

## AeroDescuentos

Se está desarrollando para una aerolínea su módulo de liquidación de tiquetes aéreos. Para el mismo, se tiene una función que aplica descuentos a la tarifa base del vuelo dependiendo del tiempo de antelación de la reserva y la edad del pasajero. Los descuentos SON ACUMULABLES.

*Normativa 005, sobre los descuentos:*

- **15%** de descuento sacando el billete con antelación superior a 20 días.
- **5%** a los pasajeros con edad inferior a 18 años y **8%** a los pasajeros con edad superior a 65 años.

La siguiente es la especificación de la función que se usará en el módulo del cálculo de los descuentos:

```
/**
 * calcular la tarifa de cada billete según el trayecto, la antelación
 * en la que se obtiene el billete y la edad del pasajero, de acuerdo
 * con la normativa 005.
 * @param tarifaBase valor base del vuelo
 * @param diasAntelacion dias de antelación del vuelo
 * @param edad - edad del pasajero
 * @throws ExcepcionParametrosInvalidos Si alguno de los parámetros es inválido:
 *   - Si la tarifaBase es menor o igual que cero.
 *   - Si los días de antelación son negativos.
 *   - Si la edad es negativa o mayor que 120 años.
 */
public long calculoTarifa(long tarifaBase, int diasAntelacion, int edad)
```

1. De acuerdo con lo indicado, y teniendo en cuenta que NO hay precondiciones, en qué casos se debería arrojar una excepción de tipo `ExcepcionParametrosInvalidos`?. Agregue esto a la especificación.

- Si la `tarifaBase` es menor o igual que cero.
- Si los días de antelación son negativos.
- Si la edad es negativa o mayor que 120 años.

2. En la siguiente tabla enumere un conjunto de clases de equivalencia que -según usted- creen una buena división del conjunto de datos de entrada de la función anterior:

Número	Clase de equivalencia (en lenguaje natural o matemático).	Resultado correcto / incorrecto.
1.	<code>tarifaBase</code> es mayor que cero	Correcto
2.	<code>tarifaBase</code> es igual a cero	Incorrecto
3.	<code>tarifaBase</code> es menor a cero	Incorrecto
4.	<code>diasAntelación</code> es igual o mayor a cero	Correcto
5.	<code>diasAntelación</code> es menor a cero	Incorrecto
6.	edad es mayor o igual a cero y menor a 120	Correcto
7.	edad es menor a cero	Incorrecto
8.	edad es mayor a 120	Incorrecto

3. Para cada clase de equivalencia, defina un caso de prueba específico, definiendo: parámetros de entrada y resultados esperados.

1.	tarifaBase = 100, diasAntelacion = 25, edad = 30	Tarifa calculada correctamente con descuentos aplicados.
2	tarifaBase = 0, diasAntelacion = 10, edad = 50	ExcepcionParametrosInvalidos arrojada debido a que tarifaBase es igual a cero.
3	tarifaBase = -50, diasAntelacion = 15, edad = 40	ExcepcionParametrosInvalidos arrojada debido a que tarifaBase es negativo.
4	tarifaBase = 200, diasAntelacion = 30, edad = 25	Tarifa calculada correctamente con descuentos aplicados.
5	tarifaBase = 150, diasAntelacion = -5, edad = 35	ExcepcionParametrosInvalidos arrojada debido a que diasAntelacion es negativo.
6	tarifaBase = 300, diasAntelacion = 20, edad = 18	Tarifa calculada correctamente con descuentos aplicados.
7	tarifaBase = 180, diasAntelacion = 15, edad = -10	ExcepcionParametrosInvalidos arrojada debido a que edad es negativa.
8	tarifaBase = 250, diasAntelacion = 25, edad = 130	ExcepcionParametrosInvalidos arrojada debido a que edad es mayor que 120 años.

4. A partir de las clases de equivalencia identificadas en el punto 2, identifique las condiciones límite o de frontera de las mismas.

1.  
Límite Inferior: tarifaBase = 1 (valor mínimo positivo).  
Límite Superior: No hay.
2.  
Límite Inferior: tarifaBase = 0.  
Límite Superior: No hay.
3.  
Límite Inferior: tarifaBase = -1 (valor mínimo negativo).  
Límite Superior: No hay.
4.  
Límite Inferior: diasAntelacion = 0.  
Límite Superior: No hay.
5.  
Límite Inferior: diasAntelacion = -1 (valor mínimo negativo).  
Límite Superior: No hay.
6.  
Límite Inferior: edad = 0.  
Límite Superior: edad = 120.
7.  
Límite Inferior: edad = -1 (valor mínimo negativo).  
Límite Superior: No hay.
8.  
Límite Inferior: No aplica.  
Límite Superior: edad = 121 (valor mínimo mayor que 120).

5. Para cada una de las condiciones de frontera anteriores, defina casos de prueba específicos.

1	tarifaBase = 1, diasAntelacion = 25, edad = 30	Tarifa calculada correctamente con descuentos aplicados.
2	tarifaBase = 0, diasAntelacion = 10, edad = 50	ExcepcionParametrosInvalidos arrojada debido a que tarifaBase es igual a cero.
3	tarifaBase = -1, diasAntelacion = 15, edad = 40	ExcepcionParametrosInvalidos arrojada debido a que tarifaBase es negativa.
4	tarifaBase = 200, diasAntelacion = 0, edad = 25	Tarifa calculada correctamente con descuentos aplicados.
5	tarifaBase = 150, diasAntelacion = -1, edad = 35	ExcepcionParametrosInvalidos arrojada debido a que diasAntelacion es negativo.
6	tarifaBase = 300, diasAntelacion = 20, edad = 0	Tarifa calculada correctamente con descuentos aplicados.
7	tarifaBase = 180, diasAntelacion = 15, edad = -1	ExcepcionParametrosInvalidos arrojada debido a que edad es negativa.
8	tarifaBase = 250, diasAntelacion = 25, edad = 121	ExcepcionParametrosInvalidos arrojada debido a que edad es mayor que 120 años.