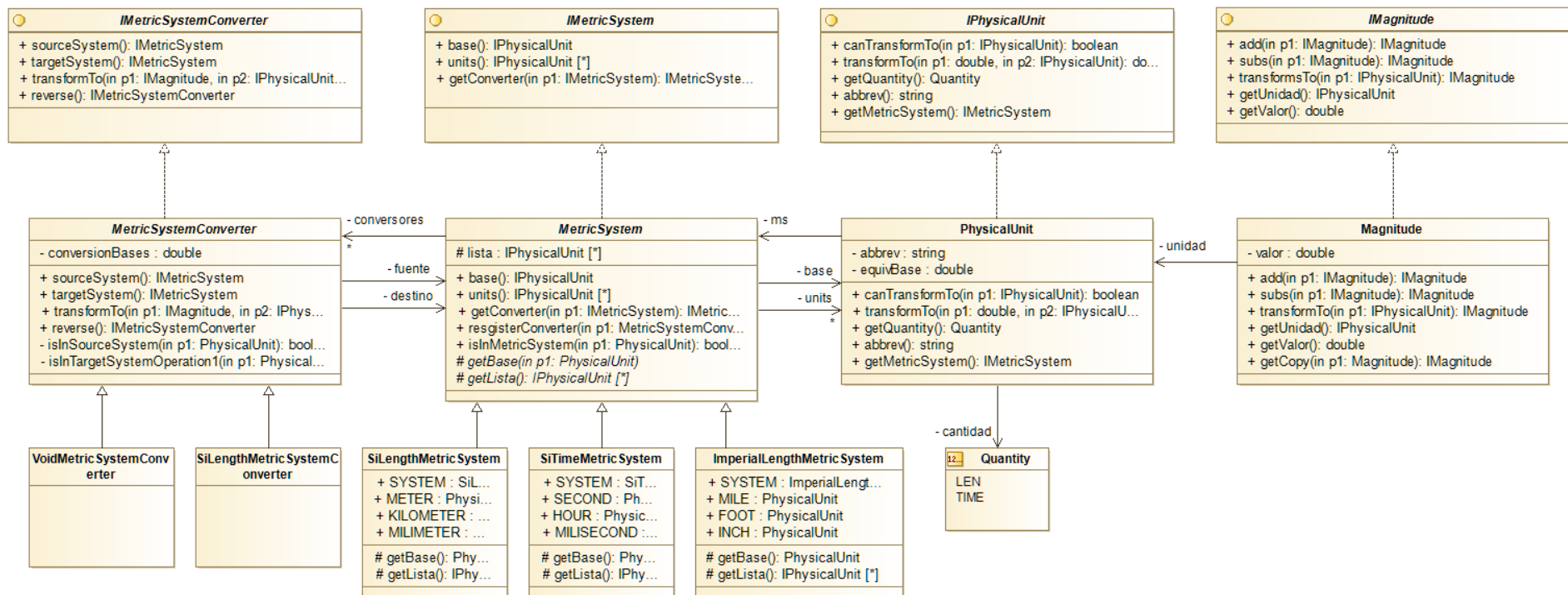


APARTADO 5

a.- Diagrama de clases



En la imagen de la página anterior presentamos nuestro diagrama de clases, el cual fuimos desarrollando de forma simultánea a la implementación en java para tener una idea más clara del funcionamiento que debía presentar nuestro sistema. El diagrama muestra claramente la funcionalidad del programa, aun lo así explicamos a grandes rasgos: las cuatro clases superiores son las interfaces de las que disponemos, en la fila intermedia debajo de cada interfaz nos encontramos con las clases que las implementan y finalmente en la sección inferior tenemos algunas subclases, que son casos específicos de MetricSystemConverter y MetricSystem, además de la enumeración Quantity. Las relaciones existentes entre las clases que implementan las interfaces se muestran claramente como arrays (convertidores y units) o elementos únicos mediante las flechas de asociación existentes entre ellas.

b.- Extensibilidad de diseño

Con el fin de añadir nuevas unidades a nuestro sistema simplemente habría que tomar el sistema métrico al que pertenecen o crearlo en caso de que no exista y crear en él una variable de tipo static final PhysicalUnit con las características propias de la unidad a añadir. Adicionalmente sería necesario incluir esta nueva unidad en el método getList() perteneciente a su sistema métrico.

Por otro lado, si quisiéramos añadir una nueva cantidad como es masa deberíamos añadir un campo MASA a la enumeración Quantity y luego crear adecuadamente una clase SiMasaMetricSystem que herede de la superclase MetricSystem. Para añadir unidades de esta cantidad simplemente habría que seguir los pasos descritos en el párrafo anterior.

c.- Ventajas y limitaciones

Consideramos que nuestro diseño es suficientemente robusto para las funcionalidades exigidas en el enunciado, pero presenta bastante generalidad por lo que no solo se limita a ellas y además, como se ha visto en el punto anterior, es fácilmente extensible. Cabe mencionar que hemos ampliado alguno de los testers proporcionados para asegurarnos que el programa funcione no solo en un caso concreto de prueba sino en todos. En contra, algunas limitaciones que presenta es la incapacidad de realizar operaciones complejas más allá de la suma y la resta; así como el manejo de unidades simples únicamente (se excluye la posibilidad de usar unidades compuestas que aparece en el apartado opcional). Otra limitación importante es la relativa a la precisión de las operaciones, ya que esta se ve acotada por el almacenamiento de valores en las variable de tipo double que java presenta; esta resulta más que suficiente para la mayoría de casos pero en alguna aplicación científica podría resultar determinante.