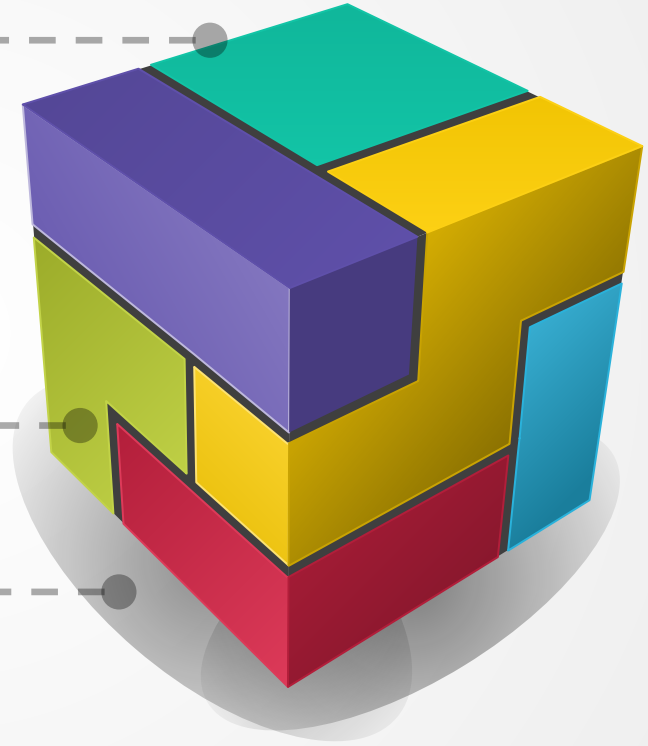


Diagrama de Clases



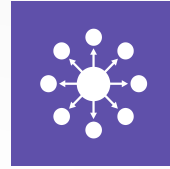
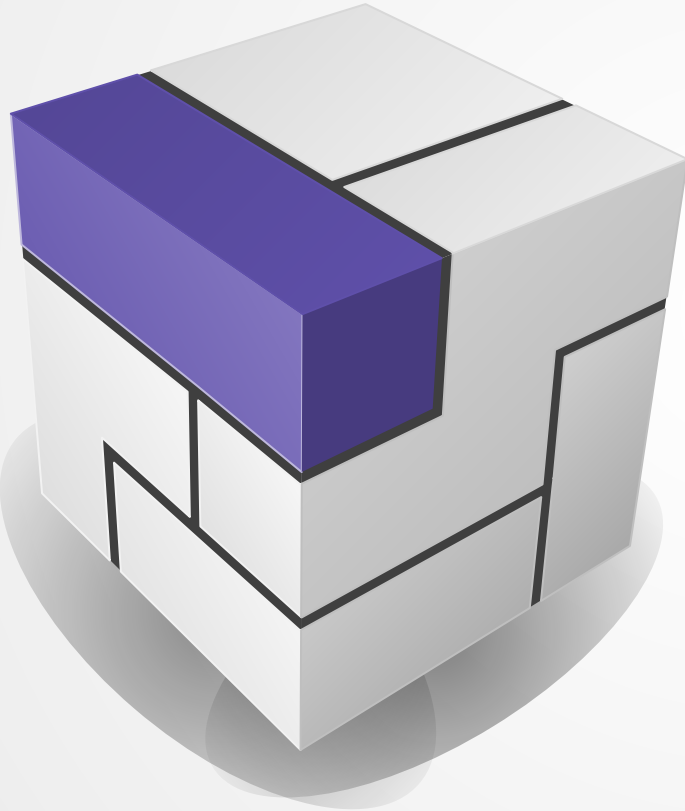


Diagrama de clases

Los diagramas de clases se utilizan para modelar la vista de diseño estática de un sistema.

Términos

01

Clase

Es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

02

Atributo

Es una propiedad de una clase identificada con un nombre, que describe un rango de valores que pueden tomar las instancias de la propiedad. Una clase puede tener cualquier número de atributos o no tener ninguno.

03

Operación

Es la implementación de un servicio que puede ser requerido a cualquier objeto de la clase para que muestre un comportamiento.

