# Programación orientada a objetos

## Colaboración y herencia en diagramas de clases

{desafío} latam_

***Utilizar elementos de la programación orientada a objetos para la implementación de una pieza de software que da solución a un problema de baja complejidad***

- Unidad 1: Flujo, ciclos y métodos
- Unidad 2: Arreglos y archivos
- Unidad 3: Programación orientada a objetos
- Unidad 4: Pruebas unitarias y TDD

***Te encuentras aquí***

{desafío} latam_

# ¿Qué aprenderás en esta sesión?

- *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.*

# ¿Cómo se relacionan los diagramas de clases con la herencia y colaboración?

# /* Herencia y colaboración */

{desafío}
latam_

# Conexiones y nomenclaturas en UML

Existen distintos tipos de conectores dentro de un diagrama de clases, nosotros estudiaremos tres para continuar con lo aprendido en las sesiones anteriores.

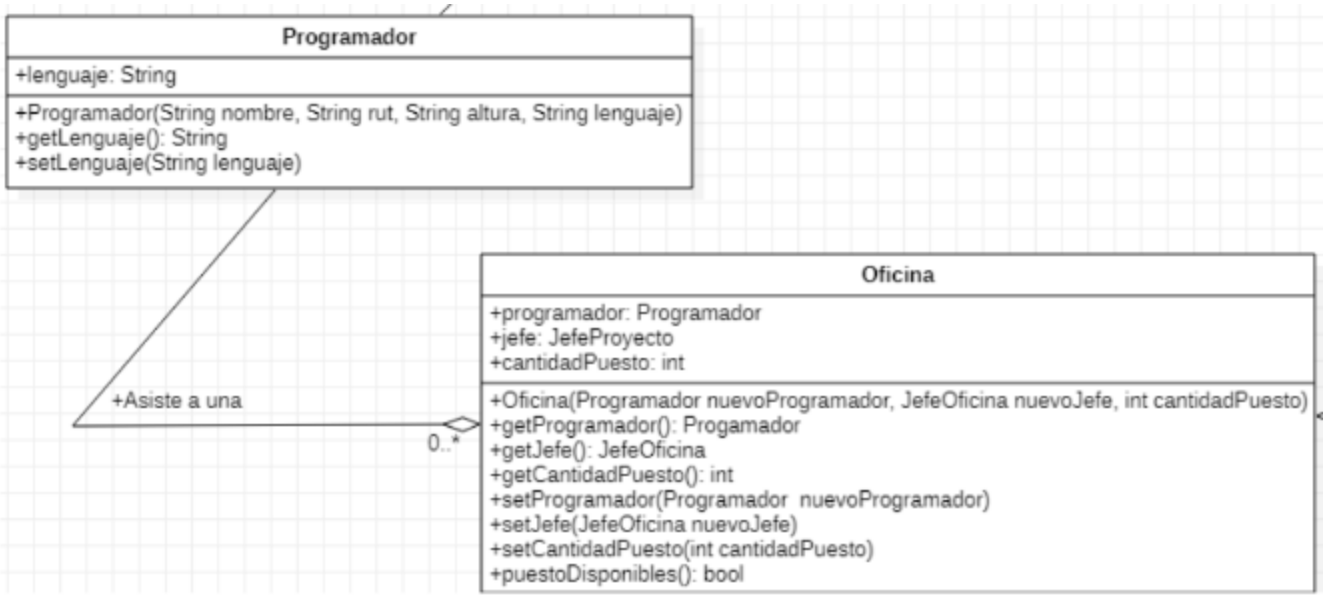
Tenemos nomenclaturas de conexiones para identificar si la relación es:

