

Conjuntos ordenados y Mapas con claves ordenadas.

Crea un paquete llamado colecciones.

Crea una Clase llamada **Usuario** (dentro del paquete colecciones) que contenga los siguientes atributos:

- nombre (String)
- email (String)
- fechaNacimiento (String con el formato DD-MM-AAAA)

Crea una clase **PruebaColecciones** (dentro del paquete colecciones) e incluye el método **main** para realizar los siguientes ejercicios:

Conjuntos ordenados

- Declara y construye un conjunto ordenado (**TreeSet**) que almacene cadenas (String).
- Introduce varias cadenas en el conjunto.
- Realiza un recorrido (**for each**) del conjunto y comprueba que el iterador retorna los elementos ordenados según el orden lexicográfico de las cadenas (orden alfabético). Este comportamiento se debe a que las cadenas implementan la interfaz Comparable<T>, es decir, tienen un orden natural.

- Implementa el orden natural de la clase Usuario. El orden natural debe basarse en el orden alfabético del nombre de usuario. Utiliza el método **compareTo** de la clase String para comparar los nombres de los usuarios.
- Declara y construye un conjunto ordenado de usuarios.
- Construye varios usuarios y añádelos al conjunto. Observa que si introducimos varios usuarios con el mismo nombre sólo el primero que se inserte permanece en el conjunto.

Un conjunto ordenado determina que un elemento es repetido si en la comparación obtiene el valor 0. Por tanto, es recomendable que la implementación del orden natural sea consistente con la igualdad (equals).

- Recorre el conjunto y muestra los usuarios. Observa que el iterador retorna los elementos según el orden natural.
- Implementa un comparador de usuarios (**interfaz java.util.Comparator<T>**) que ordene los usuarios según el orden alfabético inverso de sus nombres.
- Declara y construye un conjunto ordenado de usuarios que aplique el orden implementado por el comparador anterior. Añade usuarios a la colección (método addAll) y realiza un recorrido para comprobar que el orden es el correcto.
- Utiliza los métodos **sort** de la clase Collections para ordenar la lista, según el orden natural y con el comparador de orden inverso. Muestra el contenido de la lista tras ambas ordenaciones

Mapas con claves ordenadas:

- Declara y construye un mapa ordenado (**TreeMap**) que asocie cadenas (String) con objetos del tipo **usuario**.
- Recorre el último conjunto ordenado e inserta en el mapa los objetos del conjunto. Utiliza el nombre del usuario como clave en el mapa.
- Recorre las claves del mapa ordenado (método **keySet**) y muestra por la consola las claves. Observa que el mapa retorna las entradas ordenadas por clave. Por defecto, utiliza el orden natural de la clase que actúa de clave (String).
- Implementa un comparador de cadenas que implemente el orden alfabético inverso.
- Para ello utiliza el orden natural de las cadenas (método **compareTo**).
- Declara y construye un mapa ordenado que utilice el comparador. Introduce las entradas del mapa anterior al nuevo mapa.
- Recorre y muestra por la consola las claves del mapa. Observa que las claves están ordenadas aplicando el orden del comparador.