

### Apartado 3:

El diagrama analizado presenta errores de diseño que comprometen la escalabilidad y la eficiencia del sistema. En primer lugar, se observa una notable redundancia de datos y falta de abstracción, ya que las tres variantes de suscripción replican atributos idénticos (codigoSuscripcion, fechaInicio, fechaFin, isActiva). Por otro lado, el atributo seccion en la clase Artículo utiliza un tipo String genérico, ignorando que las posibles secciones se restringen a cuatro categorías específicas.

Asimismo, el modelo de relaciones resulta ineficiente y poco intuitivo: la clase Suscriptor carece de navegabilidad directa hacia sus suscripciones, dependiendo en su lugar de atributos booleanos y listas de códigos que obligan a realizar búsquedas costosas y manuales. Esta fragmentación se refleja en la clase Periodico, cuyos métodos de consulta obligan a iterar repetidamente sobre listas para reconstruir la información de un usuario.

En cuanto al diagrama corregido, este resuelve la redundancia previa mediante la implementación de una jerarquía de herencia, donde una clase abstracta Suscripcion centraliza los atributos comunes. En relación a la clase Artículo se ha sustituido el tipo genérico por un enumerado Seccion, lo que elimina ambigüedades en las categorías. A nivel de eficiencia, se ha transformado el modo de buscar información: el uso de colecciones avanzadas como Map<String, Suscriptor> permite un acceso directo (O(1)) por DNI, eliminando las iteraciones costosas sobre listas completas. Además, se ha establecido una navegabilidad directa desde Suscriptor hacia sus suscripciones, lo que permite que cada objeto gestione su propio estado activo sin necesidad de banderas booleanas externas o métodos de búsqueda fragmentados en la clase Periodico.

Diagrama de clases corregido:

