

DIAGRAMA DE CLASES



The image is a collage of screenshots from the video game DOOM. It shows various levels including Hell, Helltrain, and Helltrain Station. The screenshots feature different weapons like the Super Shotgun, Plasma Gun, and BFG9000. It also shows various enemies like Hell Knights, Hell Dukes, and Hell Marine. The collage is set against a dark background with a large watermark in the center that reads "AWAKELAB #programmingbootcamp".



DIAGRAMA DE CLASES

¿QUÉ VAMOS A VER?

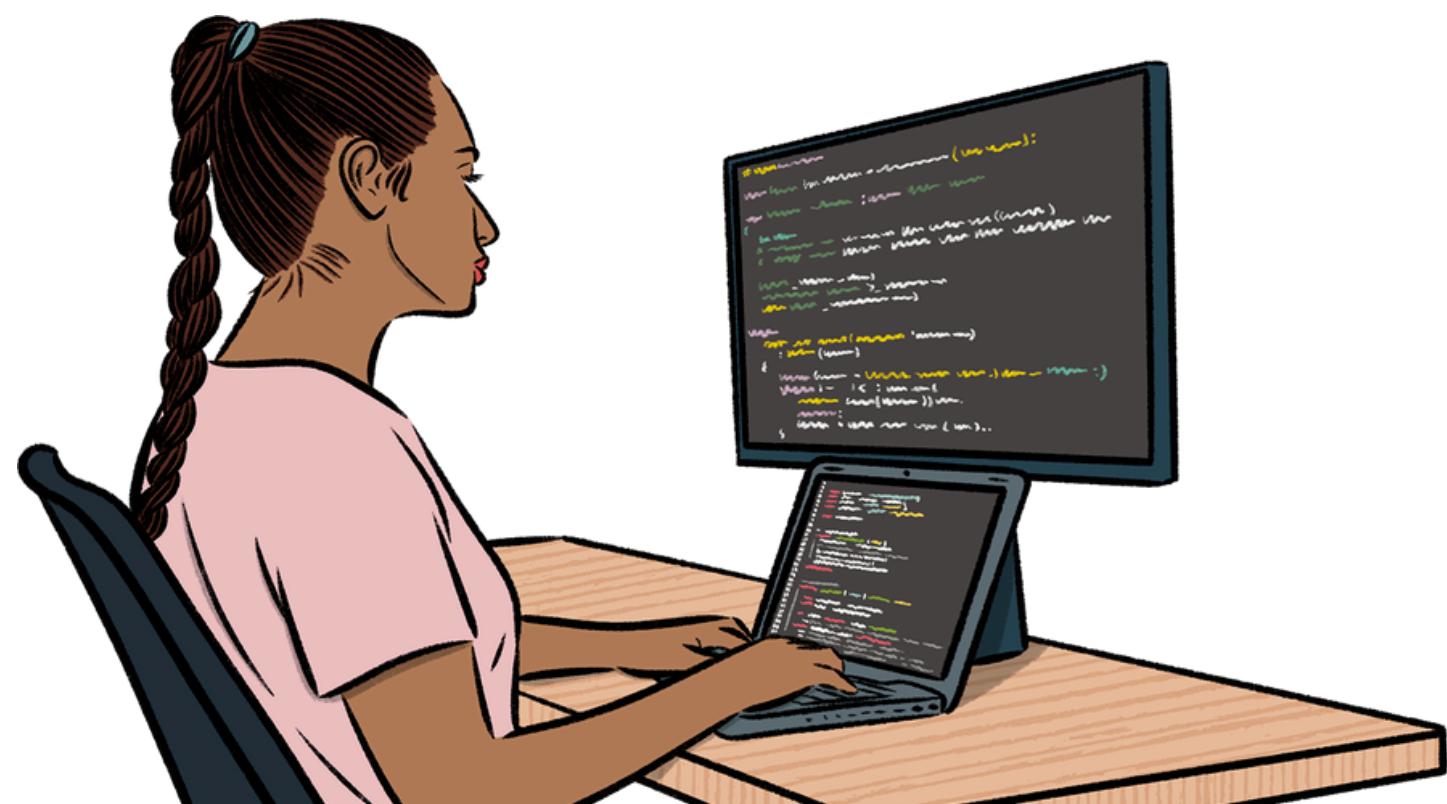
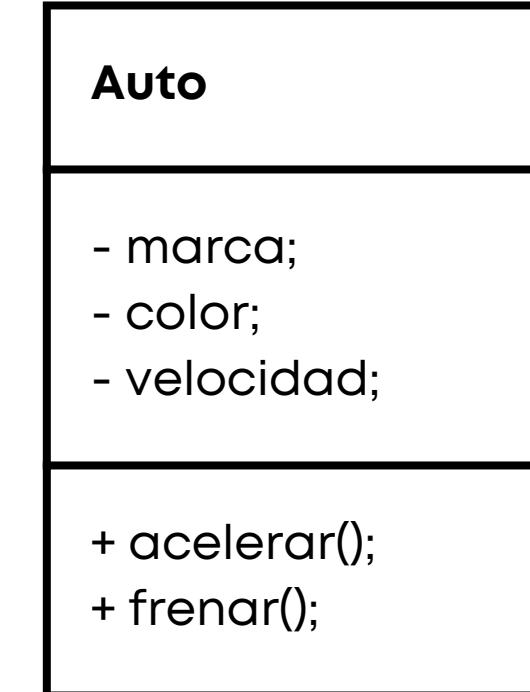
- Diagrama de clases.
- Relaciones estáticas.
- Relaciones de asociación.





