

Diagrama de Clases

Diagrama de Clases	1
¿Qué aprenderás?	2
Introducción	2
Componentes de un diagrama de Clase	3
Uso de la herramienta StarUML	4
Reconocer una clase Java dentro de un diagrama de clase	9
Nomenclatura para la construcción de una Clase	9
Ejercicio guiado: Elementos de una clase	12



¡Comencemos!

¿Qué aprenderás?

- Hacer uso de la herramienta StarUML para el diseño de diagramas UML.
- Reconocer una clase Java dentro de un diagrama de Clases para entender la lógica de la estructura de código.

Introducción

Conoceremos en profundidad lo que es un diagrama de clases y su mundo en la programación orientada a objetos, además, conoceremos una herramienta llamada StarUML que nos servirá para construir diagramas de clases.

Nos enfocaremos en describir una clase con sus atributos y operaciones para así tener una perspectiva descriptiva de lo que es una clase en diagrama de clases como en Java.

Existen otras herramientas para realizar diagramas como por ejemplo:

- Lucidchart
- diagrams.net
- argoUML
- Visio, entre otras.

¡Vamos con todo!



Componentes de un diagrama de Clase

Un diagrama de clases se compone de la siguiente estructura:

1. Nombre del Objeto a modelar.
2. Atributos o propiedades.
3. Tipo de dato del atributo.
4. Acciones u Operaciones.

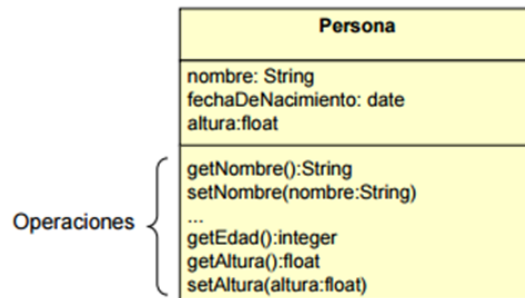


Imagen 1. Ejemplo diagrama de clases.

Fuente: Desafío Latam.

- **Nombre de Objeto:** Persona
- **Atributos:** nombre , tipo de dato String
 - **fechaDeNacimiento:** tipo de dato Date.
 - **altura:** tipo de dato Float.
- **Acciones u Operaciones:**
 - getNombre(): String
 - setNombre (String nombre)
 - getEdad() : integer
 - getAltura(): float
 - setAltura (float altura)

Uso de la herramienta StarUML

Esta herramienta nos permite realizar todos los tipos de diagramas que se explicarán en esta unidad, su uso es libre aunque si lo pagamos tendremos acceso a más opciones (para los que deseen ahondar en el diseño con esta herramienta existen planes mensuales y anuales). Para efectos de uso en la unidad, la versión libre nos será suficiente.

Para instalar la herramienta, debemos ir a la siguiente URL: <http://staruml.io/download>