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

# Tipos de relación

## *Asociación*

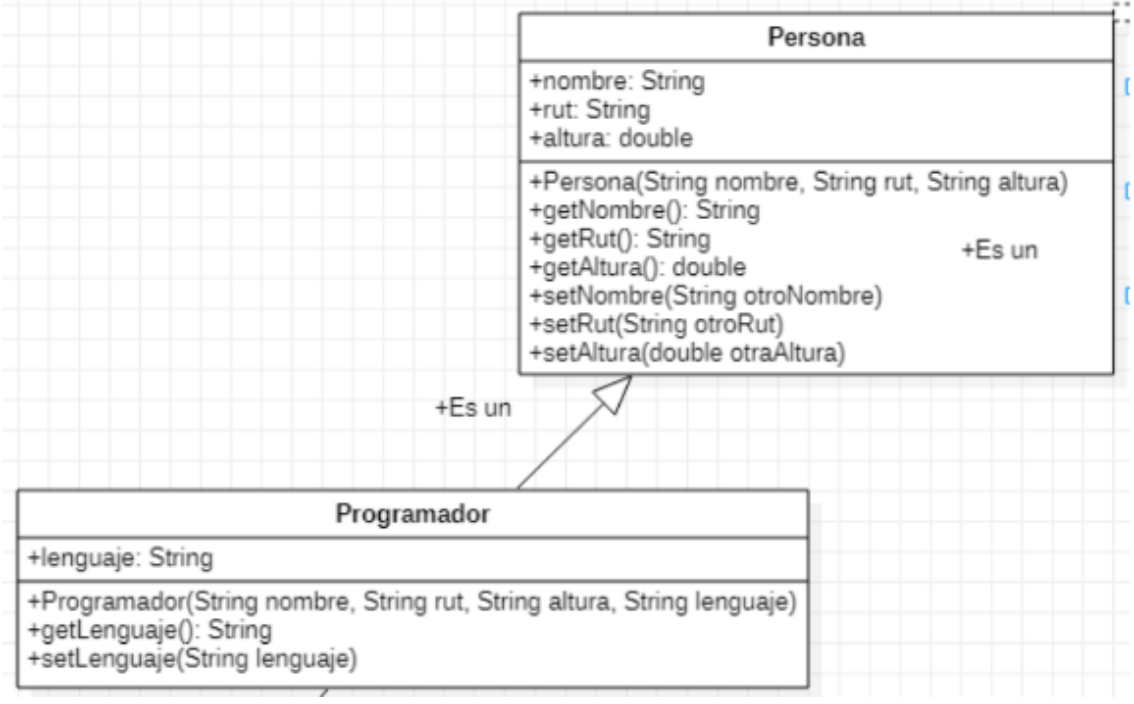
Permite realizar colaboración directa entre ambos objetos tanto de forma directa como bidireccional, también nos permite relacionar objetos como arreglos.