DIAGRAMA DE CLASES

- Es una herramienta del modelado UML (En español, lenguaje unificado de modelado) que sirve para representar el diseño de un programa bajo el paradigma de POO.
- Nos permite visualizar atributos, métodos y las relaciones entre diferentes clases.
- Su representación gráfica se divide en tres “sectores”: *nombre de la clase*, *atributos* que contiene y *métodos encontrados*.





RELACIONES ESTÁTICAS

ASOCIACIÓN

- Es el tipo de relación más general.
- Representa un conjunto de enlaces entre dos objetos.
- Ejemplo, una persona trabaja para una empresa.
- Una clase puede tener una asociación de “uno a uno”, “uno a muchos” o “muchos a muchos”.



RELACIONES ESTÁTICAS



HERENCIA

- Es el mecanismo que permite a una clase incorporar atributos y métodos de otra clase, sumándolos a los que ya posee.
- La clase de la cual se hereda se denomina **superclase**, y la que hereda es la **subclase**.
- A partir de una superclase se pueden obtener varias subclases, por ejemplo, de una clase Empleado, se puede obtener Secretaria, Técnico, o Ingeniero.



RELACIONES ESTÁTICAS



AGREGACIÓN

- Conlleva una especie de relación jerárquica entre los objetos.
- Nos encontramos ante un gran objeto que contiene a clases más pequeñas.
- Los objetos “son-parte-de” otro objeto más complejo.
- Por ejemplo, motor, ruedas y carrocería tranquilamente pueden ser clases independientes, pero todas en sí son un eje importante dentro de Automóvil.





RELACIONES ESTÁTICAS

COMPOSICIÓN

- Es una forma de agregación.
- La relación de propiedad es más fuerte. Es decir, que es una forma de modelar la estructura estática del sistema.
- Si tenemos el sistema que controla una máquina de café, existe una relación directa entre la clase **máquina**, que sería un todo, y la clase **producto**, que se refiere a la finalidad o algo más particular.

