



Superclase

```
padding-top: 8px;";  
fo_bar">  
left;" for="keywords" class="field_information_container" style="width: 100%; height: 100%; border: 1px solid #ccc; border-radius: 4px; margin-bottom: 10px;">  
ount_info" class="field_information" style="width: 100%; height: 100%; border: 1px solid #ccc; border-radius: 4px; margin-bottom: 10px;">  
: -3px;"></a>  
og" class="field_information" style="width: 100%; height: 100%; border: 1px solid #ccc; border-radius: 4px; margin-bottom: 10px;">
```

```
: right; padding-top: 7px;";  
both;"></div>  
ords" class="tag-editor" style="width: 100%; height: 100%; border: 1px solid #ccc; border-radius: 4px; margin-bottom: 10px;">  
itor ui-sortable">  
h:1px"></li>  
ceholder">  
eywords or paste via Ctrl+V
```

```
keywords_for_clipboard" style="width: 100%; height: 100%; border: 1px solid #ccc; border-radius: 4px; margin-bottom: 10px;">  
n_keywords_container" style="width: 100%; height: 100%; border: 1px solid #ccc; border-radius: 4px; margin-bottom: 10px;">  
s-feedback has-clear" style="width: 100%; height: 100%; border: 1px solid #ccc; border-radius: 4px; margin-bottom: 10px;">  
div > div.keywords_info_bar
```



Superclase

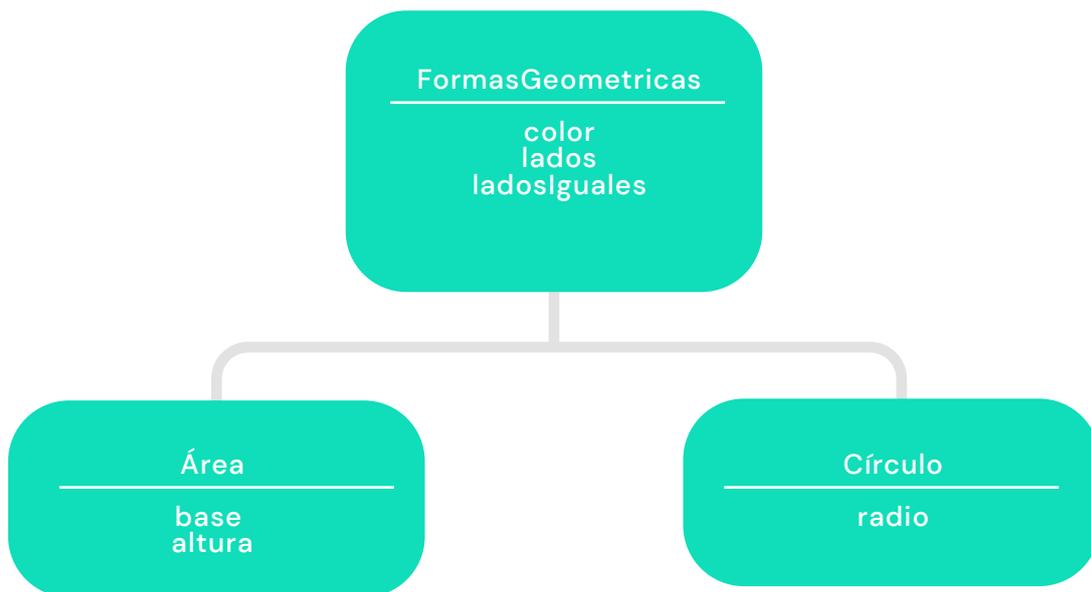
Superclases y subclases

Cuando modelamos una clase, sabemos que sus objetos tendrán sus propios atributos, pero habrá clases completamente diferentes que pueden compartir dichos atributos con otra clase, por lo que, en lugar de copiar atributos una y otra vez, podemos crear una clase con todo lo que tienen en común. Las demás clases heredan los atributos y se añaden las cosas particulares de cada una (Solvetic, 2019).

Lo anterior es lo que llamamos herencia. La superclase consiste en la clase que comparte todos los atributos y métodos y, por otro lado, las subclases son las que heredan de la primera; con esto creamos un sistema de jerarquía de clases, donde se bajan los niveles más altos de especificación y no hay restricciones. Podemos tener una clase que hereda de otra, la cual a su vez hereda de otra; así, se transfieren atributos comunes a una clase más baja que será muy específica, pero comparte funciones con otros objetos.

Ejemplo clásico

El ejemplo clásico usado para describir este comportamiento es la clase de formas geométricas, que se presenta a continuación:



Este caso se usa como estudio porque todas las formas tienen algunas propiedades comunes (por ejemplo, el color) y cada una tiene sus propias propiedades (cómo se calcula el área, si tiene lados iguales o diferentes o el radio en el caso de un círculo) (Solvetic, 2019).



Referencias bibliográficas

- Solvetic. (2019). *Java-Superclases y Subclases*. Recuperado de <https://bit.ly/3NLQkqM>