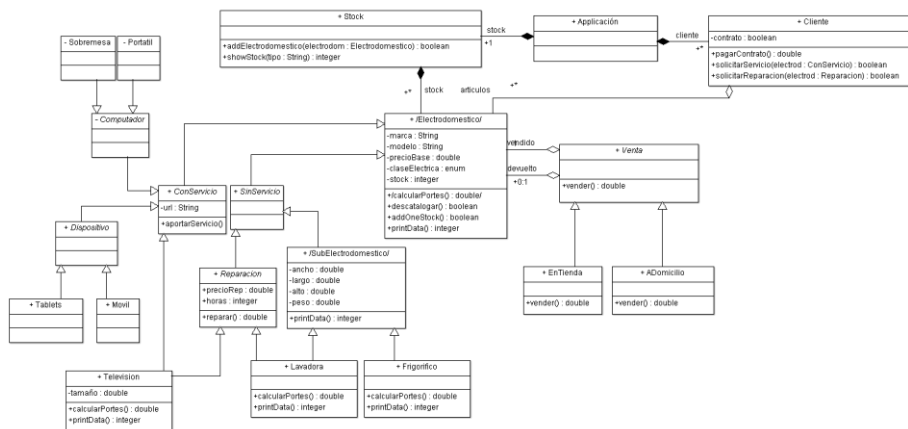


## Apartado 4



Para la realización de este apartado, hemos tenido que añadir bastantes cosas al diagrama.

Para empezar, lo más sencillo de todo, es una clase Aplicación, constituida por un Stock y una serie de Clientes. Por su parte, los clientes tienen un atributo contrato, que indicaría si tienen o no tienen contrato. La clase Cliente contiene una serie de Electrodomésticos, que representarían los Electrodomésticos que dicho cliente posee. Con ellos puede llamar a los métodos solicitarServicio y solicitarReparación. Por otra parte, el método pagarContrato devolverá el importe que debe pagar dependiendo de si tiene o no contrato.

Por la izquierda del diagrama, hemos añadido dos clases hijas: ConServicio y SinServicio, que representan los Electrodomésticos que tendrán o no acceso a Servicios técnicos. Por una parte, de la clase ConServicio heredan las clases Computador (que tanto ella como sus hijas serán privadas, pero se representan ahí por la posibilidad de necesitar implementarlas en el futuro) y Dispositivo, que a su vez tiene dos subclases Tablets y Movil. El método aportarServicio devolverá la url almacenada en la clase.

Finalmente, hemos representado un par de herencias múltiples (Para Televisión y para Lavadora) porque así lo requería el enunciado. Siento Televisión clase hija de ConServicio y Reparación y siendo Lavadora clase hija de Reparación y SubElectrodomestico. En java, este diagrama no se podría implementar pues no existe la herencia múltiple, sin embargo, podría emularse el funcionamiento solicitado en el enunciado eliminando las clases ConServicio, SinServicio y Reparación y mediante la repetición de algunos campos y métodos: Lavadora y televisión tendrían campos iguales, además de la implementación de los métodos relativos al servicio técnico y a las reparaciones (mientras que en cualquier otra subclase de Electrodoméstico estos métodos simplemente devolverían un error de alguna forma por no poder realizarse esa acción).