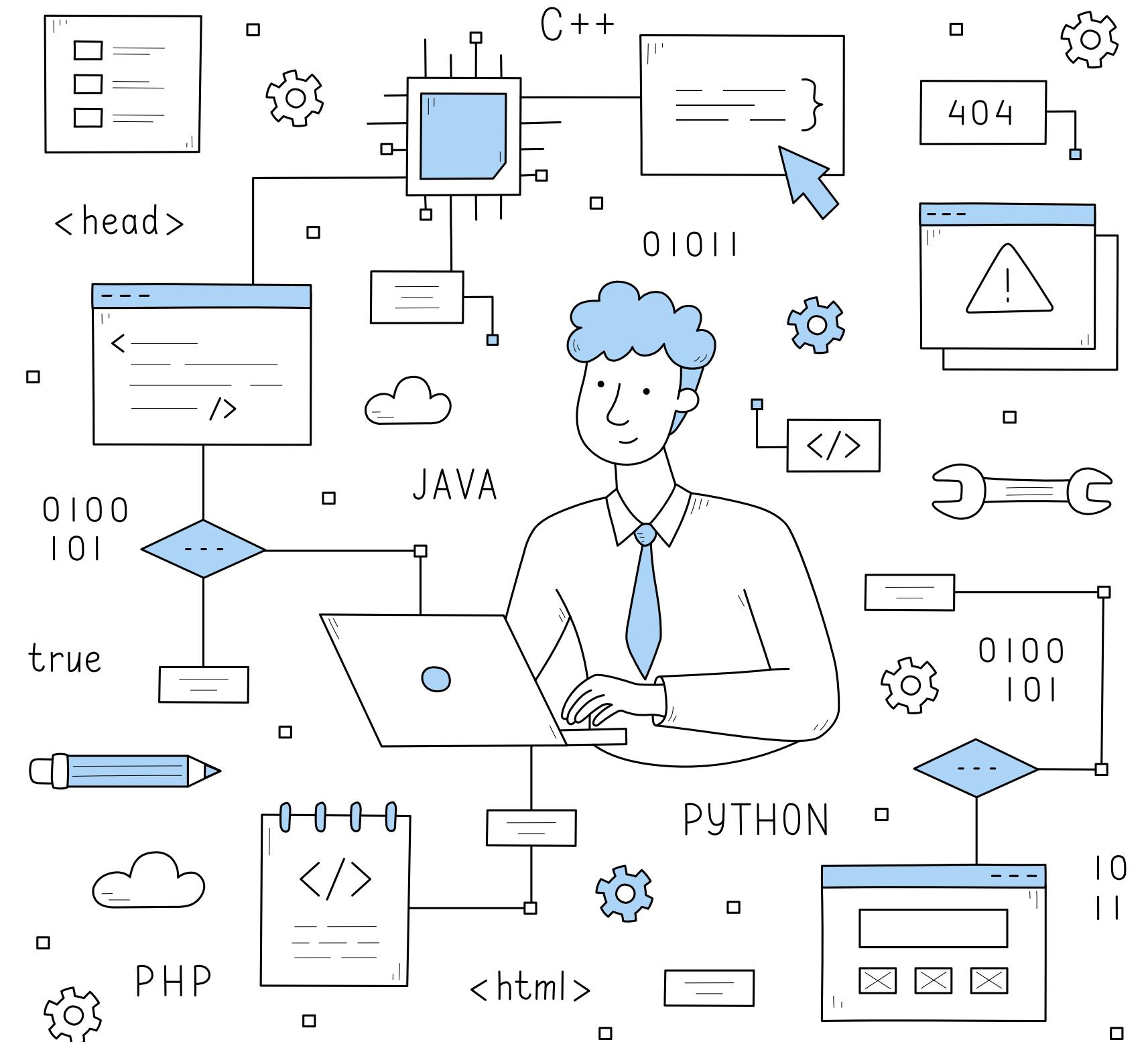




RELACIONES ESTÁTICAS

DEPENDENCIA

- Representa a dos clases o instancias que dependen netamente la una de la otra.
- Indica que una clase requiere de la otra para proporcionar alguna información o incluso llamar a los métodos que la vuelven enteramente funcional.

