

## Programación Orientada a Objetos. Práctica 2.3

### Características de la práctica

En esta práctica el alumno aprenderá cómo la *Herencia*, el *Polimorfismo* y la *Vinculación Dinámica* permiten reutilizar y redefinir el comportamiento de clases previamente definidas, y de cómo proporcionan un soporte adecuado para el *Principio de Sustitución*.

Además, también aprenderá la utilidad y necesidad de definir métodos protegidos que permitan a las clases derivadas manipular y redefinir el comportamiento de las clases base. Aprenderá también cómo un constructor de la clase derivada puede invocar al constructor de la clase base, y cómo un método de la clase derivada puede invocar al método de la clase base que está siendo redefinido.

Así mismo, también aprenderá cómo las *Interfaces* permiten especificar un interfaz operacional que podrá ser proporcionado por diversas instanciaciones de clases.

Así, se puede apreciar cómo la clase `LibroOferta` redefine el comportamiento definido por la clase `Libro` para modelar libros que ofrezcan una oferta de descuento en el precio final. Para ello, entre otras, redefine el método `getPrecioFinal` de la clase base.

La clase `LibreriaOferta` redefine el comportamiento de la clase `Libreria` de tal forma que permite crear tanto libros *normales*, como libros *en oferta*, y ambas clases de libros pueden almacenarse adecuadamente en la misma estructura de datos, gracias al polimorfismo y al *Principio de Sustitución*, considerando que cuando se invoque al método `getPrecioFinal` de un determinado libro, se invocará adecuadamente al método correspondiente al *tipo dinámico* del objeto, gracias a la vinculación dinámica.

Por otra parte, la clase `LibreriaOfertaFlex` también redefine el comportamiento de la clase `Libreria`, pero en este caso la selección de los libros *en oferta* se realiza de un modo más flexible, de tal forma que este mecanismo de selección se deja *abierto* para que pueda ser utilizado cualquier mecanismo que implemente la interfaz operacional especificada mediante la interfaz `OfertaFlex`. Nótese que cualquier clase, actual o diseñada en un futuro, que implemente dicha interfaz podrá ser utilizada para seleccionar los libros en oferta.

### Ejercicio 1. (proyecto prLibreria)

Este ejercicio pretende redefinir el comportamiento de las clases `Libro` y `Libreria` de la **práctica 2.2** para que permita tener *autores en oferta* en la librería, de tal forma que sus libros tengan precios en oferta.

La Figura 1 muestra el diagrama UML de las clases.

**Nota:** En la clase `Libreria`, se han cambiado a **protected** tanto la constante `CAP_INICIAL` como el método privado `anyadirLibro`. Si se considera necesario, también se pueden cambiar a **protected** otros métodos privados de las clases de la práctica 2.2, pero no así las *variables*.

**Nota:** se pueden añadir a las siguientes clases los métodos **privados** que se consideren necesarios.

#### La clase `LibroOferta`

La clase `LibroOferta` (del paquete `prLibreria`) deriva de la clase `Libro`, por lo que contiene información sobre un determinado libro, pero además, permite especificar un determinado porcentaje de descuento, que será aplicado al **precio base** al calcular el precio final del libro.

► `LibroOferta(String,String,double,double)`

Construye un objeto `LibroOferta`. Recibe como parámetros, en el siguiente orden, el nombre del autor, el título, el precio base y el porcentaje de descuento del libro.