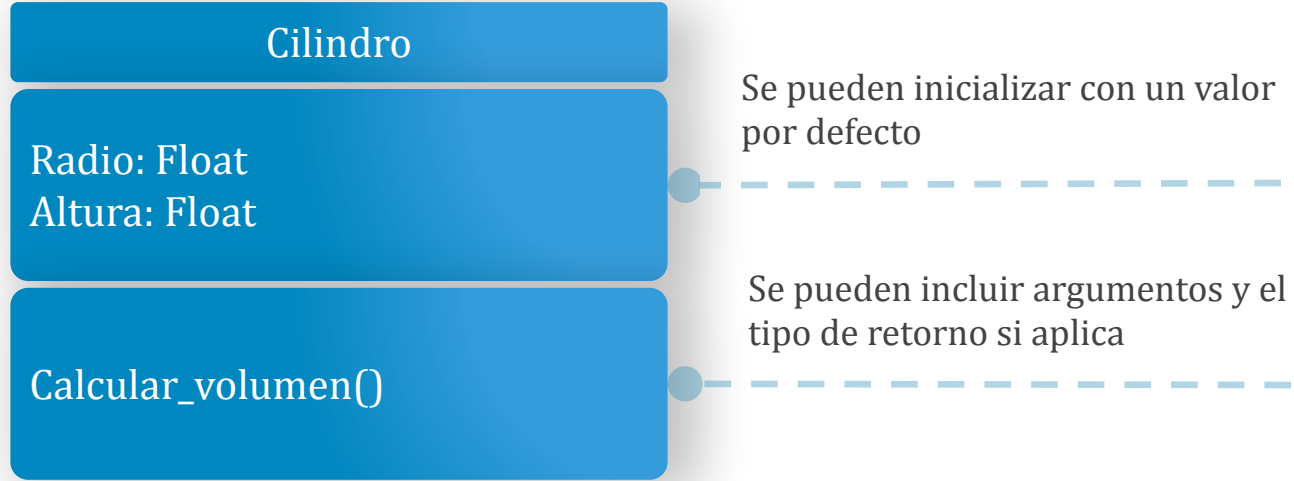
Representación Gráfica de una Clase

Nombre clase

Atributos

Operaciones

Abstrayendo el objeto “Cilindro” por medio de una Clase



Relaciones entre Clases

1

Dependencia

2

Asociación

3

Generalización

4

Agregación

5

Composición

01

Una dependencia es una relación de uso que declara que un elemento utiliza la información y los servicios de otro elemento, pero no necesariamente a la inversa.