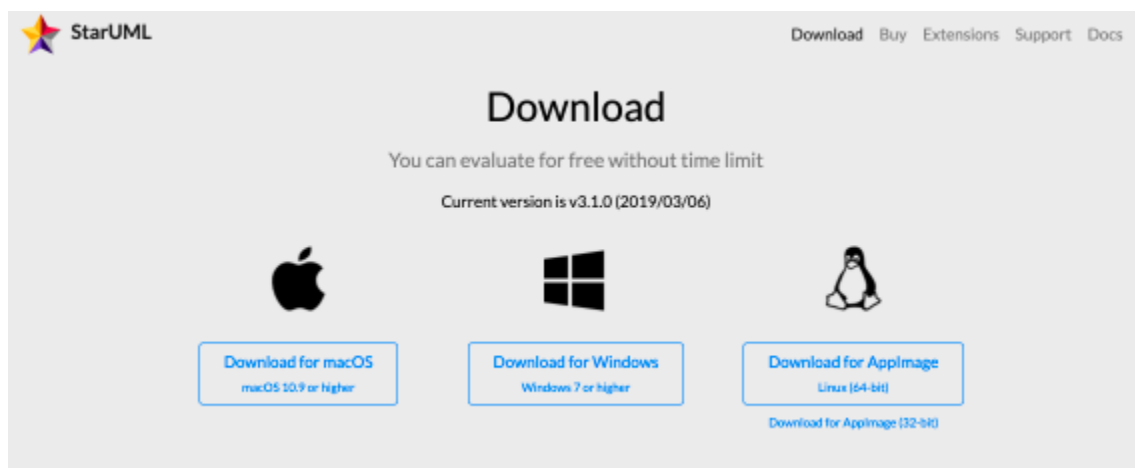


Imagen 2. Página de descarga StarUML.

Fuente: Desafío Latam.

Para crear un diagrama de clases realizaremos la siguiente acción:

Model -> Add Diagram -> Class Diagram

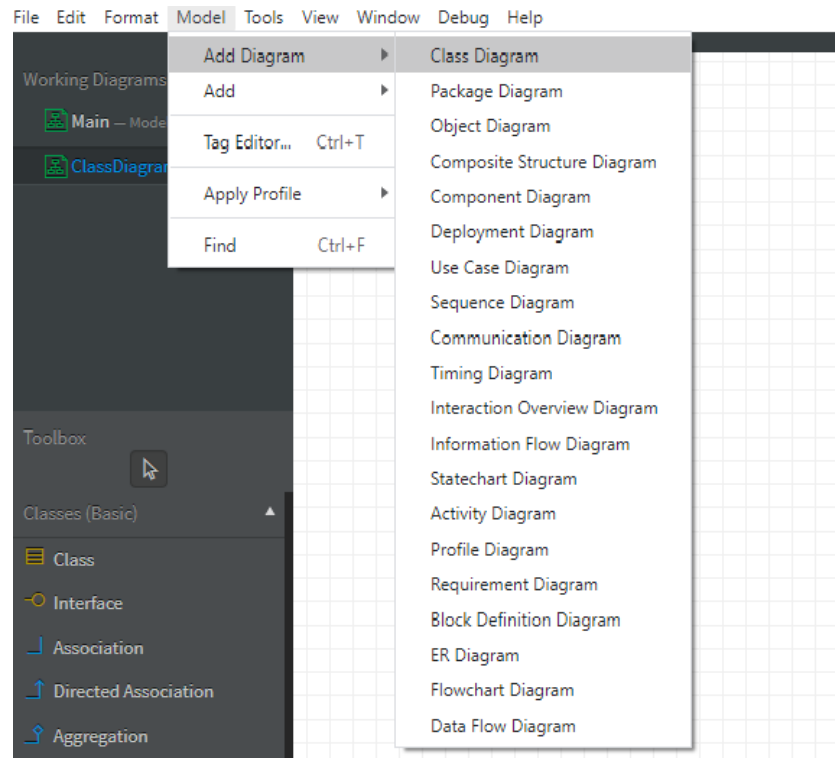


Imagen 3. Crear Diagrama de clases StarUML.

Fuente: Desafío Latam.

Para crear una Clase damos clic en “class” y dibujamos un cuadrado con el mouse en el espacio en blanco.

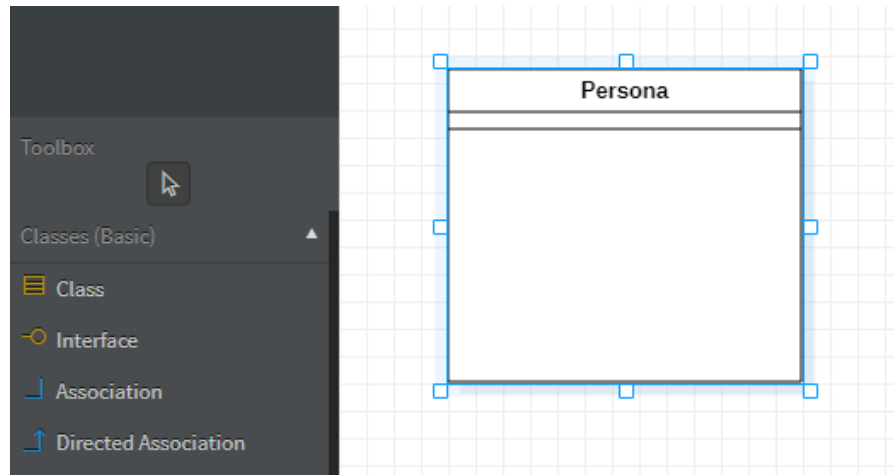


Imagen 4. Creación de clase.
Fuente: Desafío Latam.