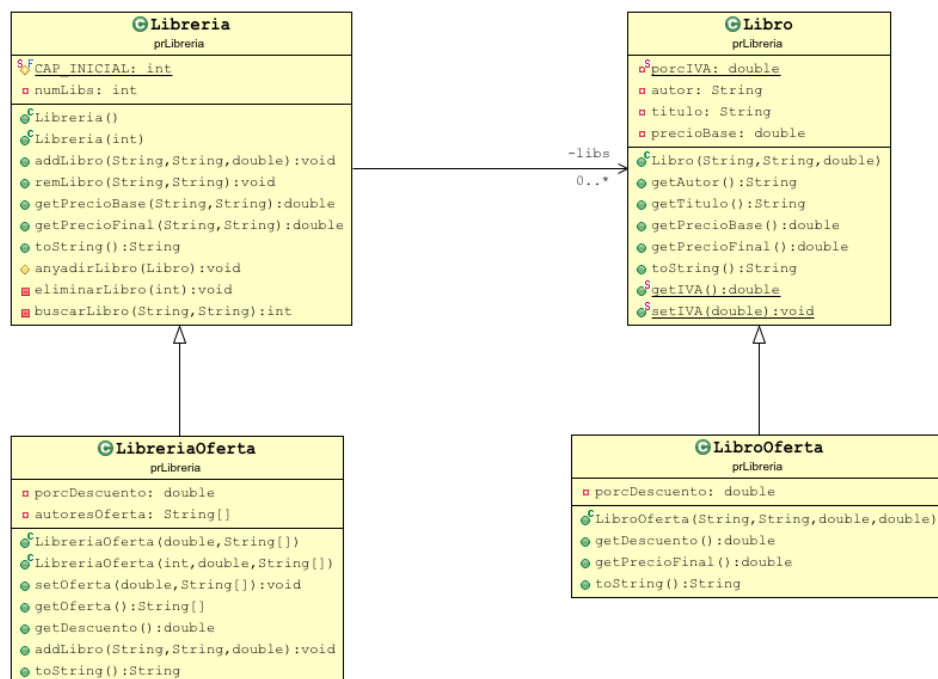


Figura 1: Diagrama de clases UML

► **getDescuento():double**

Devuelve el porcentaje de descuento del libro.

► **getPrecioFinal():double** // *[@Redefinición]*

Devuelve el precio final del libro, aplicando el descuento al precio base, e incluyendo el IVA, según las siguientes ecuaciones.

$$PX = PB - PB \times Descuento \div 100$$

$$PF = PX + PX \times IVA \div 100$$

► **toString(): String** // *[@Redefinición]*

Devuelve la representación textual del objeto, según el formato del siguiente ejemplo:

(Isaac Asimov; La Fundación; 7.30; 20.0%; 5.84; 10.0%; 6.424)

## La clase LibreriaOferta

La clase **LibreriaOferta** (del paquete **prLibreria**) deriva de la clase **Libreria**, pero permite crear y almacenar libros en oferta. Para ello, contiene además un array con los nombres de los autores en oferta, así como del porcentaje de oferta a aplicar a los libros de estos autores.

**Nota:** las comparaciones que se realicen tanto del nombre del autor como del título del libro se deberán realizar sin considerar el caso (*mayúsculas* o *minúsculas*) de las letras que lo componen.

**Nota:** se recomienda la definición de métodos privados y/o protegidos que simplifiquen y permitan modularizar la solución de métodos complejos.

► **LibreriaOferta(double,String[])**

Construye un objeto **LibreriaOferta** vacío (sin libros) con una librería base con una capacidad inicial por defecto. Además el porcentaje de descuento se recibe como primer parámetro, y el array con los nombres de los autores en oferta se recibe como segundo parámetro.

► `LibreriaOferta(int,double,String[])`

Construye un objeto `LibreriaOferta` vacío (sin libros) con una librería base con una capacidad inicial del tamaño recibido como primer parámetro. Además el porcentaje de descuento se recibe como segundo parámetro, y el array con los nombres de los autores en oferta se recibe como tercer parámetro.

► `setOferta(double,String[]):void`

Actualiza el valor del porcentaje de descuento y reemplaza el array de autores en oferta con los valores recibidos como parámetros.

► `getOferta():String[]`

Devuelve el array de autores en oferta.

► `getDescuento():double`

Devuelve el porcentaje de descuento.

► `addLibro(String,String,double):void` // *[@Redefinición]*

Si el nombre del autor recibido como primer parámetro es igual a algún autor de la lista de autores en oferta, entonces crea un nuevo objeto `LibroOferta` con el nombre del autor, el título, y el precio base recibidos como parámetros, y el descuento según el valor almacenado. En otro caso, crea un nuevo objeto `Libro` con el nombre del autor, el título, y el precio base recibidos como parámetros. En cualquier caso se invoca al método `anyadirLibro` para añadirlo a la librería.

Invoca a: `anyadirLibro` (se ha cambiado a *protegido* en la clase `Libreria`).