1

Dependencia

2

Asociación

3

Generalización

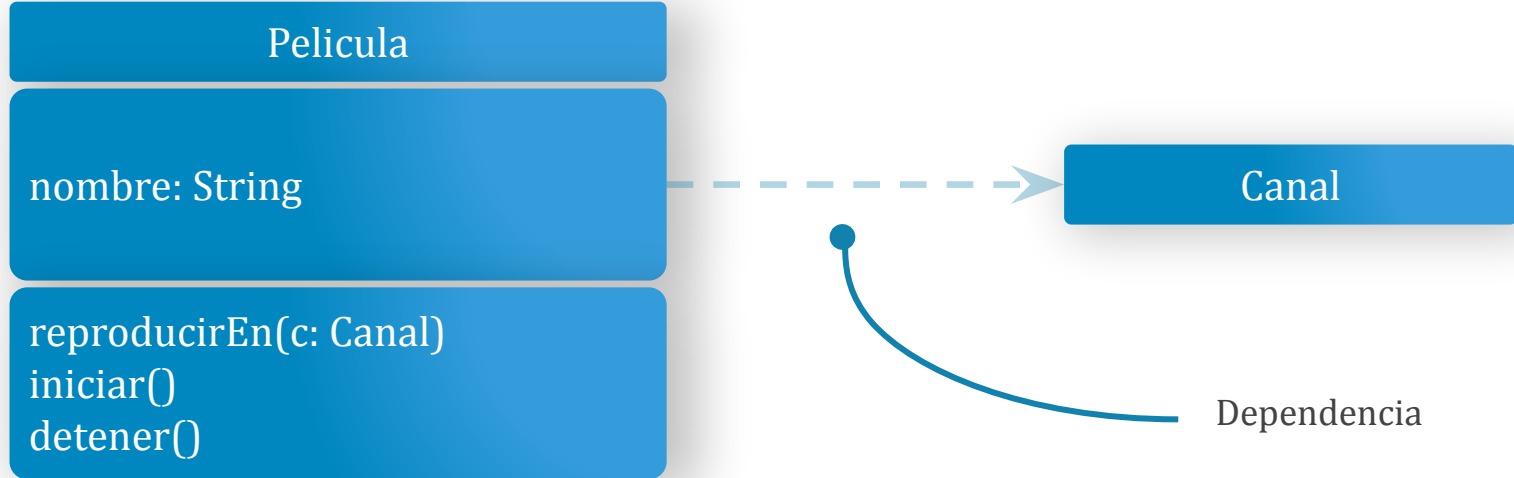
4

Agregación

5

Composición

Gráficamente, una dependencia se representa como una línea discontinua dirigida hacia el elemento del cual se depende. Las dependencias se usarán cuando se quiera indicar que un elemento utiliza a otro.



02

Es una relación estructural que especifica que los objetos de un elemento están conectados con los objetos de otro.

1

Dependencia

2

Asociación

3

Generalización

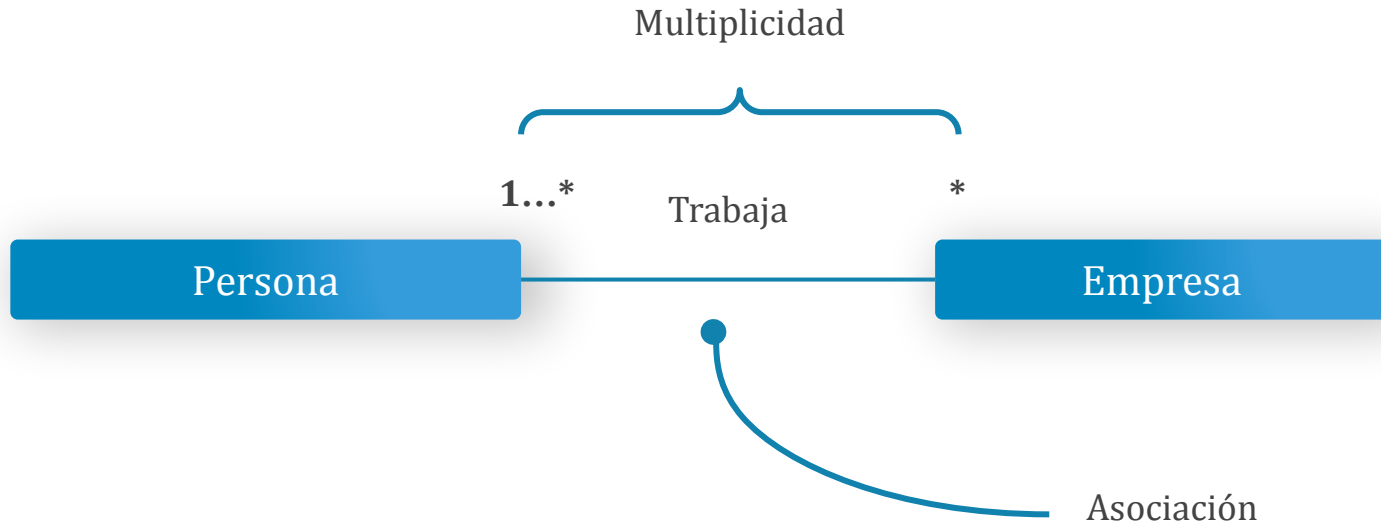
4

Agregación

5

Composición

Gráficamente, una asociación se representa como una línea continua que conecta la misma o diferentes clases. Las asociaciones se utilizarán cuando se quiera representar relaciones estructurales.



03

Es una relación entre un elemento general (llamado superclase o padre) y un caso más específico de ese elemento (llamado subclase o hijo).

1

Dependencia

2

Asociación

3

Generalización

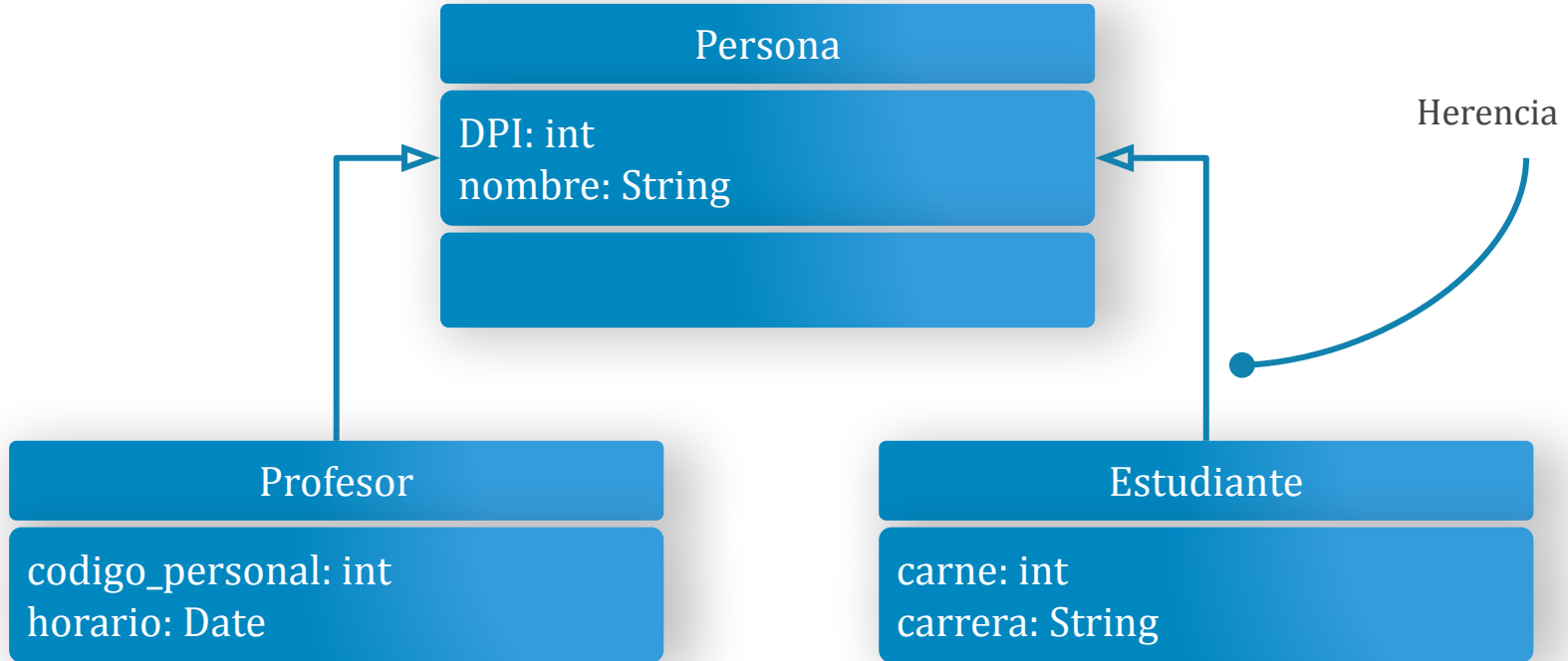
4

Agregación

5

Composición

Un hijo hereda las propiedades de sus padres, especialmente sus atributos y operaciones. Gráficamente, la generalización se representa como una línea dirigida continua, con una gran punta de flecha vacía, apuntando al padre, como se muestra:



04

Representa una relación estructural entre iguales, es decir, ambas clases están conceptualmente en el mismo nivel. A veces, se desea modelar una relación “todo/parte”, en la cual una clase representa una cosa grande (el “todo”), que consta de elementos más pequeños (las “partes”).