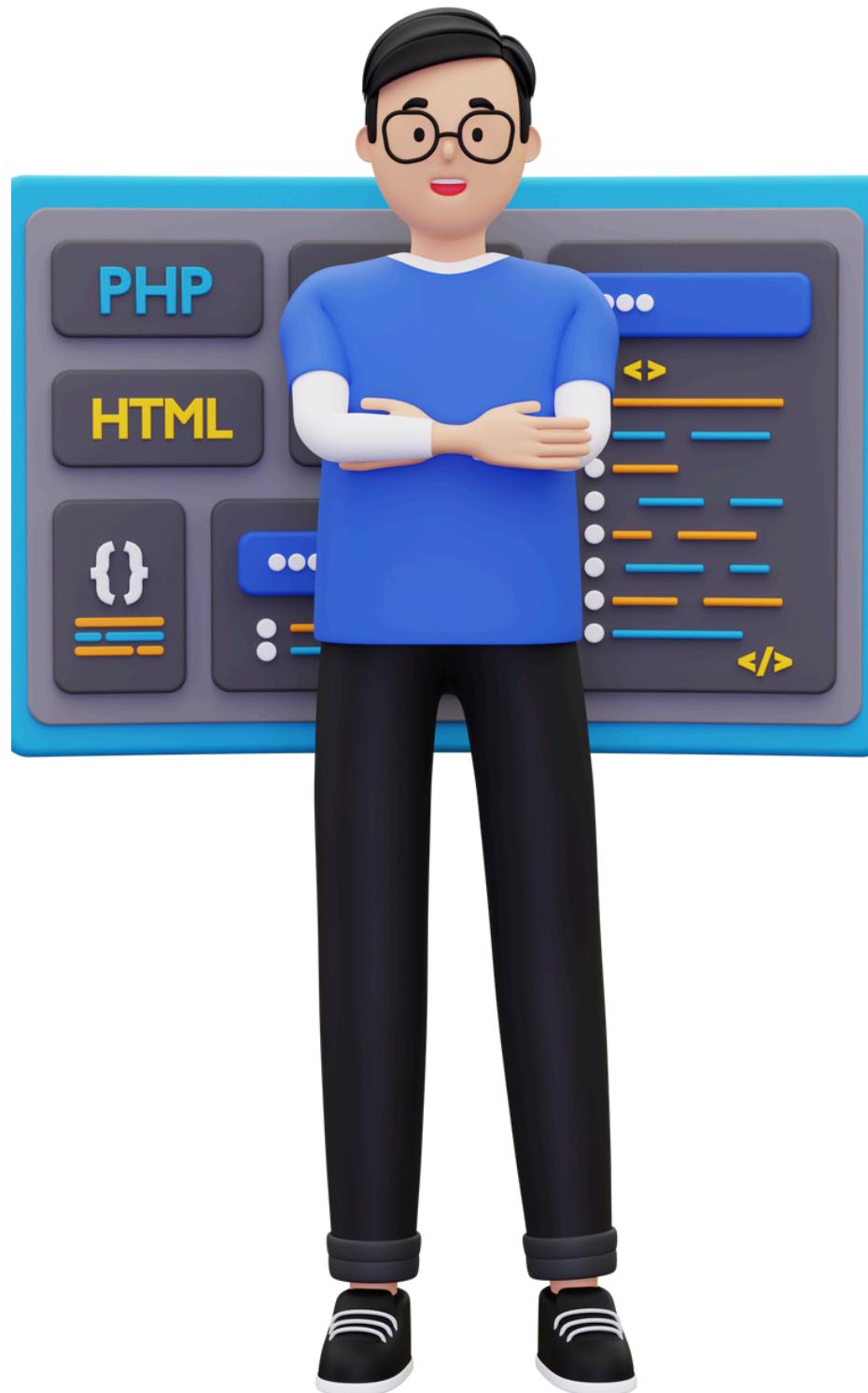


RELACIONES ESTÁTICAS



INTERFACES

- Es un conjunto de operaciones de una clase.
- Solo funciona cuando la clase en cuestión es visible desde otras instancias o paquetes.