► `toString(): String` // *[@Redefinición]*

Devuelve la representación textual del objeto, según el formato del siguiente ejemplo (sin considerar los saltos de línea):

```
20.0%[George Orwell, Isaac Asimov]
[(George Orwell; 1984; 6.20; 20.0%; 4.96; 10.0%; 5.456),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.50; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.40; 20.0%; 7.52; 10.0%; 8.272),
 (Ray Bradbury; Fahrenheit 451; 7.40; 10.0%; 8.14),
 (Aldous Huxley; Un Mundo Feliz; 6.50; 10.0%; 7.15),
 (Isaac Asimov; La Fundación; 7.30; 20.0%; 5.84; 10.0%; 6.424),
 (William Gibson; Neuromante; 8.30; 10.0%; 9.13),
 (Isaac Asimov; Segunda Fundación; 8.10; 20.0%; 6.48; 10.0%; 7.128),
 (Isaac Newton; Arithmetica Universalis; 10.50; 10.0%; 11.55)]
```

## La aplicación `PruebaLibreriaOferta`

Desarrolle una aplicación (en el paquete anónimo) que permita realizar una prueba de las clases anteriores. Así, deberá crear un objeto `LibreriaOferta` con un 20.0% de descuento para los autores *George Orwell* e *Isaac Asimov*. Además, debe añadir a la librería los siguientes libros:

```
("george orwell", "1984", 8.20)
("Philip K. Dick", "¿Sueñan los androides con ovejas eléctricas?", 3.50)
("Isaac Asimov", "Fundación e Imperio", 9.40)
("Ray Bradbury", "Fahrenheit 451", 7.40)
("Aldous Huxley", "Un Mundo Feliz", 6.50)
("Isaac Asimov", "La Fundación", 7.30)
("William Gibson", "Neuromante", 8.30)
("Isaac Asimov", "Segunda Fundación", 8.10)
("Isaac Newton", "arithmetica universalis", 7.50)
("George Orwell", "1984", 6.20)
("Isaac Newton", "Arithmetica Universalis", 10.50)
```

De tal forma que al mostrar la representación textual de la librería mostrará (sin considerar los saltos de línea):

```
20.0%[George Orwell, Isaac Asimov]
[(George Orwell; 1984; 6.20; 20.0%; 4.96; 10.0%; 5.456),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.50; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.40; 20.0%; 7.52; 10.0%; 8.272),
 (Ray Bradbury; Fahrenheit 451; 7.40; 10.0%; 8.14),
 (Aldous Huxley; Un Mundo Feliz; 6.50; 10.0%; 7.15),
 (Isaac Asimov; La Fundación; 7.30; 20.0%; 5.84; 10.0%; 6.424),
 (William Gibson; Neuromante; 8.30; 10.0%; 9.13),
 (Isaac Asimov; Segunda Fundación; 8.10; 20.0%; 6.48; 10.0%; 7.128),
 (Isaac Newton; Arithmetica Universalis; 10.50; 10.0%; 11.55)]
```

A continuación se eliminarán los siguientes libros:

```
("George Orwell", "1984")
("Aldous Huxley", "Un Mundo Feliz")
("Isaac Newton", "Arithmetica Universalis")
("James Gosling", "The Java Language Specification")
```

De tal forma que al mostrar la representación textual de la librería mostrará (sin considerar los saltos de línea):

```
20.0%[George Orwell, Isaac Asimov]
[(William Gibson; Neuromante; 8.3; 10.0%; 9.13),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.5; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.4; 20.0%; 7.52; 10.0%; 8.272),
 (Ray Bradbury; Fahrenheit 451; 7.4; 10.0%; 8.14),
 (Isaac Asimov; Segunda Fundación; 8.1; 20.0%; 6.48; 10.0%; 7.128),
 (Isaac Asimov; La Fundación; 7.3; 20.0%; 5.84; 10.0%; 6.424)]
```

Finalmente se mostrará el precio final de los siguientes libros:

```
getPrecioFinal("George Orwell", "1984"): 0.0
getPrecioFinal("Philip K. Dick", "¿Sueñan los androides con ovejas eléctricas?"): 3.85
getPrecioFinal("isaac asimov", "fundación e imperio"): 8.272
getPrecioFinal("Ray Bradbury", "Fahrenheit 451"): 8.14
getPrecioFinal("Aldous Huxley", "Un Mundo Feliz"): 0.0
getPrecioFinal("Isaac Asimov", "La Fundación"): 6.424
getPrecioFinal("william gibson", "neuromante"): 9.13
getPrecioFinal("Isaac Asimov", "Segunda Fundación"): 7.128
getPrecioFinal("Isaac Newton", "Arithmetica Universalis"): 0.0
```

## Ejercicio 2. (proyecto prLibreria)

Este ejercicio pretende redefinir el comportamiento de las clases **Libro** y **Libreria** de la **práctica 2.2** para que permita especificar diferentes criterios de ofertas en la librería, de tal forma que sus libros tengan precios en oferta.

