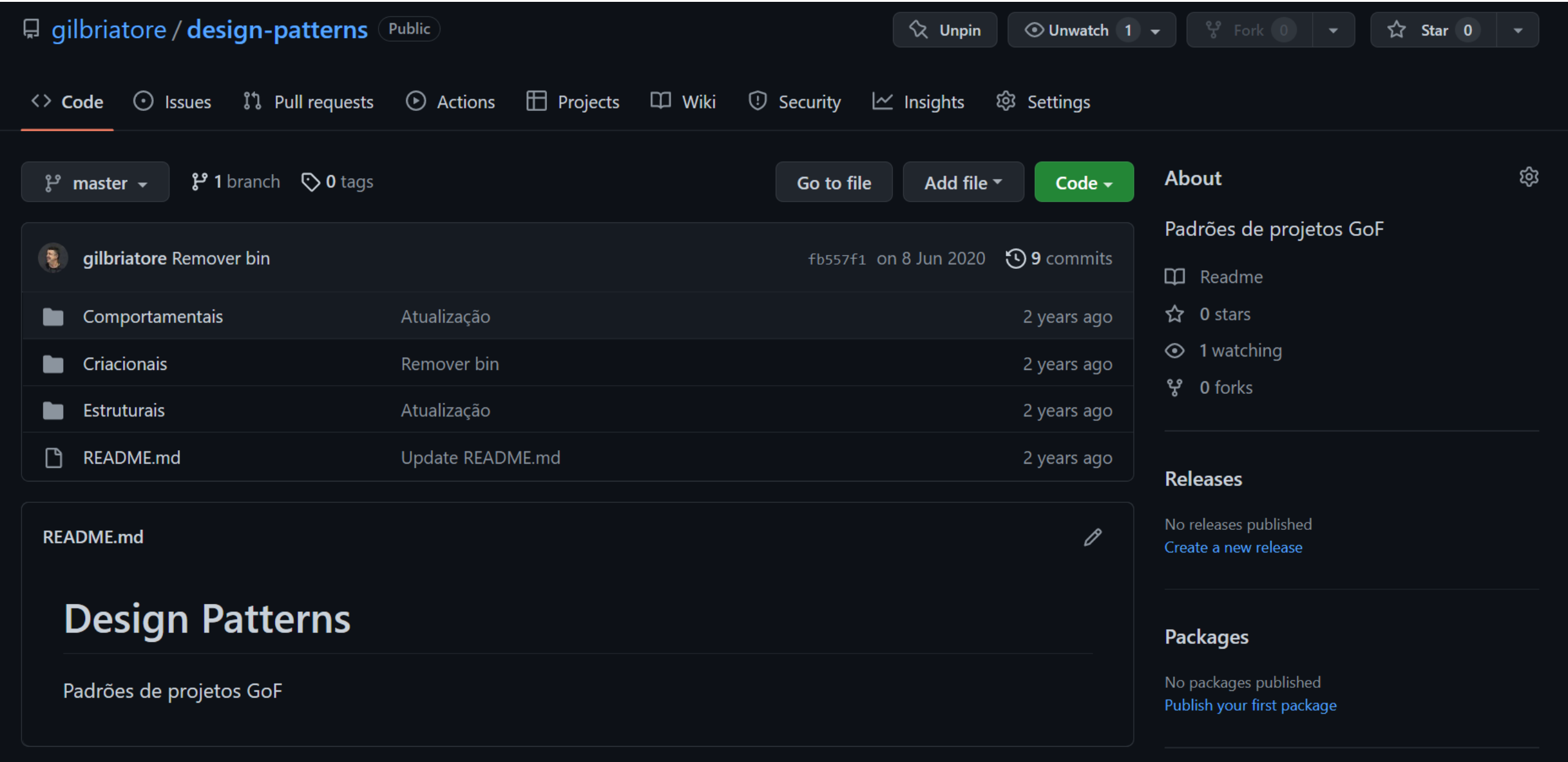
ARQUITETURA DE SOFTWARE

Padrões de projetos criacionais

Geucimar Briatore
geucimar@up.edu.br

Atualizado em 10/2022

https://github.com/gilbriatore/design-patterns



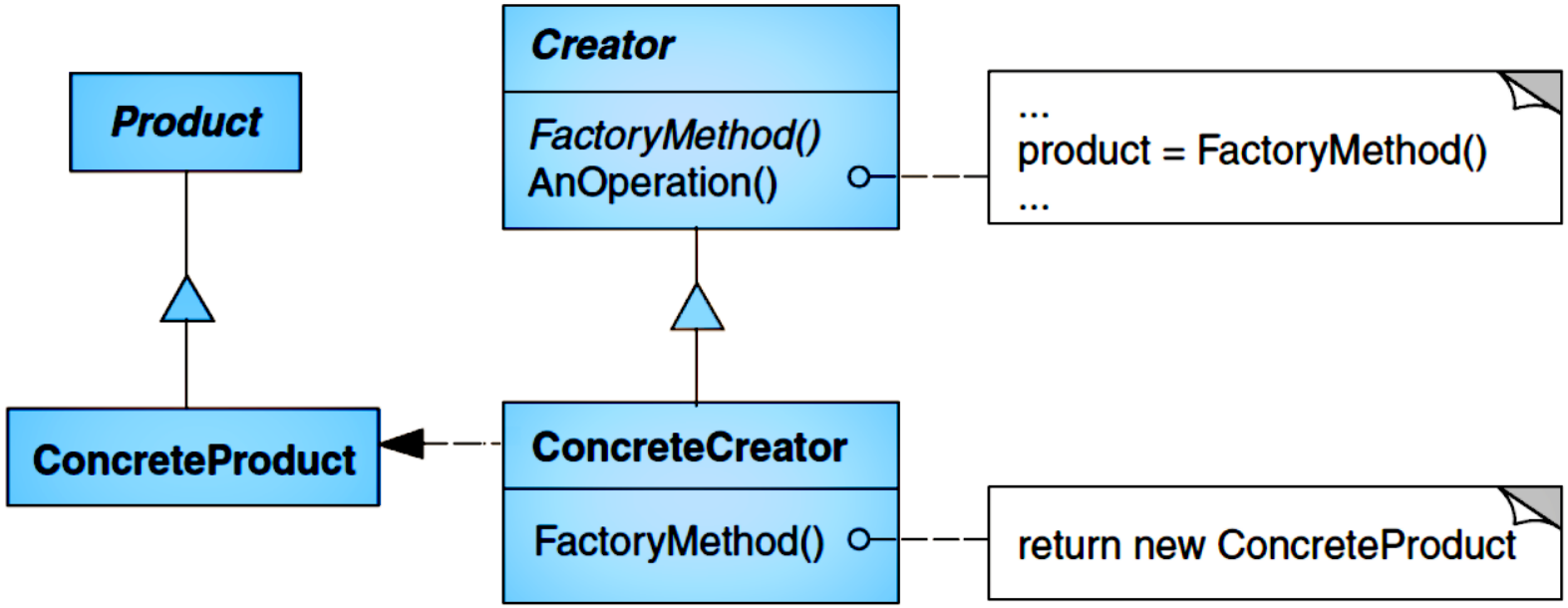
Padrões de projeto do grupo criacional

		Propósito		
		1. Criacional	***2. Estrutural***	***3. Comportamental***
Escopo	**Classe**	*Factory Method*	*Class Adapter*	*Interpreter* *Template Method*
	Objeto	*Abstract Factory* *Builder* *Prototype* *Singleton*	*Object Adapter* *Bridge* *Composite* *Decorator* *Facade* *Flyweight* *Proxy*	*Chain of Responsibility* *Command* *Iterator* *Mediator* *Memento* *Observer* *State* *Strategy* *Visitor*

Factory Method

Tipo: Criacional

Propósito: definir uma interface para criar um objeto, mas deixar as subclasses decidirem que classe instanciar. Permite delegar a instanciação para subclasses;

