¿Qué sucede si no usamos ***@Override***?

El uso de @Override es opcional, pero muy importante.

@Override es una anotación: Le dice al compilador de Java que el método que sigue está sobrescribiendo un método de la clase padre. Por ejemplo, en RecomendacionPorGenero y RecomendacionPopular, usamos @Override para indicar que estamos reemplazando el método mostrar() de la clase Recomendacion.

**Si NO usamos @Override**:

1. **El código sigue funcionando igual**: Java no requiere @Override para que la sobrescritura funcione. Si el método tiene la misma firma (mismo nombre, mismos parámetros, mismo tipo de retorno) que el método de la clase padre, se sobrescribirá correctamente.
2. **Menos seguridad**: Sin @Override, el compilador no verifica si el método realmente está sobrescribiendo algo. Si cometes un error (por ejemplo, escribes mal el nombre del método o cambias los parámetros), Java pensará que estás creando un método nuevo en lugar de sobrescribir el original, y no obtendrás el comportamiento esperado.

**Por qué usar @Override**:

* Evita errores humanos (como escribir mal el nombre del método).
* Hace el código más claro para otros programadores, indicando que estás sobrescribiendo un método.
* Es una buena práctica, especialmente para principiantes, porque te ayuda a entender mejor la herencia.

@Override: Esto indica que estamos sobrescribiendo el método de la clase padre.

extends Recomendacion: Esto hace que RecomendacionPorEdad herede de Recomendacion, por lo que tiene acceso a la variable titulo y al método mostrar().