Colaboración y Herencia en diagramas de clases

¡Comencemos!

¿Qué aprenderás?

- Comprender el comportamiento de las relaciones entre los conceptos de herencia y colaboración.
- Crear Diagramas de clases de alto nivel para tener una mejor lectura del código que vamos a desarrollar.

Introducción

Comprenderemos la importancia de crear diagramas de clases con conceptos avanzados de programación.

Realizar y entender el buen uso de un diagrama de clases nos ayudará a obtener de manera más rápida y ágil el código, sin necesidad de mirar y analizar a fondo todos nuestros códigos.

¡Vamos con todo!

UML - Herencia y Colaboración

Conexiones y nomenclaturas de Herencia y Colaboración en UML

Existen distintos tipos de conectores dentro de un diagrama de clases, nosotros estudiaremos tres para continuar con lo aprendido en las unidades anteriores. Tenemos nomenclaturas de conexiones para identificar si la relación es:

- Directa, de tipo normal o arreglo.
- Bidireccional de tipo normal o arreglo.
- Tipo herencia.

A continuación, describiremos los tipos de relación:

- *Asociación*

Conexión entre clases, esta relación nos permite realizar colaboración directa entre ambos objetos tanto de forma directa como bidireccional, también nos permite relacionar objetos como arreglos.

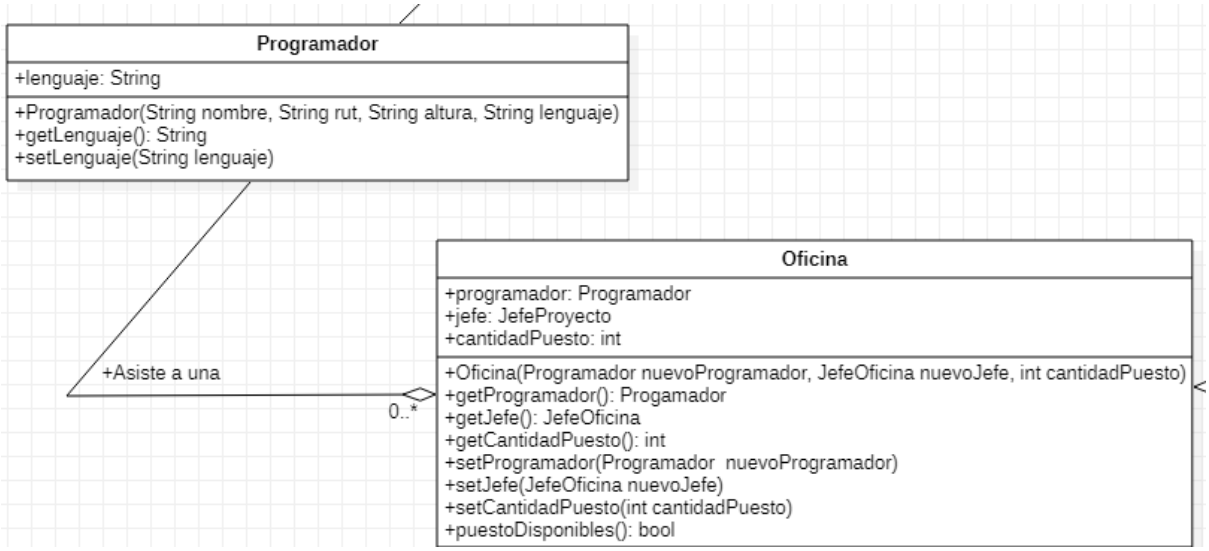

Imagen 1. Relación de una asociación de tipo directo de arreglos de objetos.
Fuente: Desafío Latam.

- *Generalización / Especialización*

Este tipo de conexión las usamos para representar herencia en un diagrama de clases.