Para cambiar nombre de la clase solo debemos dar doble clic en la caja y cambiar el nombre, en el ejemplo es “Persona”.

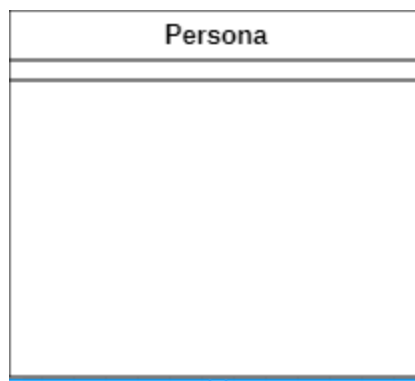


Imagen 5. Cambiar nombre de clase.
Fuente: Desafío Latam.

Para agregar atributos a la clase seleccionamos el icono y empezamos agregando el nombre de atributo y tipo de dato:

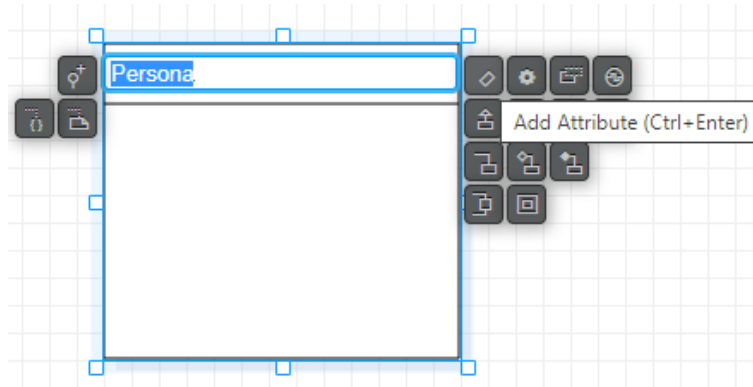


Imagen 6. Agregar Atributos.
Fuente: Desafío Latam.

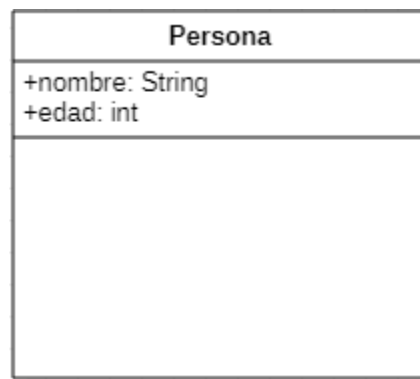


Imagen 7. Agregando atributos y tipo de dato.
Fuente: Desafío Latam.

Agregando acciones u operaciones:

