

Índice

1. Metamodelos	2
----------------	---

1. Metamodelos

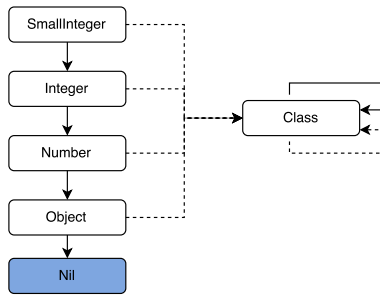


Figura 1: Metamodelo Básico

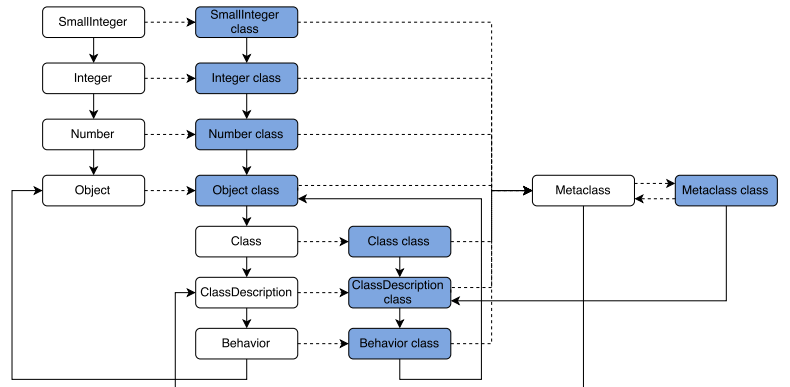


Figura 2: Metamodelo SmallTalk80

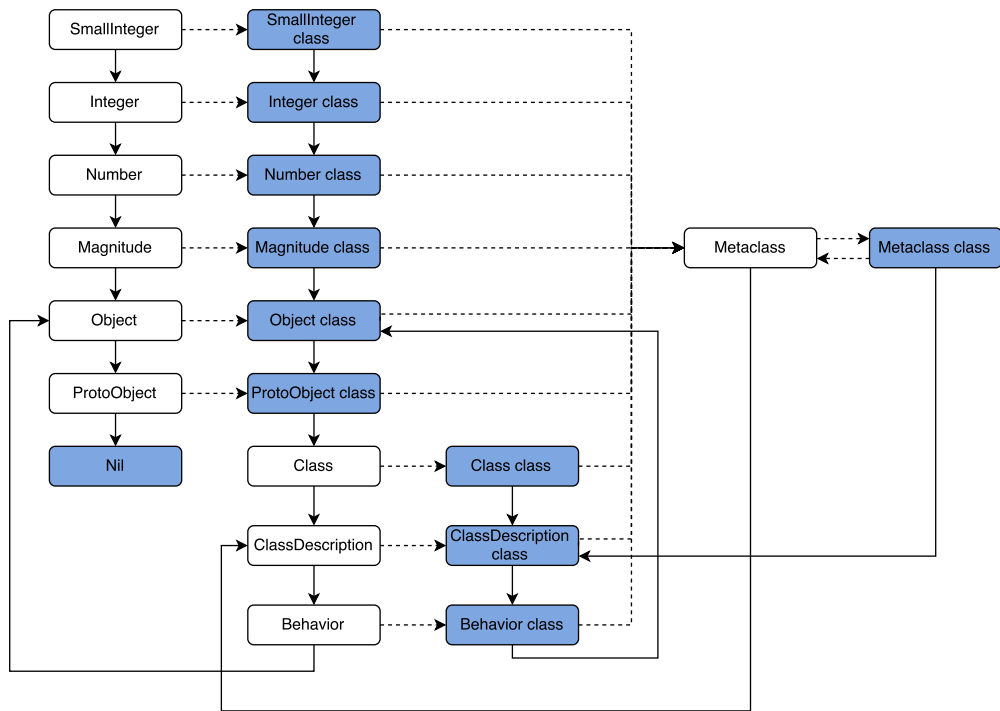


Figura 3: Metamodelo de Pharo4.0

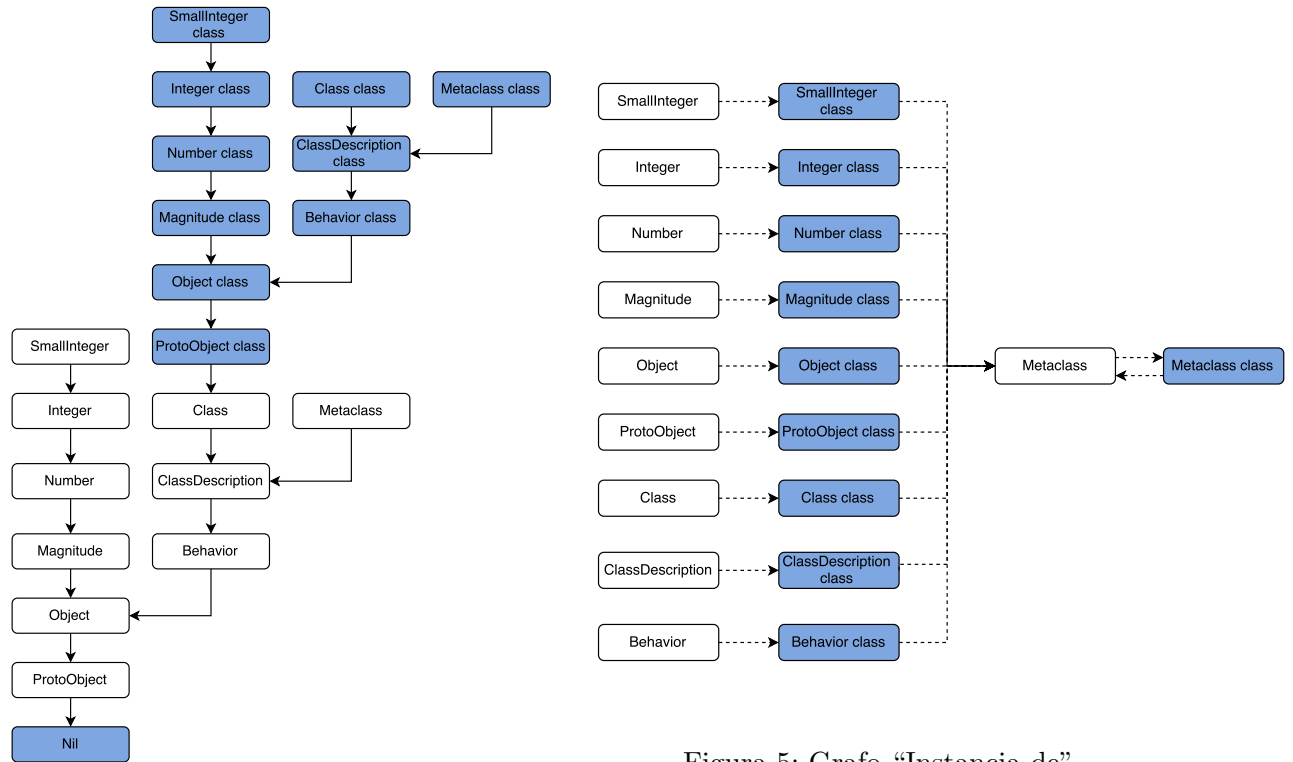


Figura 5: Grafo "Instancia de"

Figura 4: Grafo de Subclasificaciones

Figura 6: Metamodelo de Pharo4.0

Referencias: Línea punteada significa "Instancia de", y Línea lisa "Hereda de".

1. Los objetos en azul (¿las metaclasses?), solo tienen una instancia. No contestan al mensaje **new**. ¿Quién hace la alocaación por primera vez?

Si a una metaclass se le agregan variables de instancia que inicializo en su método **initialize** pero luego las modifiko, debo reinicializar la metaclass para que se apliquen los cambios a las variables. No es como el caso de las variables de instancia de una clase.

2. Cuando se envía **new** a cualquier objeto, el que aloca la memoria es **Behavior>>basicNew**.

- a) ¿Hace falta entonces dentro de cualquier implementación propia de **new** enviar la colaboración **super new** necesariamente? ¿Si no lo hago alguien se encarga de esto?
- b) Si es **Behavior** quien implementa la alocaación, ¿quién lo hace para **Object** y los que subclasifican de él? (El debugger no me deja meterme más adentro del **self new initialize** de **Object**)

- c) **ProtoObject new** tira el siguiente error y se rompe todo:

```
*** System error handling failed ***
Original error: MessageNotUnderstood: ProtoObject>>inspect.
```

¿Tiene algo que ver?

3. Enviarle la colaboración **superclass** a un objeto me devuelve lo que apunta la flecha lisa. Ej:

```
SmallInteger superclass  
>> Integer  
Metaclass superclass  
>> ClassDescription
```

4. Enviarle la colaboración `class` a un objeto me devuelve lo que apunta la flecha punteada. Ej:

```
SmallInteger class  
>> Smallinteger class  
Integer class class  
>> Metaclass
```

5. Enviar la colaboración `Metaclass new new` hace colgar Pharo. ¿Por qué? ¿Es el único objeto con el que pasa eso?