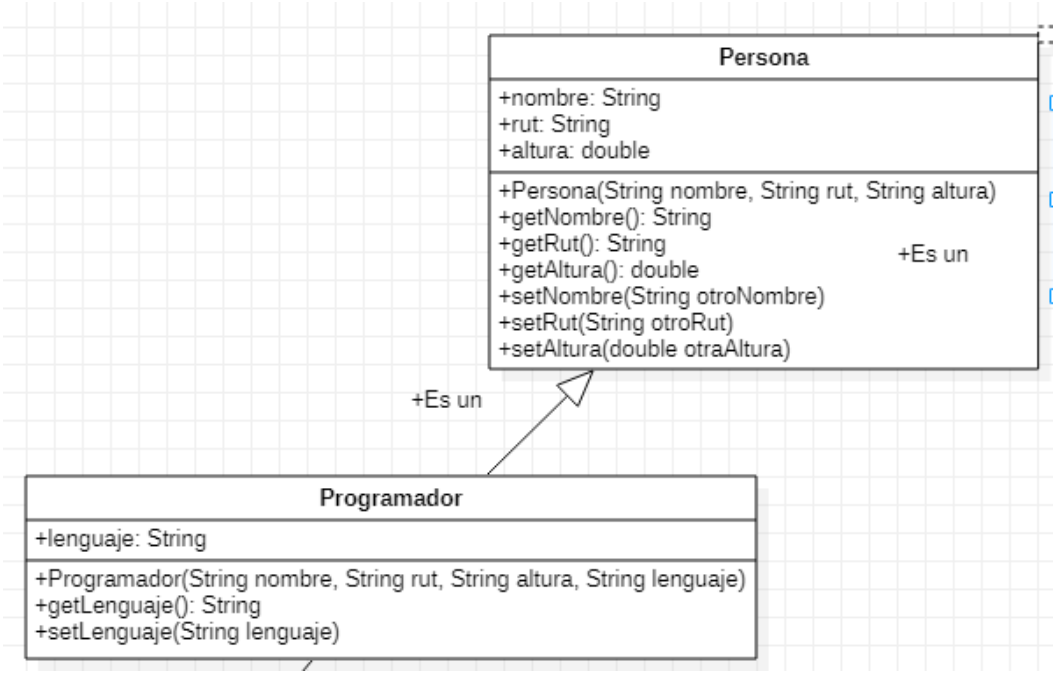

Imagen 2. Herencia Persona-Programador.
Fuente: Desafío Latam.

A continuación, describiremos la nomenclatura que se utiliza para las conexiones de clases:

Multiplicidad	Significado	Ejemplo
1	Uno y solo uno	Un Computador tiene solo un Mouse.
0..1	Cero a uno	Una Casa puede tener ninguna o una Piscina.
N..M	N hasta M	Una Juguetería tiene varios Juguetes.
*	Cero a varios	Una Biblioteca puede tener cero o varios Estudiantes.
0 .. *	Cero a Varios	Un Estacionamiento puede tener cero o varios Autos.
1..*	1 a varios	Un Escritorio tiene uno o varios Lápices.

Tabla 1. Nomenclatura conexiones de clases.
Fuente: Desafío Latam.

Para complementar la información de esta sección te invitamos a acceder al material complementario y revisar los links sugeridos.

Ejercicio guiado: Oficina de informática

Realizar un diagrama de clases para el personal de una oficina de informática, sabemos que en la oficina existen un jefe de proyecto y N programadores.

El **jefe de proyecto** tiene como atributos los siguientes campos:

- nombre.
- rut.
- altura.
- área.

Un **programador** tiene como atributos los siguientes campos:

- nombre.
- rut.
- altura.
- lenguaje.

Una **oficina** tiene la siguiente estructura:

- **programador:** Programador.
- **jefeProyecto:** JefeProyecto.
- **cantidadPuesto:** int.

Realizar un diagrama de clases utilizando herencia y colaboración.

Solución

1. Debemos realizar un análisis de las clases involucradas, en el ejercicio nombran tres: *Programador*, *Jefe de proyecto* y *Oficina*.

2. Después de identificar las clases, necesitamos analizar cuáles clases tienen relación y cuáles no, en este caso programador y jefe de proyecto tienen relación ya que ambos son personas.

3. Se crea una superclase de tipo *Persona* con los atributos nombre, rut, altura y se crea dos subclases, *Programador* con atributo lenguaje y *Jefe de proyecto* con atributo área.

4. Se utilizan las conexión de herencia entre *Programador, JefeProyecto* y *Persona*.

5. Se conecta la clase *Programador* y *JefeProyecto* con la clase *Oficina*.

6. La conexión de *Programador* con *Oficina* es de cero a muchos, ya que puede o no puede haber programador, pero si debe existir un Jefe de proyecto.

