

UNIVERSIDAD PRIVADA DE TACNA



INGENIERIA DE SISTEMAS

TITULO:

**TRABAJO DE INVESTIGACIÓN**

**CURSO:**

BASE DE DATOS II

**DOCENTE(ING):**

Patrick Cuadros Quiroga

Alumno:

Jhosmell Gyno Alfaro Musaja

(2015053223)

# Índice

1. Diferencia Clase Abstracta e interfaz	1
--	---

# Diferencia Clase Abstracta e interfaz

## 1. Diferencia Clase Abstracta e interfaz

Una clase abstracta puede heredar o extender cualquier clase (independientemente de que esta sea abstracta o no), mientras que una interfaz solamente puede extender o implementar otras interfaces.

Una clase abstracta puede heredar de una sola clase (abstracta o no) mientras que una interfaz puede extender varias interfaces de una misma vez.

Una clase abstracta puede tener métodos que sean abstractos o que no lo sean, mientras que las interfaces sólo y exclusivamente pueden definir métodos abstractos.

En java concretamente (ya que has puesto la etiqueta Java), en las clases abstractas la palabra `abstract` es obligatoria para definir un método abstracto (así como la clase). Cuando defines una interfaz, esta palabra es opcional ya que se infiere en el concepto de interfaz.

En una clase abstracta, los métodos abstractos pueden ser `public` o `protected`. En una interfaz solamente puede haber métodos públicos. En una clase abstracta pueden existir variables `static`, `final` o `static final` con cualquier modificador de acceso (`public`, `private`, `protected` o `default`). En una interfaz sólo puedes tener constantes (`public static final`).

- En definitiva utilizar una u otra depende de tus necesidades. Yo en todos los años que llevo programando en Java sobre todo, apenas he utilizado las clases abstractas más que una o dos veces. En cambio las interfaces se utilizan mucho cuando trabajas con interfaces gráficas por ejemplo. Uno de los usos más comunes es para crear aplicaciones que hagan uso del concepto de hebras para ejecutar procesos que consumen más tiempo de CPU. Esto se hace mediante la interfaz `Runnable` de Java, por poner uno de los innumerables ejemplos para los que se utilizan las interfaces.

En definitiva, es decisión tuya utilizar una clase abstracta o una interfaz. Si necesitas programar una clase de la que vayan a heredar otras pero que esas otras clases que heredan, compartan alguna funcionalidad (ejemplo: clase abstracta `-¿Persona`. Clases que heredan `-¿Alumno`, `Profesor...` Todas pueden tener un atributo `nombre` y métodos `get` y `set` de este atributo que hagan lo mismo), en ese caso te recomiendo que utilices una clase abstracta. En caso contrario, interfaz.