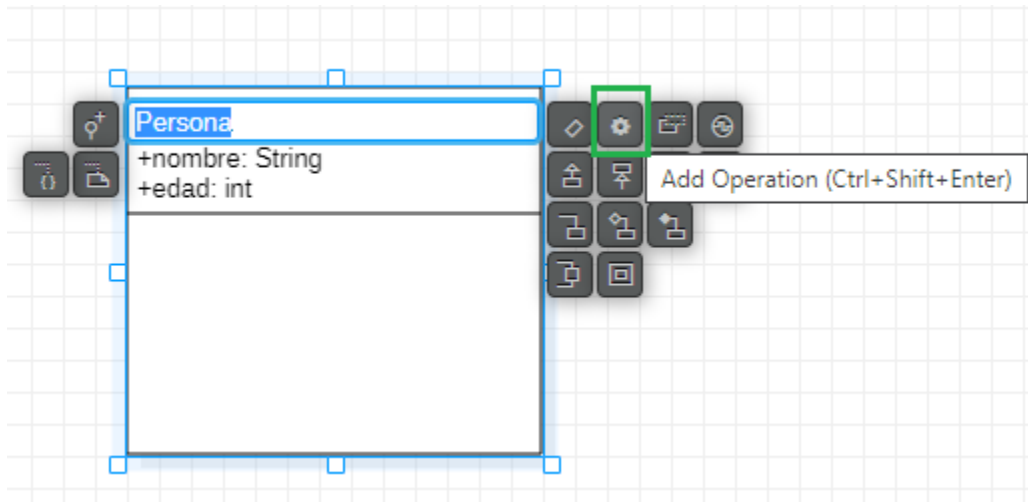


Imagen 8. Botón que permite agregar Operaciones o Acciones.
Fuente: Desafío Latam.

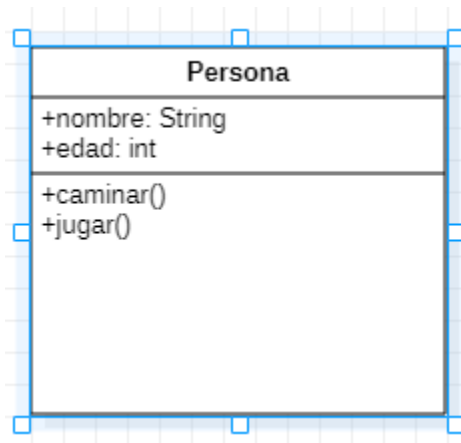


Imagen 9. Operaciones agregadas.
Fuente: Desafío Latam.

Reconocer una clase Java dentro de un diagrama de clase

Antes de comenzar a explicar qué es una clase en Java, primero explicaremos qué es un objeto en Java.

Un objeto en Java es cualquier elemento de la vida real que sea tangible. Este elemento se compone por atributos o propiedades con su tipo de datos asociado, los atributos siempre son privados, constructores, getter (accesadores), setter (mutadores) y eventos propios del objeto.

Cuando se programa un Objeto en Java, este objeto pasa a llamarse Clase ya que desde el modelo ideal del objeto se construye en Java para su ejecución.

En pocas palabras, un Objeto es cualquier cosa tangible de la vida real y una clase es el Objeto en ejecución con operaciones programables dentro del lenguaje Java.

Nomenclatura para la construcción de una Clase

Nombre de la clase: El nombre de la clase siempre debe comenzar con mayúscula, si el nombre de la clase es compuesta (más de una palabra), cada palabra debe comenzar con mayúscula.

Ejemplo:

- Persona: Clase con nombre simple.
- PersonaJuridica: Clase con nombre compuesto.

Atributos o Propiedades: El nombre debe ser lo más descriptivo posible por norma y por seguir buenas prácticas de programación. La regla de nomenclatura es la siguiente:

- Siempre se escribe la primera letra en minúscula para nombre simples y, si es un nombre compuesto, el segundo nombre debe comenzar siempre con mayúscula, todo esto acompañado con el tipo de dato y su tipo de acceso.

Ejemplo:

- *private* - Tipo accesor.
- *int* - Tipo de dato.
- *atributo* - Nombre (simple) - fechaNacimiento (nombre compuesto).

private int nombre

private int fechaNacimiento

Constructores: Siempre lleva el mismo nombre de la Clase, y puede recibir o no parámetros. Esto lo veremos en detalle en la unidad "Orientación a Objetos".

Operaciones: Las operaciones se forman por su tipo de acceso, público o privado, además sabremos si este retorna o no retorna algún tipo de dato.

El nombre del método siempre debe ser representado por el nombre de una acción, la nomenclatura del nombre del método siempre es con minúscula si este es un nombre simple; si el nombre del método es compuesto la segunda palabra del nombre del método debe comenzar con su primera letra mayúscula; si el nombre del método tiene dos o más palabras compuestas en su nombre cada palabra adicional debe comenzar con letra mayúscula.

