



Apellidos:

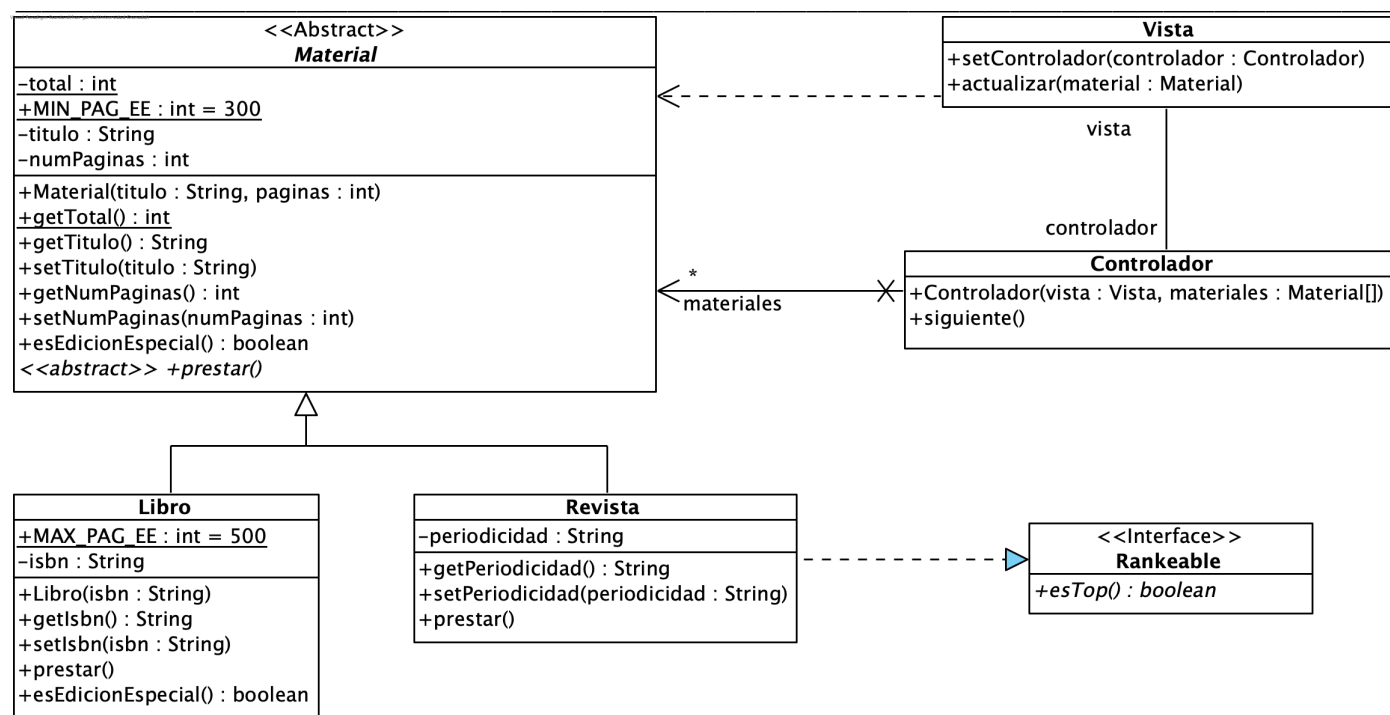
Nombre: Grupo:

INSTRUCCIONES:

- Escriba en cada folio «APELLIDOS, NOMBRE» (en **MAYÚSCULAS**) y entregar al final (si no se entrega, no se corrige).
- Acceda a su cuenta personal especificando el código **w10pdoovix**.
- Acceda desde el **escritorio** a la unidad de disco «**Home**» y guarde allí sus ejercicios del examen.
- Cree el Proyecto en Netbeans: seleccione File > New Project > Java with **Ant** > Java Application (¡que nadie use Maven!)
- La entrega no puede contener errores de compilación. Use el icono del Martillo con la escoba  y luego el botón play. .
- Por eso, añada a la entrega de Prado una captura de pantalla **completa** de la salida por consola de la ejecución.
- Entregue un zip de su proyecto: seleccione File > Export Project > To ZIP > Root Project (comprobar que es su proyecto) > Build ZIP > Browse > escritorio > disco Home > apellido1_apellido2_nombre.zip > Export

Consideraciones iniciales

[0,5 puntos] Evite usar múltiples retornos en sus métodos optando, en cambio, por un único punto de retorno por razones de diseño. Evite el uso de «números mágicos» en su código. Además, en lugar de asignar valores booleanos directamente como true o false, construya expresiones booleanas.