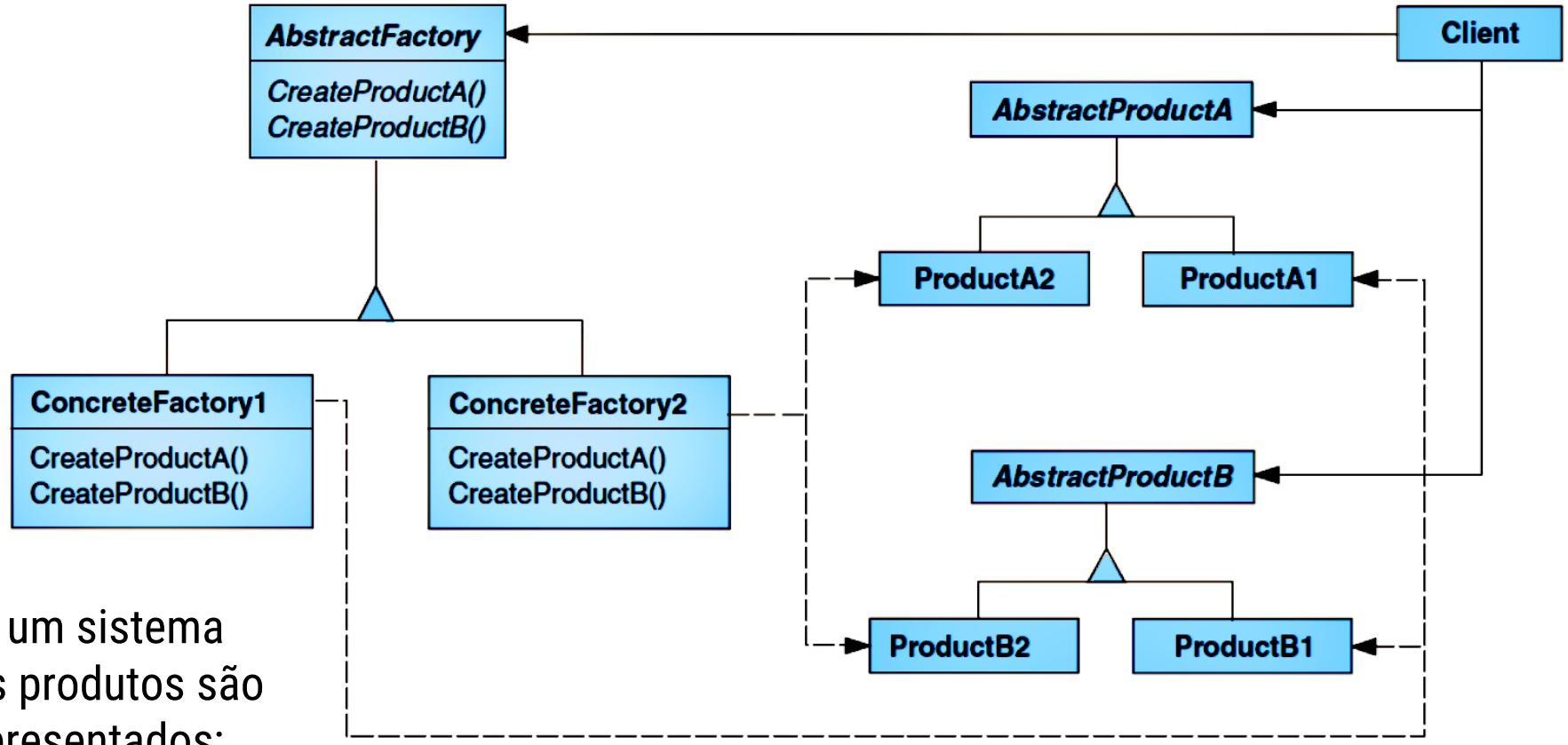


Aplicação: utilizar quando a classe de criação não sabe qual é a classe de objeto deve ser instanciada; a classe de criação quer que suas subclasses determinem os objetos a serem criados; a classe delega responsabilidade para uma dentre várias subclasses auxiliares, e é necessário saber qual subclasse auxiliar que é a delegada.