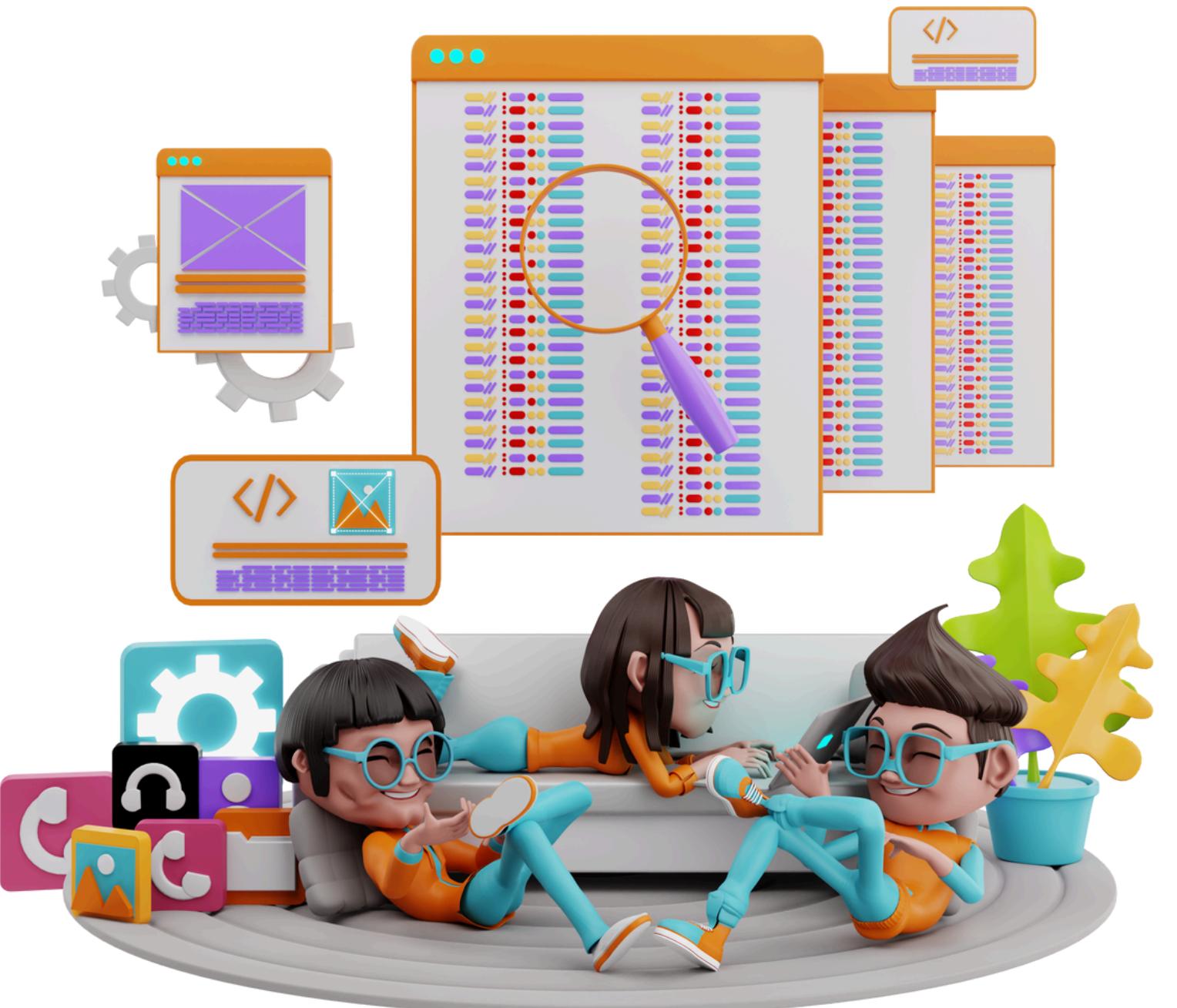




RELACIONES ESTÁTICAS

PAQUETES

- Se usan para dividir el modelo de las clases.
- Agrupan a otras clases o incluso más paquetes según el criterio aplicado para organizar los algoritmos.
- Las dependencias que pueden existir entre los paquetes se crean a partir de las relaciones existentes entre los elementos que se agrupan dentro de ellos.



RELACIONES ESTÁTICAS



CLASES

- Se representa como una caja, separada en tres secciones.
 - **Superior:** Nombre de la clase.
 - Se encuentra centrado dentro de la caja y en caso de pertenecer a una Clase Abstracta, se representa en cursiva.
 - **Central:** Lista de atributos.
 - visibilidad nombre : tipo = valor-inicial
 - La visibilidad se representa por + (public), - (private) o # (protected).
 - **Inferior:** Lista de métodos.
 - visibilidad nombre(parámetros) : tipo de retorno





RELACIONES DE ASOCIACIÓN

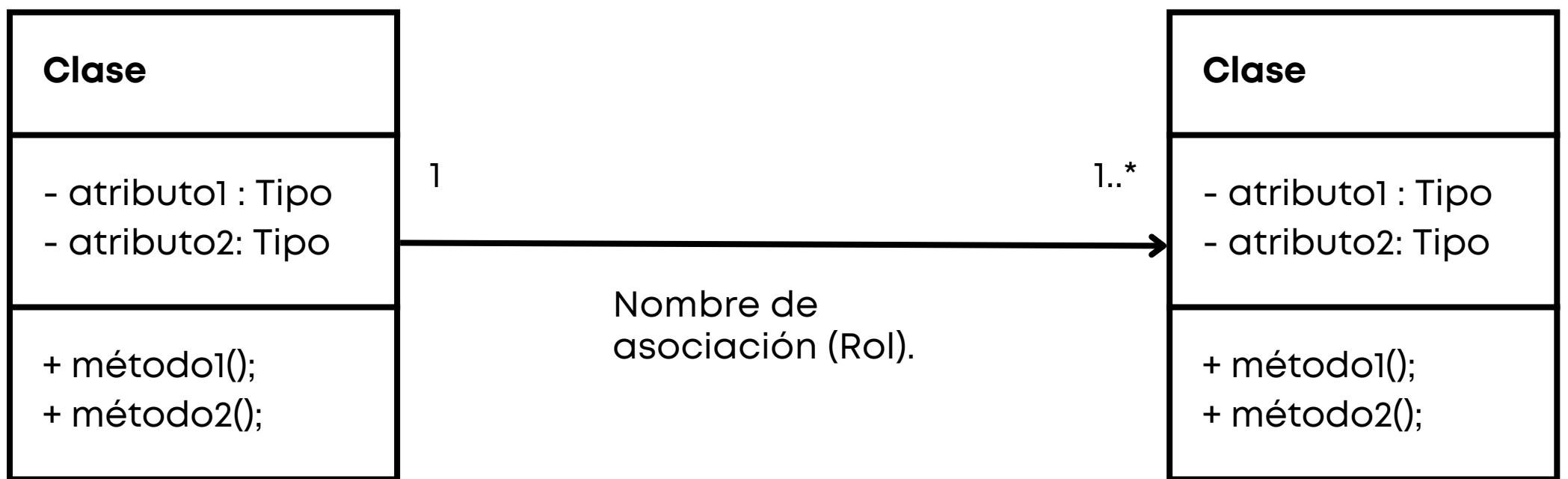
- Nos presentan la posibilidad de que la instancia de una clase puede estar relacionada o conectada con otra instancia de la misma clase.
- Posee diferentes propiedades de relación:
- **Multiplicidad:** Puede ser un número concreto, un rango o una colección. La letra “n” y el símbolo “*” representan cualquier número.
- **Orden:** Se agrega en caso de que las instancias guarden un criterio de ordenamiento. Su palabra clave es “{ordered}”.