1

Dependencia

2

Asociación

3

Generalización

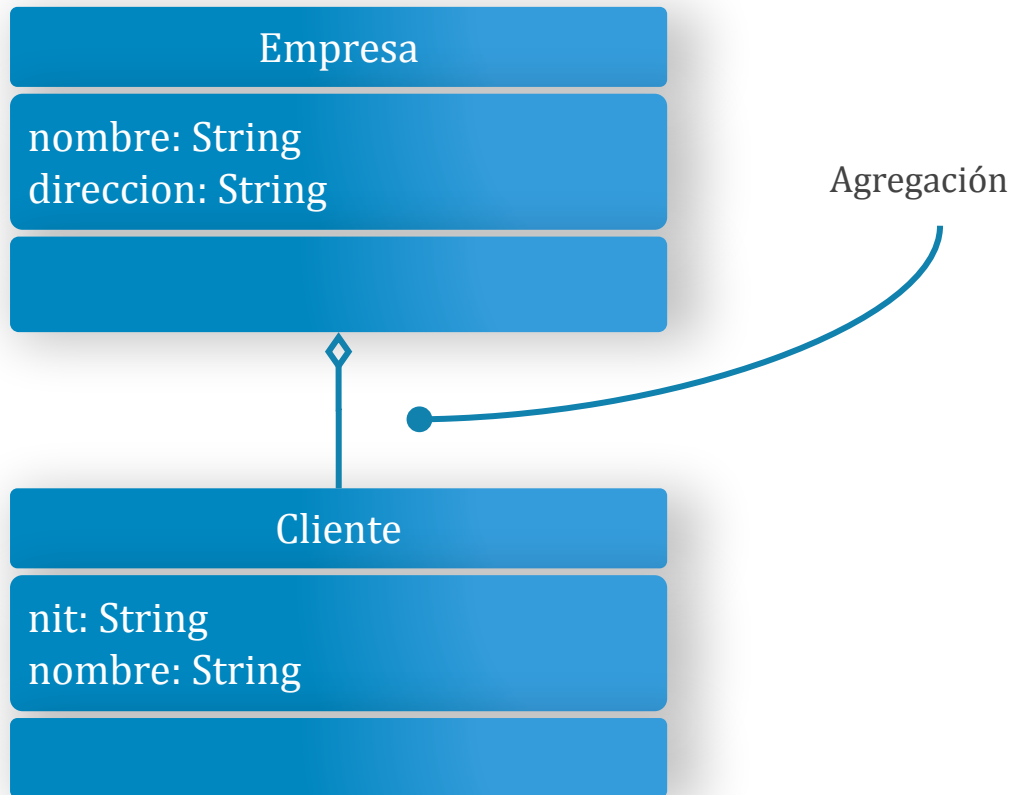
4

Agregación

5

Composición

En realidad, la agregación es sólo un tipo especial de asociación y se especifica añadiendo a una asociación normal un rombo vacío en la parte del todo, como se muestra:



05

La vida de la clase contenida debe coincidir con la vida de la clase contenedor. Los componentes constituyen una parte del objeto compuesto.

