

**Entornos de Desarrollo (ED) Unidad de Trabajo 9 Tarea 04****OBJETIVOS DE APRENDIZAJE**

- Diferenciar las técnicas de diseño de pruebas.
- Utilizar técnicas de diseño de pruebas.
- Escribir casos de prueba.

**TA04**

---

**DISEÑO DE PRUEBAS UTILIZANDO TÉCNICAS DE CAJA NEGRA****Clases de equivalencia****Rúbrica y guía de calificación:**

- ☒ Utiliza este documento para responder.
- ☒ Determina razonadamente las clases de equivalencia apropiadas. **total 3 puntos**
- ☒ Crea la tabla formalmente correcta con los datos correctos resultantes del análisis anterior. **total 6 puntos**

Se te ha pedido que implementes las pruebas para una clase denominada *SalesList* (*lista de ventas*), cuyo atributo es **List<Venta>**. Cada venta tiene una *descripción: String*, un *importe: double*, una *cantidad: int* y un *precio: double*.

La clase tiene los siguientes métodos:

```
//1.  
addSale(String concepto, int cantidad, double precio)
```

Añade un nuevo elemento a la **Lista** con el concepto especificado como descripción del producto. El **importe** se calculará multiplicando la **cantidad** y el **precio**. Devolverá 0 si todo está bien y -1 si hay algún error. **No** podremos sumar ventas con **importe = 0** o **precio < 0**, pero sí podemos añadir ventas con importes negativos (**pero no precios negativos**).

```
//2.  
getTotal()
```

Devolverá la suma total de los ingresos de la **Lista**.

```
//3  
getAverage()
```

Devolverá el promedio de ingresos.

Diseña los posibles casos de prueba para cada método de la clase.

- Con respecto al método ***addSale()***, solo tienes que completar la tabla de casos de prueba.
- Para los métodos ***getTotal()*** y ***getAverage()***, solo necesitas establecer las condiciones previas para obtener el resultado deseado, ya que no tienen parámetros.

Inserta aquí la tabla:

---

## Valores límite

### Rúbrica y guía de calificación:

- ☒ Determina razonadamente los valores límite.
- ☒ Crea la tabla formalmente correcta con los datos correctos resultantes del análisis anterior. **total 9 puntos**

Vamos a probar una función que tiene como entrada el día de un mes (entero entre 1 y 31) y un número de mes (entero entre 1 y 12) y devuelve cuántos días quedan en este mes (un entero entre 1 y 30, dependiendo del mes).

```
int getDaysLeft(int dayOfMonth, int monthNumber)
{
    ...
}
```

Escribe la tabla con los posibles casos de prueba para cubrir todos los valores límite.