El método u operación puede o no recibir parámetros de entradas.

Ejemplos:

- *private* - Tipo accesor.
- *int* - Retorno de tipo de dato entero.
- *void*: No retorna ningún valor.
- *enviarMensajePersonal* - Nombre de método compuesto.
- *salir* - Nombre de método simple.
- Método de nombre simple, sin parámetros y no retorna ningún valor:

```
public void imprimir() { }
```

- Método de nombre simple, con un parámetro y no retorna ningún valor:

```
public void imprimir(String input) { }
```

- Método con nombre simple, sin parámetros y que retorna un valor:

```
public int numero() { return 3 ; }
```

- Método con nombre simple, con parámetros y que retorna un valor:

```
public int numero(int valor ) { return valor ; }
```

Tomando el ejemplo anterior veremos una clase Java orientada a Diagrama de Clase.

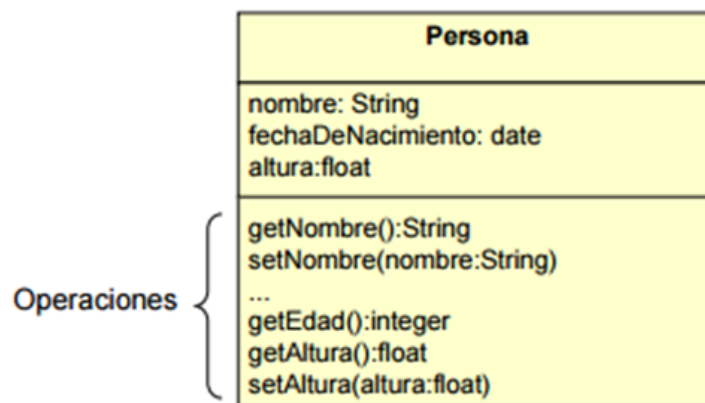


Imagen 10. Diagrama de clases orientado a objetos en Java.
Fuente: Desafío Latam.

Ejercicio guiado: Elementos de una clase

Crear un diagrama de clases para establecer elementos de una casa, la información que se tiene es la siguiente:

- Cocina
 - altura : double
 - cantidad hornillas : int
 - prenderHorno() : String
- Televisor
 - marca : String
 - definición : String
 - apagarTelevisor()
- Escritorio
 - tamaño : double
 - tipo : String

Solución ejercicio guiado

Para esto utilizaremos starUML:

- Crear cada clase con sus atributos y operaciones.

Lo primero es identificar los objetos, sus atributos y sus operaciones. Posteriormente se debe seguir las normas de la creación de un diagrama de Clases.

La siguiente imagen ilustra la solución con starUML:

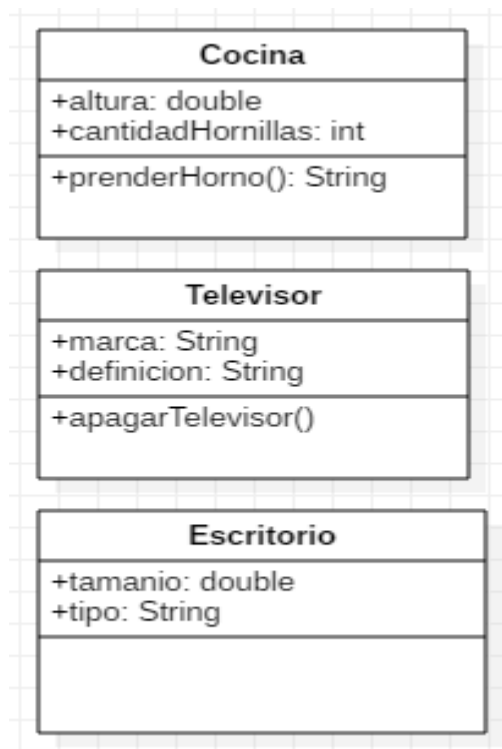


imagen 11 : Solución Ejercicio Guiado.
Fuente: Desafío Latam.