Abstract Factory

Tipo: Criacional

Propósito: fornecer uma interface para criação de famílias de objetos relacionados ou dependentes sem expor suas classes concretas;

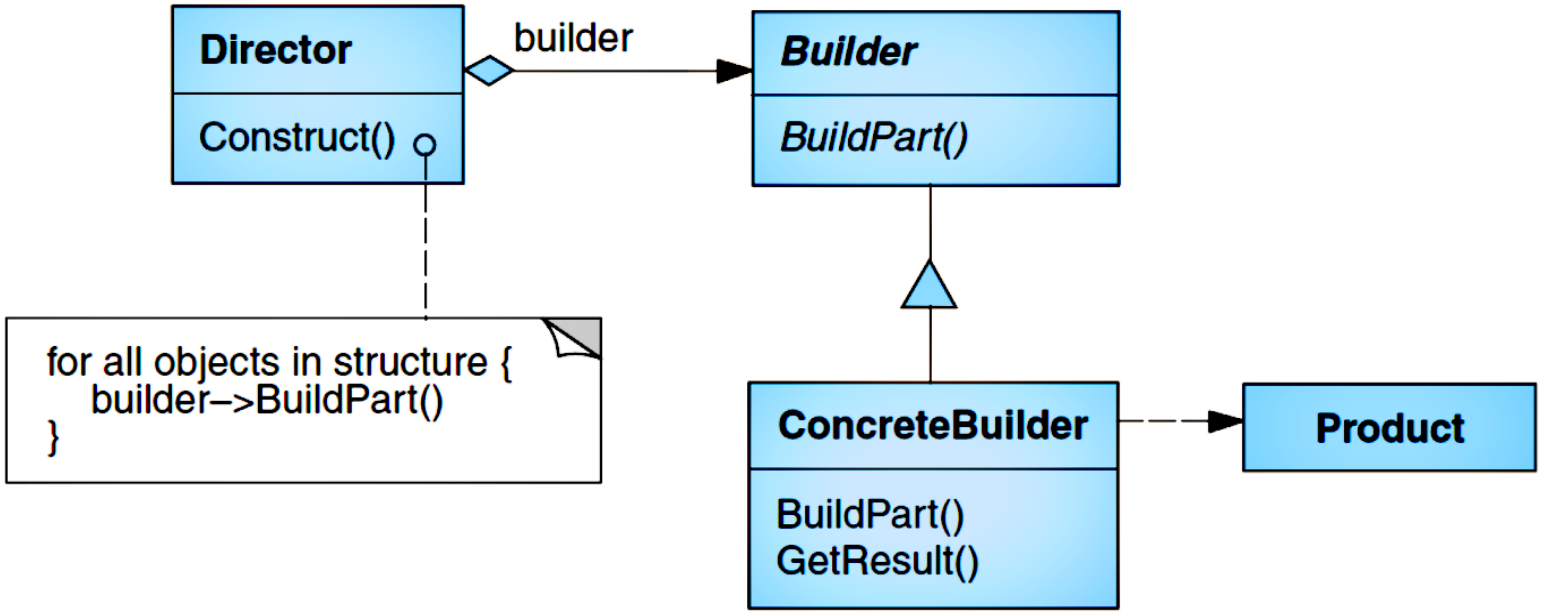


Aplicação: utilizar quando um sistema não deve expor como seus produtos são criados, compostos ou representados; um sistema deve fornecer apenas uma biblioteca de classes de produtos e revelar somente suas interfaces, não suas implementações.

Builder

Tipo: Criacional

Propósito: separar a construção de um objeto complexo da sua representação de modo que o mesmo processo de construção possa criar diferentes representações;