1

Dependencia

2

Asociación

3

Generalización

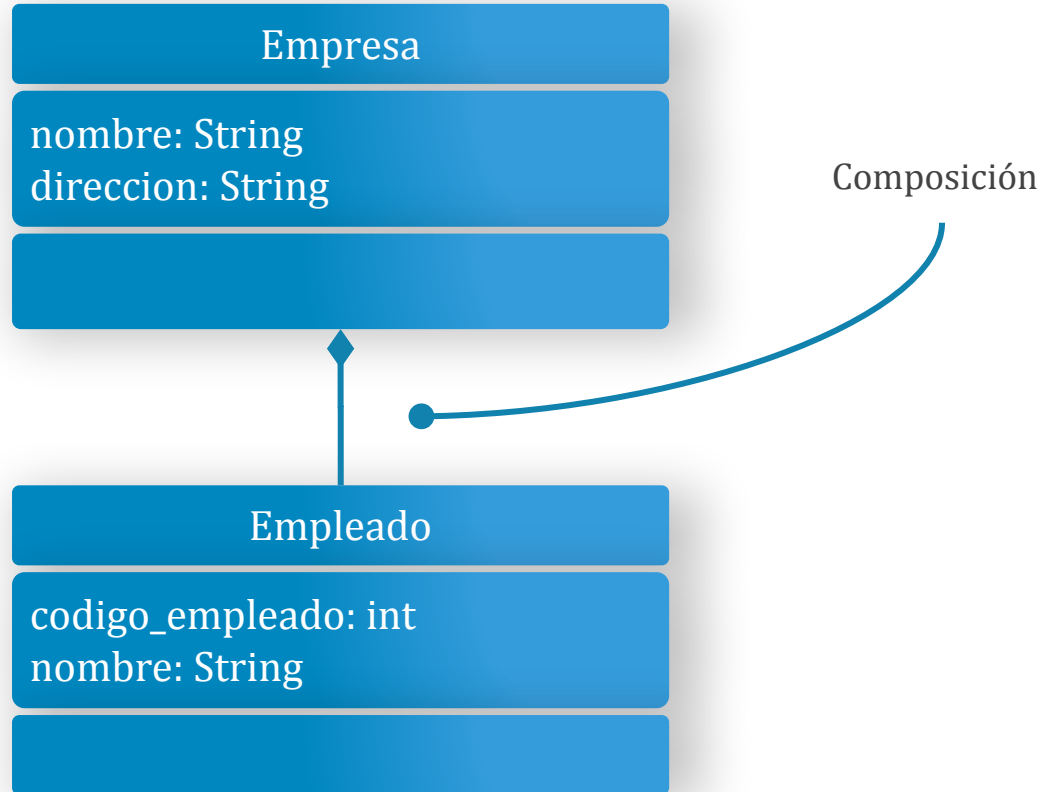
4

Agregación

5

Composición

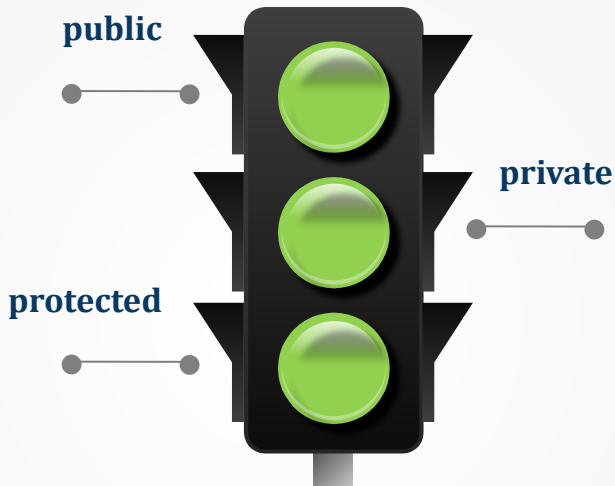
Gráficamente la composición se representa por medio de una línea continua con un rombo lleno en la parte del todo, como se muestra:



Visibilidad

Cualquier clase con visibilidad hacia la clase dada puede utilizar la característica; se especifica precediéndola del símbolo +

Cualquier descendiente de la clase puede utilizar la característica; se especifica precediéndola del símbolo #



Sólo la propia clase puede utilizar la característica; se especifica precediéndola del símbolo -

Uno de los detalles de diseño que se puede especificar para un atributo y operación es su visibilidad.

Ejemplo:

Un centro de instalaciones deportivas quiere hacer una aplicación de reservas.

- En el centro existen instalaciones deportivas (piscinas, gimnasios, frontones, etc.).
- El centro en cuestión tiene socios, de los cuales se almacenan su dirección, correo electrónico, teléfono y nombre.
- Cada instalación es reservada por un socio en una fecha dada desde una hora de inicio hasta una hora de fin.
- Cada reserva puede tener asociada uno o varios artículos deportivos que se alquilan aparte (balones, redes, raquetas, etc.).
- Por ejemplo si se quiere hacer una reserva para jugar a voleibol se debe reservar un polideportivo (instalación), más la red y un balón (artículos).

Un centro de instalaciones deportivas quiere hacer una aplicación de reservas.

- En el **centro** existen **instalaciones** deportivas (piscinas, gimnasios, frontones, etc.).
- El **centro** en cuestión tiene **socios**, de los cuales se almacenan su **dirección, correo electrónico, teléfono y nombre**.
- Cada **instalación** es **reservada** por un **socio** en una **fecha** dada desde una **hora de inicio** hasta una **hora de fin**.
- Cada **reserva** puede tener asociada uno o varios **artículos** deportivos que se alquilan aparte (balones, redes, raquetas, etc.).
- Por ejemplo si se quiere hacer una reserva para jugar a voleibol se debe reservar un polideportivo (instalación), más la red y un balón (artículos).

- **Centro**

- ---

- **Instalación**

- nombre

- **Socio**

- nombre
- teléfono
- dirección
- correo

- **Artículo**

- nombre

- **Reserva**

- fecha
- hora_inicio
- hora_fin

Socio

nombre: String
telefono: int
direccion: String
correo: String

reservar ()

Instalacion

nombre: String

setNombre()
getNombre()

Reserva

fecha: Date
hora_inicio: Timestamp
hora_fin: Timestamp

cancelar_reserva ()

Centro

nombre: String

setNombre()
getNombre()

Articulo

nombre: String

setNombre()
getNombre()

