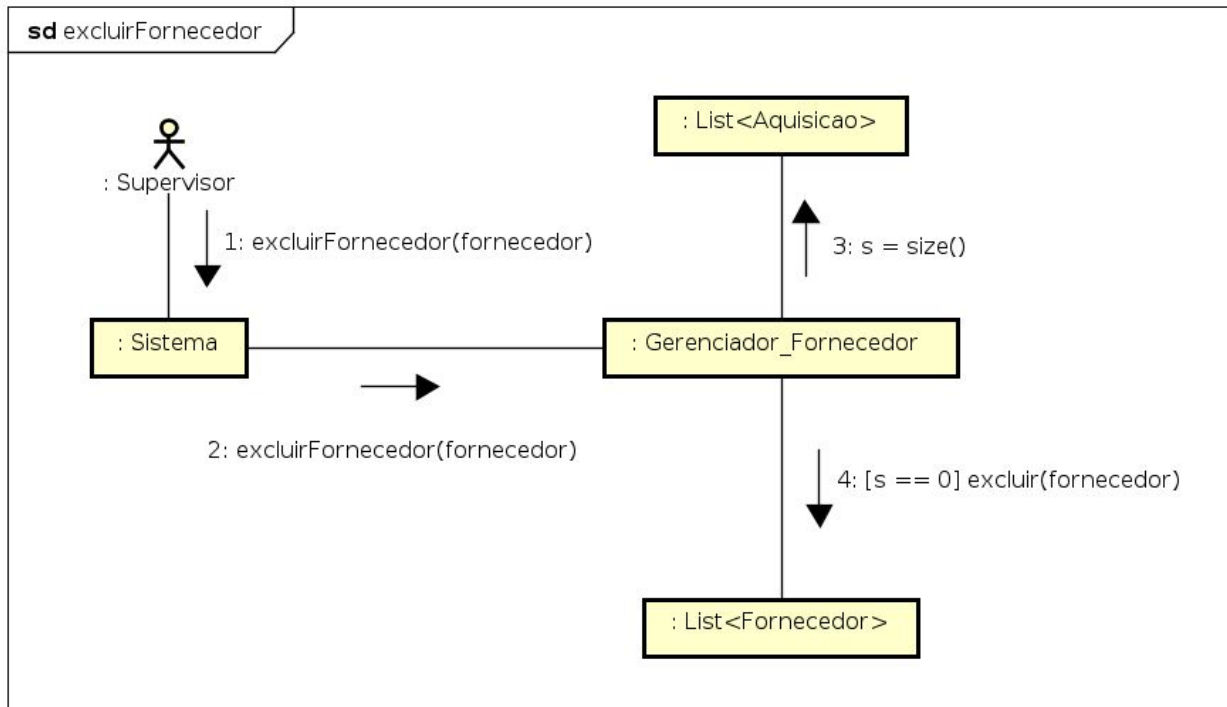
## buscarUsuário



### Padrões GRASP:

- Especialista: A coleção de usuários é responsável pela consulta de usuários
- Controlador: Representa um subsistema (funcionários da biblioteca)
- Coesão alta: A característica semântica do sistema é garantida, com um baixo nível de complexidade.

## excluirFornecedor



### Padrões GRASP:

- **Especialista:** As coleções Aquisição e Fornecedor são responsáveis respectivamente por saber as informações das aquisições (para busca de fornecedores atrelados a uma delas) e por excluir um fornecedor da coleção de existentes
- **Controlador:** Representa um subsistema (fornecedores da biblioteca)
- **Acoplamento baixo:** Separação da dos fornecedores e aquisições em duas classes distintas com uma coleção cada.
- **Coesão alta:** A característica semântica do sistema é garantida, com um baixo nível de complexidade.

## Observações

1. Como o PlantUML não tem suporte para o Diagrama de Comunicação, tivemos a necessidade de mudar a ferramenta. Escolhemos assim, o Astah Professional que suporta geração de código fonte a partir deste diagrama, indispensável para o trabalho seguinte.
2. O grupo encontrou dificuldades na execução deste trabalho, uma vez que o anterior se encontrava incompleto e superficial. Foram necessárias mudanças nos Contratos de Operação e alguns detalhes no Modelo Conceitual para que a execução deste trabalho fosse realizada corretamente.



# Conclusão

No contexto da modelagem de software, os diagramas representam um passo fundamental para a concepção do programa final, pois permitem a análise das interações entre classes de cada evento do sistema.

Através dos diagramas criados, foi possível modelar o comportamento de objetos e classes, a partir de uma determinada funcionalidade. Para a modelagem dos diagramas, nota-se que para se determinar as mensagens que serão transmitidas, foram necessários um certo nível de criatividade e o conhecimento de Orientação a Objetos. Esse nos permitiu manter a integridade semântica e a execução correta do sistema, mantendo-o simples e funcional.

Dessa forma, foi possível compreender a importância desse diagrama para a modelagem, através da análise das responsabilidades e das interações entre as classes, tendo em vista a coesão e o acoplamento entre elas. O programa deve ser reaproveitável (modularizado, com baixo acoplamento) e, ao mesmo tempo, atender as suas especificações (confiabilidade ou coesão).