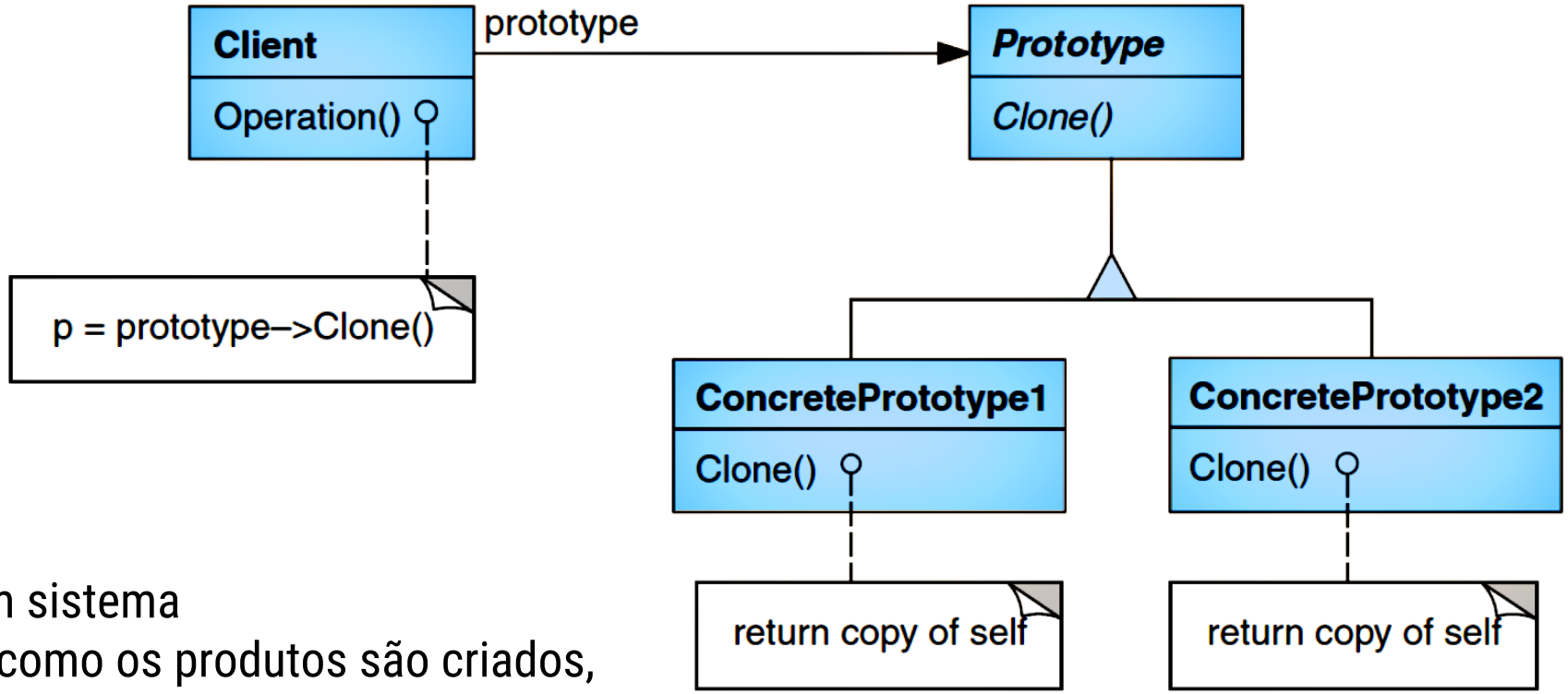


Aplicação: utilizar quando o algoritmo para criação de um objeto complexo deve ser independente das partes que compõem o objeto e de como elas são montadas.
O processo de construção deve permitir diferentes representações para o objeto que é construído.

Prototype

Tipo: Criacional

Propósito: especificar os tipos de objetos a serem criados usando uma instância-protótipo e criar novos objetos pela cópia desse protótipo;



Aplicação: utilizar quando um sistema precisa ser independente de como os produtos são criados, compostos e representados; e quando as classes a instanciar forem definidas em tempo de execução, por exemplo, por criação dinâmica; ou para evitar a construção de uma hierarquia de classes de fábricas paralela à hierarquia de classes de produto; ou quando as instâncias de uma classe puderem ter uma dentre poucas combinações diferentes de estados.

Singleton

Tipo: Criacional

Propósito: garantir que uma classe tenha somente uma instância e fornecer um ponto global de acesso;

Aplicação: utilizar quando for preciso haver apenas uma instância de uma classe, e essa instância tiver que dar acesso aos clientes através de um ponto bem conhecido; a única instância tiver de ser extensível através de subclasses, possibilitando aos clientes usar uma instância estendida sem alterar o seu código.