La Figura 2 muestra el diagrama UML de las clases.

**Nota:** En la clase **Libreria**, se han cambiado a **protected** tanto la constante **CAP\_INICIAL** como el método privado **anyadirLibro**. Si se considera necesario, también se pueden cambiar a **protected** otros métodos privados de las clases de la práctica 2.2, pero no así las *variables*.

**Nota:** se pueden añadir a las siguientes clases los métodos **privados** que se consideren necesarios.

### La *Interfaz* OfertaFlex

La *interfaz* **OfertaFlex** (del paquete **prLibreria**) especifica los métodos necesarios para calcular el porcentaje de descuento que se debe aplicar a un determinado libro.

► **getDescuento**(Libro): **double**

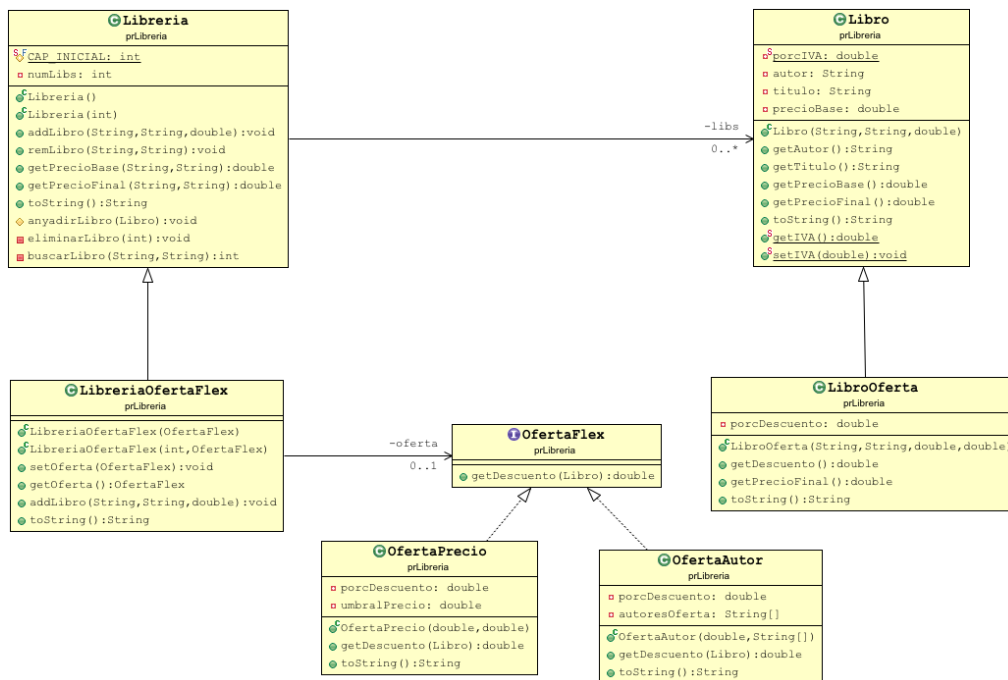


Figura 2: Diagrama de clases UML

Calcula y devuelve el porcentaje de descuento que se debe aplicar a un determinado libro recibido como parámetro. En caso de que no se deba aplicar ningún descuento, devolverá el valor cero.

### La clase `OfertaPrecio`

La clase `OfertaPrecio` (del paquete `prLibreria`) implementa la *interfaz* `OfertaFlex` y proporciona un método para calcular el porcentaje de descuento a partir de un determinado umbral en el precio base del libro, ambos especificados en la construcción del objeto.

#### ► `OfertaPrecio(double, double)`

Construye un objeto con el porcentaje de descuento y el umbral del precio, recibidos como parámetros en ese mismo orden.

#### ► `getDescuento(Libro): double` // *[@Redefinición]*

Calcula y devuelve el porcentaje de descuento que se debe aplicar a un determinado libro recibido como parámetro si el precio base del libro es mayor o igual al umbral especificado en la construcción del objeto. En caso de que no se deba aplicar ningún descuento, devolverá el valor cero.

