



--	--	--	--	--

APELLIDOS: \_\_\_\_\_ NOMBRE: \_\_\_\_\_

DNI: \_\_\_\_\_ TITULACIÓN: ☐ IC ☐ IS ☐ TI ☐ DG GRUPO: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Disponemos de datos sobre cortes de suministro eléctrico producidos en los Estados Unidos. Para cada corte se tiene la siguiente información:

- **Descripción:** causa del corte, de tipo String, consultable.
- **Fecha de inicio del corte:** fecha y hora de inicio del corte, de tipo LocalDateTime, consultable.
- **Fecha de restablecimiento:** fecha y hora en la que se restableció el suministro, de tipo LocalDateTime, consultable.
- **Compañía:** compañía responsable del suministro de la electricidad, de tipo String, consultable.
- **Región:** región en la que se produjo el corte, de tipo String, consultable.
- **Pérdida de demanda:** cantidad de potencia perdida durante el corte en MW, de tipo Double, consultable. Si no es conocida, tomará el valor null.
- **Consumidores afectados:** número de consumidores afectados por el corte, de tipo Integer, consultable. Si no es conocido, tomará el valor null.
- **Etiquetas:** etiquetas asociadas al corte, de tipo List<String>, consultable.
- **Severidad:** nivel de severidad del corte, de tipo Nivel, consultable. El nivel puede tomar los valores BAJO (si el número de consumidores afectados ha sido inferior a 10000 o es null), MEDIO (si está comprendido entre 10000 y 100000, ambos inclusive) y ALTO (si es superior a 100000).
- **Es crítico:** indica si el corte ha sido crítico o no, de tipo Boolean, consultable. Un corte es crítico cuando la pérdida de demanda es conocida y superior a 200 MW, o tiene una duración superior a las 10 horas.

Por ejemplo, la siguiente línea del fichero:

```
Severe Weather;3/13/2011;14:20;3/14/2011;15:46;PacifiCorp;Oregon;Unknown;9000;  
winter storm, snow
```

indica que, entre el 13 de marzo de 2011 a las 14:20 horas y el 14 de marzo a las 15:46 horas, se produjo un corte en la región de 'Oregon', suministrada por la compañía 'PacifiCorp'. El corte afectó a 9000 usuarios, siendo la pérdida de demanda desconocida. La causa del corte fue 'Severe Weather', y se le asociaron las etiquetas 'winter storm' y 'snow'.

Cree en el proyecto los paquetes fp.cortes y fp.cortes.test, y dentro de ellos implemente los tipos que se piden a continuación.

### Ejercicio 1: Tipo base (1 punto)

Implemente el tipo CorteElectrico mediante un record, de acuerdo con la siguiente información:

#### Constructores:

- C1: recibe un parámetro por cada propiedad básica del tipo, en el mismo orden en el que están definidas.

Representación como cadena: una cadena con todas las propiedades básicas del tipo.

#### Restricciones:

- R1: la fecha de restablecimiento debe ser igual o posterior a la fecha de inicio del corte.
- R2: el número de consumidores afectados debe ser mayor o igual que 0, o null.
- R3: la lista de etiquetas debe contener al menos una etiqueta.

Criterio de igualdad: dos cortes son iguales si tienen la misma fecha de inicio y región.

Orden natural: los cortes se ordenan por fecha de inicio, y a igualdad de esta, por región.

## Ejercicio 2: Tipo contenedor (1 punto)

Implemente el tipo `InformeCortes` de acuerdo con la siguiente descripción:

### Propiedades:

- **Nombre:** nombre del informe, de tipo `String`, consultable.
- **Fecha:** fecha del informe, de tipo `LocalDate`, consultable.
- **Cortes:** lista de cortes de suministro, de tipo `List<CorteElectrico>`, consultable.
- **Número de cortes:** número de cortes que contiene el informe, de tipo `Integer`, consultable.

### Constructores:

- C1: recibe un parámetro por cada propiedad básicas del tipo, excepto los cortes, que inicialmente estarán vacíos.
- C2: recibe un parámetro para el nombre y la fecha, y un `Stream<CorteElectrico>` para inicializar los cortes.

Representación como cadena: una cadena con todas las propiedades del tipo, excepto los cortes, aunque sí se debe mostrar el número de cortes del informe.

Criterio de igualdad: dos informes son iguales si lo son sus nombres y sus fechas.

### Otras operaciones:

- *`void incorporaCorte(CorteElectrico c)`*: añade el corte al final de la lista de apagones, si el corte no está ya en la lista. Si está no hace nada.
- *`void incorporaCortes(List<CorteElectrico> cortes)`*: añade los cortes de la lista al final de la lista de cortes.
- *`void eliminaCorte(CorteElectrico c)`*: elimina el corte del informe. Si el corte no está en el informe, no hace nada.

## Ejercicio 3: Factoría (1 punto)

Cree una factoría `FactoriaCortes` que permita leer los datos de un fichero CSV y almacenarlos en el tipo contenedor. La factoría debe contener dos métodos:

- *`InformeCortes leerCortes(String nombreFichero)`*: lee un fichero de cortes de suministro y construye un objeto de tipo `InformeCortes` con la fecha del día actual y un nombre formado por la concatenación de la palabra "Informe" y la fecha del día actual.
- *`CorteElectrico parsearCorte(String lineaCSV)`*: crea un objeto de tipo `CorteElectrico` a partir de una cadena de caracteres. La cadena de caracteres debe tener el mismo formato que las líneas del fichero CSV.

## Ejercicio 4: Tratamientos secuenciales

Añada los siguientes tratamientos secuenciales al tipo contenedor y pruébelos en la clase `TestInformeCortes`. Debe resolver los métodos **mediante streams**, salvo que se le indique expresamente que debe utilizar bucles:

1. *`Double mediaAfectadosEnRegiones(Nivel s, Set<String> regiones)`*: calcula el número medio de consumidores afectados por cortes del nivel de severidad dado como parámetro en el conjunto de regiones dado como parámetro. Tenga en cuenta que debe excluir del cálculo los cortes cuyo número de consumidores afectados no sea conocido. Si no se puede calcular la media, se debe lanzar la excepción `NoSuchElementException`. Realice un test del método. Ejemplo: si el nivel de severidad es MEDIO y el conjunto es ["Maryland", "Massachusetts"], el resultado debe ser 86500.0 (1 punto)

2. *List<String> compañíasCortesMasRecientes(String etiqueta, Integer n)*: obtiene una lista con los nombres de las compañías con cortes más recientes que contienen la etiqueta dada, del más reciente al más antiguo, con un máximo de n nombres y sin repetir ninguno. Realice un test del método. Ejemplo: si la etiqueta es "winter storm" y n es 3, el resultado debe ser ["Duke Energy Carolinas", "Tennessee Valley Authority", "Ameren Missouri"]. (1 punto)
3. *SortedMap<String, SortedSet<String>> compañíasConCortesCriticosPorRegion()*: obtiene un SortedMap que relaciona cada región con los nombres de las compañías que han sufrido cortes críticos en dicha región. Resuelva este ejercicio **mediante bucles y mediante streams**. Realice un test del método. Con los datos del fichero, la primera pareja del SortedMap obtenido es {"Alabama, Florida, Georgia" = ["Southern Company"]}. (2 puntos, 1 por cada versión)
4. *Map<Nivel, Double> porcentajeCortesPorSeveridadEnRegion(String region)*: obtiene un Map que relaciona cada nivel de severidad con el porcentaje de cortes con dicha severidad que se han producido en la región dada como parámetro, con respecto al total de cortes producidos en dicha región. Realice un test del método. Ejemplo: si la región es "Madison, Wisconsin", el resultado debe ser {BAJO = 100.0}. (1,5 puntos)
5. *String compañíaConMasAfectadosEnFecha(LocalDate f)*: obtiene el nombre de la compañía con el mayor número total de consumidores afectados en una fecha dada. Se considerarán los cortes que estén activos en esa fecha, es decir, aquellos en los que aún no se haya restablecido el suministro, y cuyo número de consumidores afectados sea conocido. Si no existe ninguna compañía con consumidores afectados en esa fecha, se lanzará una excepción de tipo *NoSuchElementException*. Realice un test del método. Ejemplo: si la fecha es 17/11/2013, el resultado debe ser "Detroit Edison Co". (1,5 puntos)