**Diferencia entre interfaz y clase abstracta**

Existen varias diferencias entre una clase abstracta y una interfaz:

1. 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.
2. Una clase abstracta puede heredar de una sola clase (abstracta o no) mientras que una interfaz puede extender varias interfaces de una misma vez.
3. 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.
4. 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.
5. En una clase abstracta, los métodos abstractos pueden ser public o protected. En una interfaz solamente puede haber métodos públicos.
6. 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).

**– Herencia múltiple:**

Podemos lograr múltiples herencias usando interfaces. Como ABAP no admite más de una clase Padre (Clase Superior), sólo puede tener una clase abstracta como clase Padre.

**– Nueva funcionalidad:**

Si añadimos un nuevo método en la interfaz, todas las clases que han implementado la interfaz tienen que implementar este método.

Si no implementamos el método, va a resultar un error en tiempo de ejecución RUN – Time error.

Para la clase abstracta, si añadimos un método no abstracto, no es necesario volver a implementar el método en cada clase heredada.

**– Comportamiento por defecto:**

Podemos tener un comportamiento predeterminado de un método en la clase abstracta. No podemos tener ningún comportamiento predeterminado en la interfaz, ya que sólo contiene la firma de los métodos.

**– Visibilidad:**

Todos los componentes de la interfaz son públicos por defecto.

Para la clase abstracta, podemos establecer la visibilidad de cada componente.

**Referencia:**

* ¿Cuál es la diferencia entre Interfaz y Clase Abstracta en java? (2017, 6 octubre). Recuperado 11 febrero, 2020, de <https://es.stackoverflow.com/questions/107742/cu%C3%A1l-es-la-diferencia-entre-interfaz-y-clase-abstracta-en-java>
* Diferencias entre interfaz y clases abstractas - Logali Training. (2018, 29 mayo). Recuperado 11 febrero, 2020, de <https://training.logaligroup.com/diferencias-entre-interfaz-y-clases-abstractas/>