# Tipos de relación

## *Generalización / Especialización*

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




{desafío} latam_

# Nomenclatura 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. |

# 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.

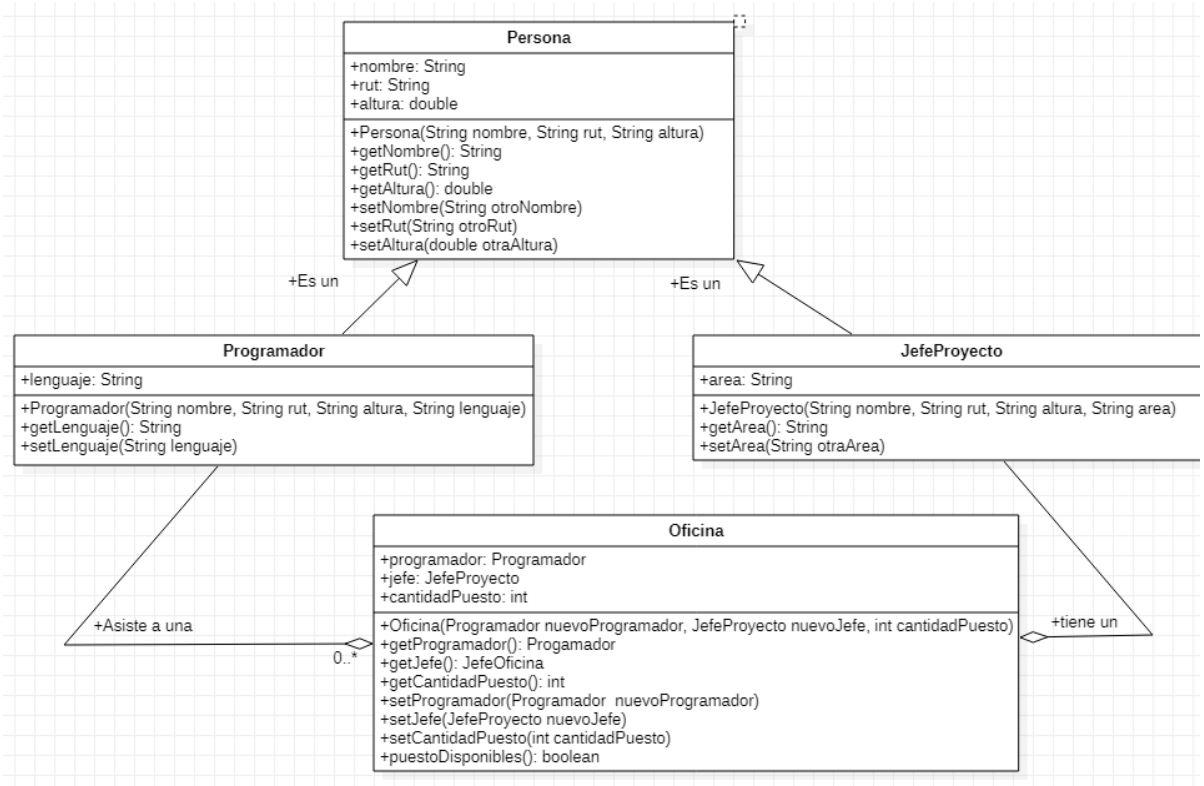
# Oficina de informática

## *Solución*

- Debemos realizar un análisis de las clases involucradas, en el ejercicio nombran tres: Programador, Jefe de proyecto y Oficina.
- 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.
- 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.
- Se utilizan la conexión de herencia entre Programador, JefeProyecto y Persona.
- Se conecta la clase Programador y JefeProyecto con la clase Oficina.
- 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.

# Oficina de informática

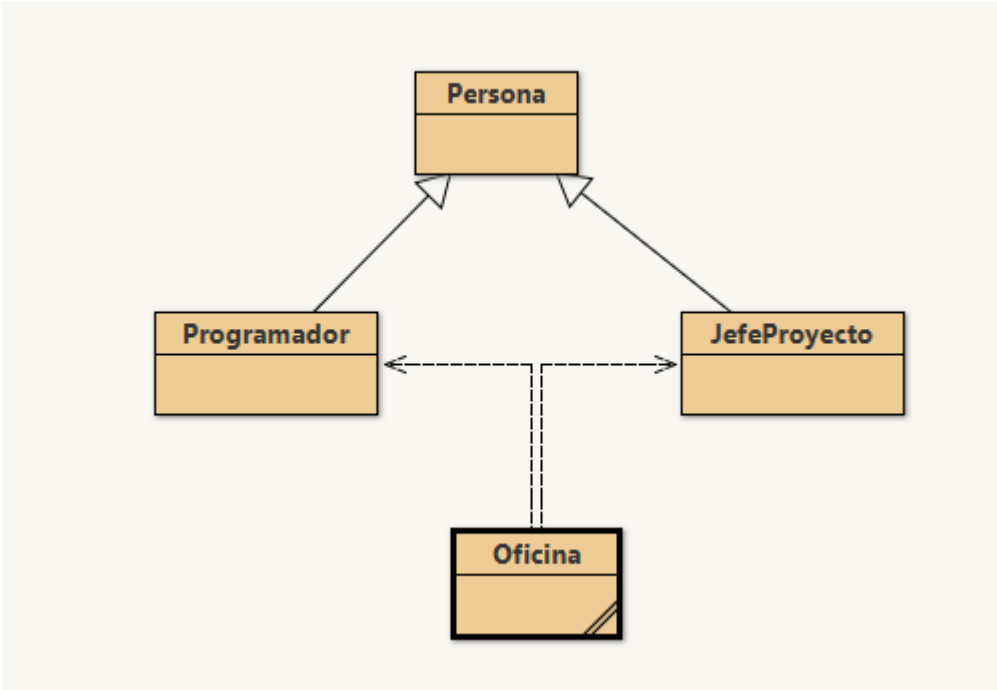
## *Solución*




{desafío} latam_

# Oficina de informática

*Solución*



# Oficina de informática

*Solución*

## ¡Compartamos nuestro desarrollo y comentemos!

{desafío} latam_

# Propongamos otro caso para trabajar en conjunto

{desafío} latam_

# ¿Para qué utilizaremos la relación de asociación directa?

{desafío} latam_

# Próxima sesión...

- *Desafío evaluado*

{desafío}
latam_

{desafío}
latam_

Academia de
talentos digitales