

Diferencias entre una clase Abstracta vs Interfaces

Clase Abstracta:

Se podrán definir métodos con implementación y sin implementación (abstract).

No se pueden usar instancias (crear objetos).

Se utilizará para definir subclases. Siempre será heredada para poder utilizar y sobrescribir los métodos de su clase padre.

La herencia de los atributos y métodos se hará de forma líneal. De una clase padre a una clase hija.

Se pueden ir heredando métodos abstractos y no abstractos.

Redefinir nuevas clases sin crear nuevos objetos.

Se piensa en una clase abstracta cuando el enfoque es más en **objetos** (de las subclases).

Se pueden encontrar clases como Figure que se pueden crear clases como Circle, Square, Triangle.

Interfaces:

Nuevos modificadores de acceso (default y private). Esto permite añadir comportamiento a los métodos.

Se podrán definir métodos con implementación y sin implementación.

Los métodos se pueden implementar en muchas familias de clases. Por lo cual no es líneal.

Se utiliza una interface cuando existen métodos que se pueden implementar en muchas familias (La relación va más allá entre dos clases).

Se piensa en una interface cuando el enfoque es más en las **acciones** que pueden tener en común muchos objetos.

Se encuentra acciones como Drawable Dibujable, Runnable Ejecutable, Callable Llamable, Visualizable Visualizable.

Una buena práctica es que el diseño de las aplicaciones siempre esté orientado a interfaces y no a la implementación. Crear buenas abstracciones. Encontrar el comportamiento común. Enfocarse en la declaración de los métodos.