#### ► `toString(): String` // *[@Redefinición]*

Devuelve la representación textual del objeto, según el formato del siguiente ejemplo:

20.0%(8)

### La clase `OfertaAutor`

La clase `OfertaAutor` (del paquete `prLibreria`) implementa la *interfaz* `OfertaFlex` y proporciona un método para calcular el porcentaje de descuento a partir de los nombres de autores en oferta, ambos especificados en la construcción del objeto.

**Nota:** las comparaciones que se realicen tanto del nombre del autor como del título del libro se deberán realizar sin considerar el caso (*mayúsculas* o *minúsculas*) de las letras que lo componen.

► `OfertaAutor(double,String[])`

Construye un objeto con el porcentaje de descuento y el array con los nombres de los autores en oferta, recibidos como parámetros en ese mismo orden.

► `getDescuento(Libro): double // [Redefinición]`

Calcula y devuelve el porcentaje de descuento que se debe aplicar a un determinado libro recibido como parámetro si el nombre del autor se encuentra en el array de autores en oferta especificado en la construcción del objeto. En caso de que no se deba aplicar ningún descuento, devolverá el valor cero.

► `toString(): String // [Redefinición]`

Devuelve la representación textual del objeto, según el formato del siguiente ejemplo:

20.0%[George Orwell, Isaac Asimov]

## La clase `LibreriaOfertaFlex`

La clase `LibreriaOfertaFlex` (del paquete `prLibreria`) deriva de la clase `Libreria`, pero permite crear y almacenar libros en oferta. Para ello, contiene una referencia a un objeto que implemente la interfaz `OfertaFlex`.

**Nota 1:** las comparaciones que se realicen tanto del nombre del autor como del título del libro se deberán realizar sin considerar el caso (*mayúsculas* o *minúsculas*) de las letras que lo componen.

**Nota 2:** se recomienda la definición de métodos privados y/o protegidos que simplifiquen y permitan modularizar la solución de métodos complejos.

► `LibreriaOfertaFlex(OfertaFlex)`

Construye un objeto `LibreriaOfertaFlex` vacío (sin libros) con una librería base con una capacidad inicial por defecto. Además almacena la referencia al objeto para calcular las ofertas de los libros, que se recibe como primer parámetro.

► `LibreriaOfertaFlex(int,OfertaFlex)`

Construye un objeto `LibreriaOfertaFlex` vacío (sin libros) con una librería base con una capacidad inicial del tamaño recibido como primer parámetro. Además almacena la referencia al objeto para calcular las ofertas de los libros, que se recibe como segundo parámetro.

► `setOferta(OfertaFlex):void`

Actualiza el valor del objeto para calcular ofertas de libros al objeto recibido como parámetro.

► `getOferta():OfertaFlex`

Devuelve el objeto para calcular ofertas de libros.

► `addLibro(String,String,double): void // [Redefinición]`

Crea un nuevo objeto `Libro` con el nombre del autor, el título, y el precio base recibidos como parámetros. Si el porcentaje de descuento calculado para este libro es distinto de cero, entonces vuelve a crear un nuevo objeto `LibroOferta` con el nombre del autor, el título, y el precio base recibidos como parámetros, y el descuento según el valor calculado por el objeto para el cálculo de las ofertas. En cualquier caso se invoca al método `anyadirLibro` para añadirlo a la librería.

Invoca a: `OfertaFlex.getDescuento` y `anyadirLibro` (se ha cambiado a *protegido* en la clase `Libreria`).

► `toString(): String` // *[@Redefinición]*

Devuelve la representación textual del objeto, según el formato del siguiente ejemplo (sin considerar los saltos de línea). En la primera línea aparecerá la información del tipo de oferta que se está realizando (en este caso se trata de una oferta del 20 % por autores). El resto de información mostrará los libros almacenados en la librería.

```
20.0%[George Orwell, Isaac Asimov]
[(George Orwell; 1984; 6.20; 20.0%; 4.96; 10.0%; 5.456),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.50; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.40; 20.0%; 7.52; 10.0%; 8.272),
 (Ray Bradbury; Fahrenheit 451; 7.40; 10.0%; 8.14),
 (Aldous Huxley; Un Mundo Feliz; 6.50; 10.0%; 7.15),
 (Isaac Asimov; La Fundación; 7.30; 20.0%; 5.84; 10.0%; 6.424),
 (William Gibson; Neuromante; 8.30; 10.0%; 9.13),
 (Isaac Asimov; Segunda Fundación; 8.10; 20.0%; 6.48; 10.0%; 7.128),
 (Isaac Newton; Arithmetica Universalis; 10.50; 10.0%; 11.55)]
```