RELACIONES DE ASOCIACIÓN

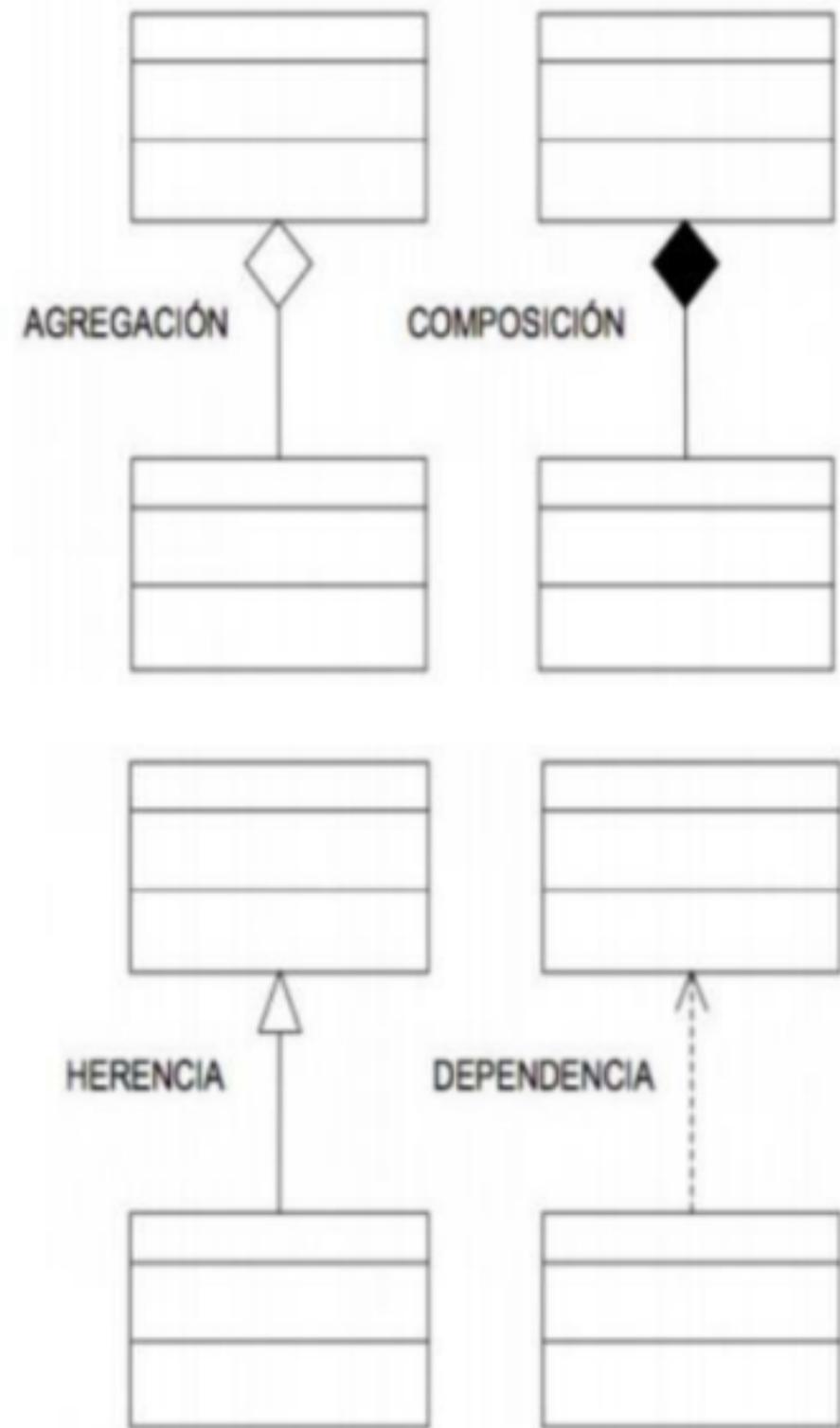
- **Navegabilidad:** Es el pase de una clase a otra y se representa con una flecha sin relleno, indicando el sentido de la navegación.
- **Rol:** Expresa la forma en que una clase hace uso de otra mientras se mantiene la asociación.