Ejercicio 1. [1,5 puntos] Implemente la clase **Material** según el diagrama. Ten en cuenta que está en cursiva y el método **prestar()** también está en cursiva. Se desea contabilizar el total de materiales que se **creen** en el sistema. Implementa el método **esEdicionEspecial()** de tal modo que devuelva que es un edición especial es el número de páginas (numPaginas) es mayor o igual a **300**.

Ejercicio 2. [2 puntos] Implemente la clase **Libro** según se indica en el diagrama de clases. El método **prestar()** enviará un mensaje por consola diciendo «Se ha prestado el libro XXX cuyo isbn es YYY», donde XXX es el título del libro e YYY es el isbn. Sobreescrba el método **esEdicionEspecial()** de Libro, teniendo en cuenta que un libro será edición especial si el número de páginas está entre **300** y entre **500**; si tiene más de 500 páginas, ya no lo es o si tiene menos de 300. Por favor, **sobreescrbe y reutiliza** 😊.

Ejercicio 3. [1 punto] Implemente la clase **Revista** según se indica en el diagrama de clases. El método *prestar()* enviará un mensaje por consola diciendo «Se ha prestado la revista XXX cuya periodicidad es YYY», donde XXX es el título de la revista e YYY es la periodicidad. Implemente también la interfaz **Rankeable**, donde el método *esTop()* devolverá que la revista es top si tiene una periodicidad igual a la constante “semanal”.

Ejercicio 4. [2 puntos] Implemente en la clase **Vista** que se suministra, los métodos del diagrama de clases. Tenga en cuenta que hereda de JFrame, por lo que dispondrá de los atributos de tipo JTextField que hemos preparado con los nombres *tfTitulo*, *tfTipo* y *tfEsEdicionEspecial*.

- Implementa el método *actualizar(Material material)* de modo que cambie:
 - o El texto de la caja de texto *tfTitulo* con el título del material
 - o El texto de *tfTipo* con nombre del tipo de material, Libro o Revista.
 - o El texto “Es edición especial” o “No es edición especial” en *tfEsEdicionEspecial*
 - o Después de esas 3 líneas de código, puede usar las líneas siguientes para repintar la ventana:
 - `super.repaint();`
 - `super.revalidate();`
- Implemente la acción de hacer clic en el botón **Siguiente**. Ahora mismo, verás un mensaje por consola que dice cambiando al siguiente material. Cambie ese código y dígame al controlador que cambie de material.

Ejercicio 4. [1 punto] Implemente la clase **Controlador** según el diagrama:

- Implemente el **constructor** para recibir la vista, los materiales y actualice la vista justo después.
- Implemente el método *siguiente()* para que cada vez que en la Vista pulsemos el botón siguiente, el controlador le diga a la Vista cuál es el siguiente Material que debe mostrar en la ventana.

Ejercicio 5. [2 puntos] En el programa principal, pruebe lo siguiente:

- a) **[0,25 puntos]** Cree un `ArrayList<Material>` materiales con 2 libros y 2 revistas. Un libro que sea edición especial y el otro no; y una revista que sea top y otra que no lo sea. Puedes reutilizar el método *main* que se proporciona
- b) **[0,25 puntos]** Después de la línea donde se crea la Vista, cree un Constructor con dicha vista y el `ArrayList` de materiales anterior.
- c) **[0,25 puntos]** Imprima el número de materiales total que se ha creado en el sistema.
- d) **[0,5 puntos]** Imprima el número de libros que hay en el `ArrayList` de materiales; no imprima el número de revistas.
- e) **[0,5 puntos]** Imprima cuáles son las revistas top de ese array de materiales, es decir los títulos de las revistas top.
- f) **[0,25 puntos]** Imprima si un material es un número especial o no, de ese array.
- g) 5.3, 5.4 y 5.5, por favor, hágalo en un único bucle, no? 🤪

[🎁 1 punto extra 🎁] Crea una clase *Usuario* que tiene un `ArrayList<Material>` de materiales. Crea un constructor de copia de *Usuario* que realice una copia profunda para que los datos de los materiales, ya sean libros o revistas se copien por completo su estado, sus valores; y no sus referencias.

```
Usuario u1 = new Usuario("mi-nombre", materiales);
Usuario u2 = new Usuario(u1);
```