## La aplicación PruebaLibreriaOfertaFlex

Desarrolle una aplicación (en el paquete anónimo) que permita realizar una prueba de las clases anteriores. Así, deberá crear un objeto `LibreriaOfertaFlex` con un objeto `OfertaAutor` que especifica un 20.0 % de descuento para los autores *George Orwell* e *Isaac Asimov*. Además, debe añadir a la librería los siguientes libros:

```
("george orwell", "1984", 8.20)
("Philip K. Dick", "¿Sueñan los androides con ovejas eléctricas?", 3.50)
("Isaac Asimov", "Fundación e Imperio", 9.40)
("Ray Bradbury", "Fahrenheit 451", 7.40)
("Aldous Huxley", "Un Mundo Feliz", 6.50)
("Isaac Asimov", "La Fundación", 7.30)
("William Gibson", "Neuromante", 8.30)
("Isaac Asimov", "Segunda Fundación", 8.10)
("Isaac Newton", "arithmetica universalis", 7.50)
("George Orwell", "1984", 6.20)
("Isaac Newton", "Arithmetica Universalis", 10.50)
```

De tal forma que al mostrar la representación textual de la librería mostrará (sin considerar los saltos de línea):

```
20.0%[George Orwell, Isaac Asimov]
[(George Orwell; 1984; 6.20; 20.0%; 4.96; 10.0%; 5.456),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.50; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.40; 20.0%; 7.52; 10.0%; 8.272),
 (Ray Bradbury; Fahrenheit 451; 7.40; 10.0%; 8.14),
 (Aldous Huxley; Un Mundo Feliz; 6.50; 10.0%; 7.15),
 (Isaac Asimov; La Fundación; 7.30; 20.0%; 5.84; 10.0%; 6.424),
 (William Gibson; Neuromante; 8.30; 10.0%; 9.13),
 (Isaac Asimov; Segunda Fundación; 8.10; 20.0%; 6.48; 10.0%; 7.128),
 (Isaac Newton; Arithmetica Universalis; 10.50; 10.0%; 11.55)]
```

A continuación se eliminarán los siguientes libros:

```
("George Orwell", "1984")
("Aldous Huxley", "Un Mundo Feliz")
("Isaac Newton", "Arithmetica Universalis")
("James Gosling", "The Java Language Specification")
```

De tal forma que al mostrar la representación textual de la librería mostrará (sin considerar los saltos de línea):

```
20.0%[George Orwell, Isaac Asimov]
[(William Gibson; Neuromante; 8.3; 10.0%; 9.13),
 (Philip K. Dick; ¿Sueñan los androides con ovejas eléctricas?; 3.5; 10.0%; 3.85),
 (Isaac Asimov; Fundación e Imperio; 9.4; 20.0%; 7.52; 10.0%; 8.272),
```

```
(Ray Bradbury; Fahrenheit 451; 7.4; 10.0%; 8.14),  
(Isaac Asimov; Segunda Fundación; 8.1; 20.0%; 6.48; 10.0%; 7.128),  
(Isaac Asimov; La Fundación; 7.3; 20.0%; 5.84; 10.0%; 6.424)]
```

Finalmente se mostrará el precio final de los siguientes libros:

```
getPrecioFinal("George Orwell", "1984"): 0.0  
getPrecioFinal("Philip K. Dick", "¿Sueñan los androides con ovejas eléctricas?"): 3.85  
getPrecioFinal("isaac asimov", "fundación e imperio"): 8.272  
getPrecioFinal("Ray Bradbury", "Fahrenheit 451"): 8.14  
getPrecioFinal("Aldous Huxley", "Un Mundo Feliz"): 0.0  
getPrecioFinal("Isaac Asimov", "La Fundación"): 6.424  
getPrecioFinal("william gibson", "neuromante"): 9.13  
getPrecioFinal("Isaac Asimov", "Segunda Fundación"): 7.128  
getPrecioFinal("Isaac Newton", "Arithmetica Universalis"): 0.0
```