



## Consignas

### Proyecto

Deberá generar un proyecto en Java, con JUnit 4 y las siguientes características:

- Nombre: PB220252CTM-Recuperatorio-**ApellidosNombres** debiendo reemplazar “ApellidosNombres” por los suyos
- Se considerará desaprobado si el proyecto no compila.

### Entrega

Se deberá comprimir el proyecto y entregarlo en la plataforma MieL, en la práctica destinada para tal fin.

### Enunciado

Se necesita desarrollar un sistema que permita administrar la venta de instrumentos musicales. Dichos instrumentos poseen una marca, modelo, número de serie y precio base. Por el momento solo contaremos con guitarras, bajos y baterías.

Para cada instrumento se debe permitir calcular su precio final de venta según los siguientes criterios:

- Guitarras: precio base + 50% de su precio base.
- Bajos: precio base + \$20000 + 10% de su precio base.
- Baterías: precio base + 30% de su precio base.

Las marcas de los instrumentos podrán ser por ejemplo Fender o Yamaha. Los números de serie deben poseer 8 dígitos y deben ser únicos sin importar el grupo al que pertenezca cada instrumento.

Para las ventas, es necesario registrar un identificador incremental de 1 en 1 autogenerado, los instrumentos incluidos en la misma y el monto final de la venta. Para el cálculo del monto final, se deberá contemplar le precio final de cada instrumento en la venta.

No se debe permitir el registro de instrumentos musicales con un valor base menor o igual a cero.



Tampoco se permite agregar el mismo instrumento musical, considerando el número de serie, a la misma venta.

Todos los instrumentos deben estar siempre ordenados por defecto, por precio final.

## Requisitos funcionales

- Registrar instrumentos musicales para poder vender.
- Registrar ventas de los instrumentos musicales.
- Obtener el precio final de instrumentos.
- Generar un reporte de todas ventas realizadas que superen un monto determinado.

## Requisitos técnicos

- Desarrollar el sistema aplicando TDD. Las verificaciones deben alinearse con lo solicitado en cada prueba.
- Aplicar herencia y polimorfismo. Evalúe el uso de clases abstractas y/o interfaces según considere necesario.
- Usar colecciones adecuadas o mapas, según lo antes descrito.
- Aplicar las restricciones necesarias usando excepciones.

## Pruebas a desarrollar

1. deberiaPermitirseElRegistroDeUnaGuitarra.
2. noDeberiaPermitirseElRegistroDeUnBajoConPrecioBaseInvalido.
3. noDeberiaPermitirseElRegistroDeUnaBateriaEnUnaVentaConElMismoNumeroDeSerie.
4. deberiaPermitirseRegistrarUnaVentaConUnaGuitarraUnaBateriaYUnBajo.
5. deberiaPoderCalcularseElPrecioFinalDeUnaBateria.
6. deberiaPoderCalcularseElPrecioFinalDeUnaVentaConUnaGuitarraYUnBajo.
7. DeberiaPoderObtenerseUnaColeccionDeTodosLosInstrumentosOrdenadosPorPrecioFinalDescendente.
8. deberiaPoderObtenerseUnaColeccionDeBateriasOrdenadasPorPrecioFinalAscendente.
9. DeberiaPoderObtenerseUnReporteDeTodasLasVentasQueIncluyanBajosDondeLaClaveSeaLaVentaYEElValorSeaElPrecioFinal.