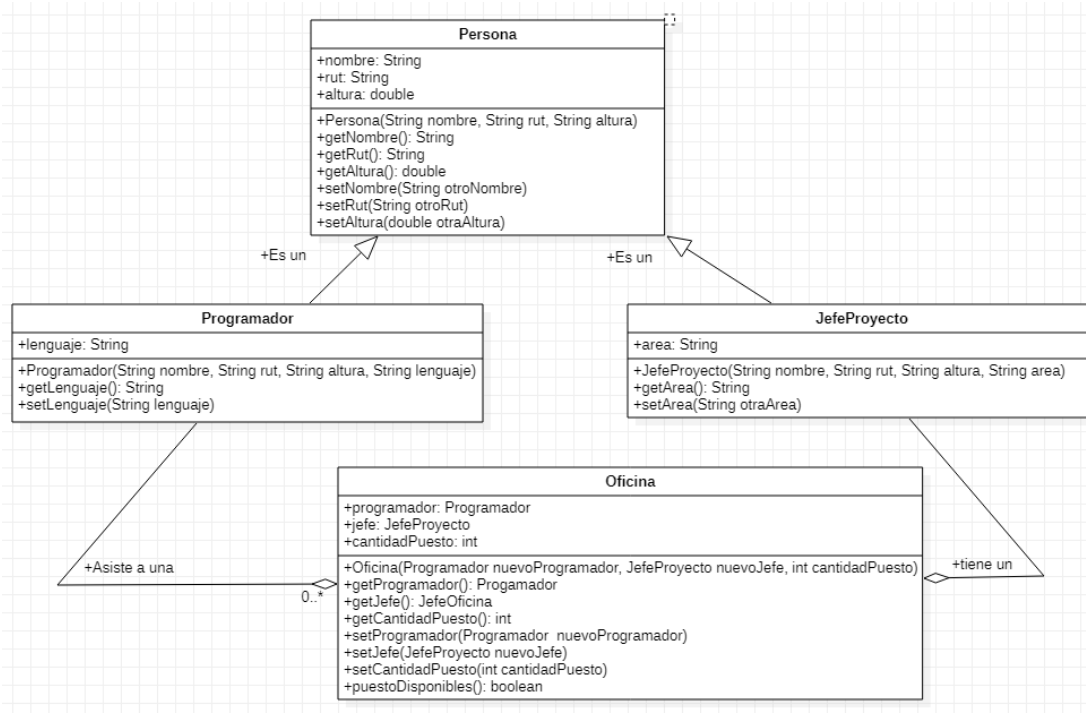

Imagen 3. Ejercicio resuelto en diagramas de Clases.
Fuente: Desafío Latam.

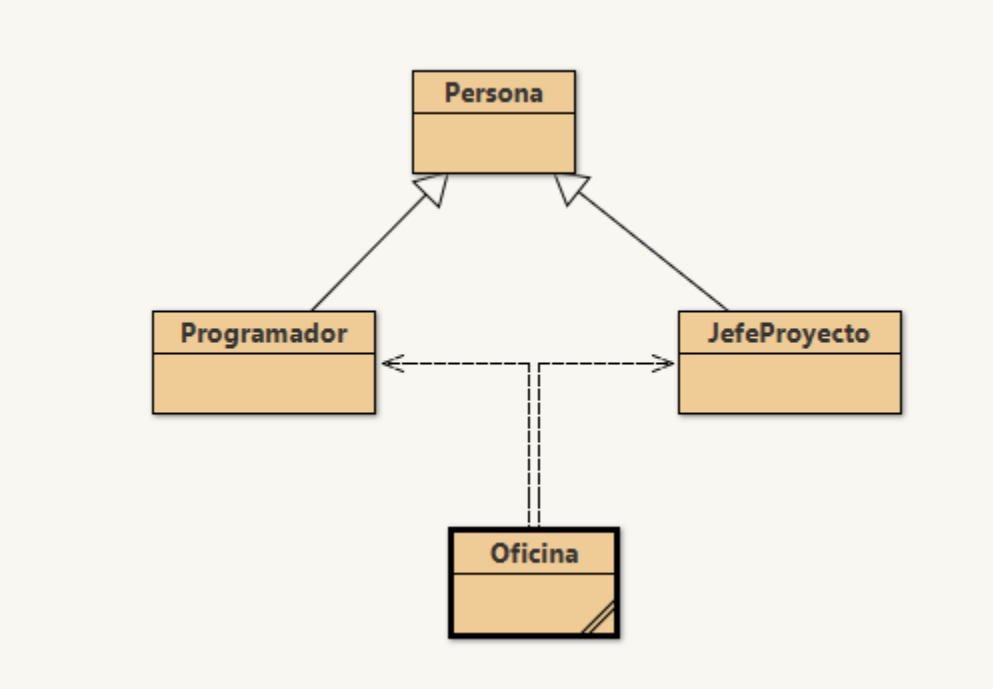

Imagen 4. Visualización del Ejercicio resuelto con clases Java.
Fuente: Desafío Latam,