RELACIONES DE ASOCIACIÓN

- También existen otros tipos de notaciones específicas para las relaciones.
- **Agregación:** Se representa con un rombo hueco en la clase cuya instancia es una agregación de otra.
- **Composición:** Se representa con un rombo lleno en la clase cuya instancia contiene las instancias de otra clase.
- **Dependencia:** Corresponde a una línea discontinua con una flecha apuntando a la clase cliente.
- **Herencia:** Esta relación se representa como una línea continua con una flecha hueca en el extremo que apunta a la superclase.





RELACIONES DE ASOCIACIÓN

INTERFACES

- Se representa por una caja similar a las Clases, con la diferencia de que solo tiene dos compartimentos.
- En la zona superior se almacena el nombre de la interfaz, y en la inferior, la lista de métodos que la van a componer.
- Existe una forma más simple de representar una interfaz, y es con un círculo pequeño y hueco, que debajo lleva el nombre, más no los métodos.

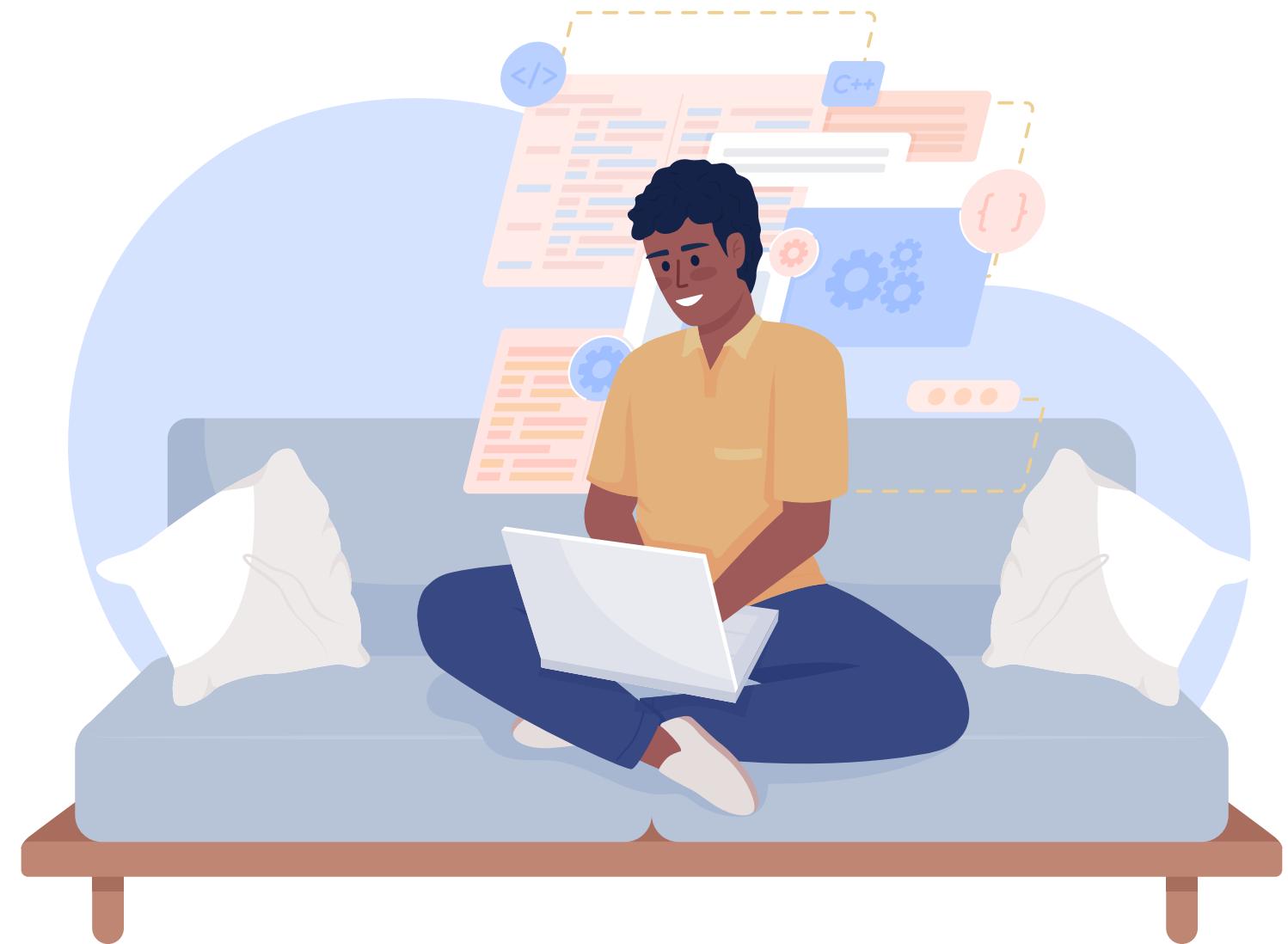
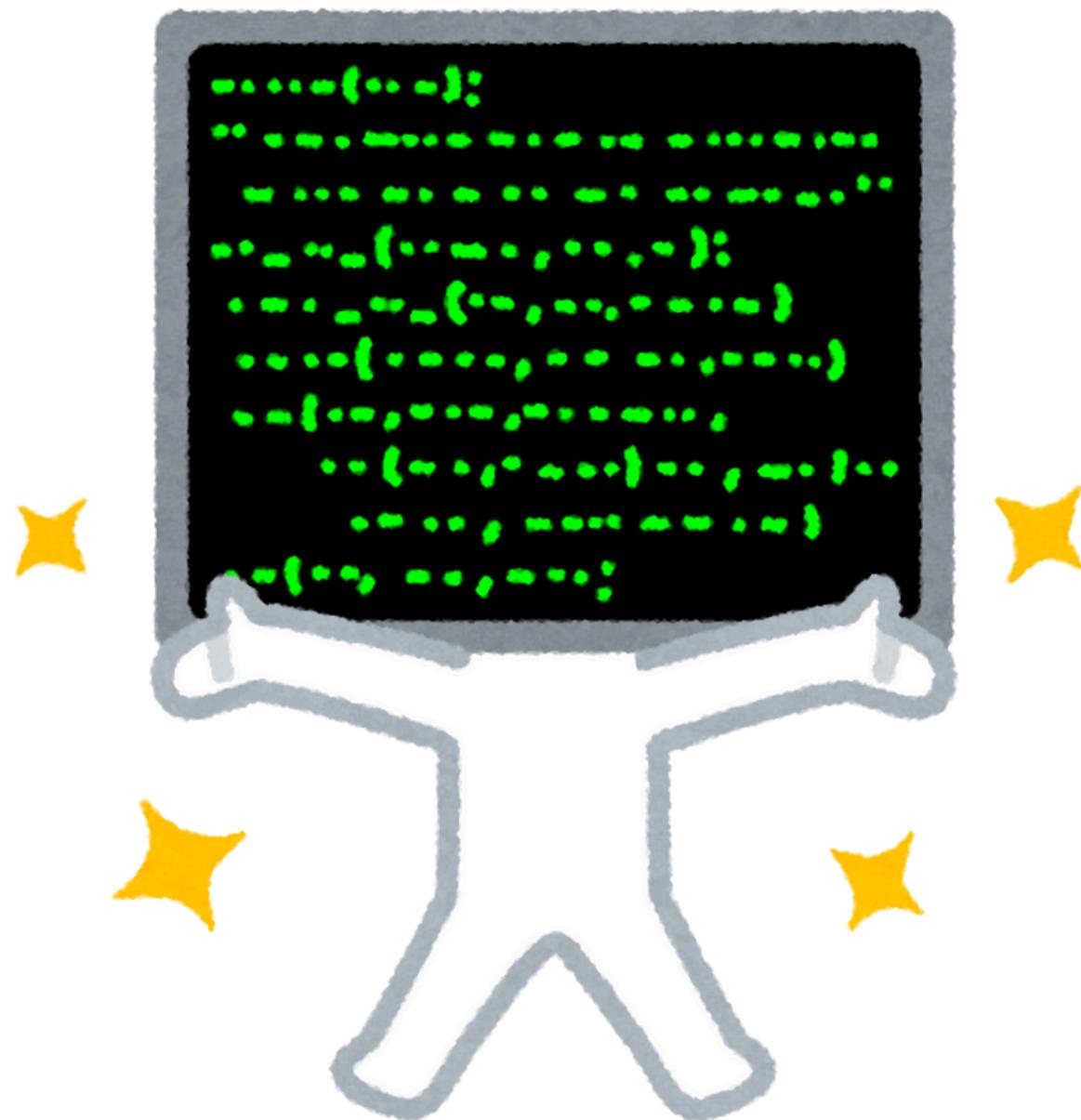




DIAGRAMA DE CLASES

EJEMPLO

- Desarrolle un diagrama de clases para un sistema encargado de la gestión de préstamos y reservas de libros y revistas de una biblioteca.





AWAKELAB

#programmingbootcamp

nodovirtual.awakelab.cl

 jELOU futurejob by  Adalid Chile