

## Examen Práctico Convocatoria de Junio de 2023

---

### Previo. Gestión de fechas.

- Para representar las fechas utilizaremos las clases `java.time.LocalDate` (sólo el día) y `java.time.LocalDateTime` (día y hora) sin información de la zona horaria de acuerdo con el sistema de calendario ISO-8601.
  - El formato (por defecto) es `año-mes-día` para `LocalDate` y `año-mes-díaThora:min:segundos`, `2023-01-16T10:00:00` para `LocalDateTime`.
  - Las fechas (tanto `LocalDate` como `LocalDateTime`) disponen del método de clase `now()` que devuelve la fecha, y la fecha y hora actual, respectivamente.
  - Las fechas (tanto `LocalDate` como `LocalDateTime`) son **comparables**, aun así, también están disponibles los métodos `isAfter`, `isBefore` e `isEqual`.
  - Los objetos `LocalDate` y `LocalDateTime` son **inmutables**.
- 

El objetivo de la aplicación es la implementación de un sistema de gestión de procesos de votación telemáticos (televoto) en una organización.

### 1. Funcionalidad básica

Una **persona** representa a un miembro de una organización. La clase que implementa el tipo `Persona` define objetos inmutables que se caracterizan por:

- Dni: cadena de texto.
- Nombre: cadena de texto.
- Fecha de nacimiento (`LocalDate`).

La clase define un único constructor que inicializa todos los atributos.

Implementa los métodos `equals/hashCode` en la clase `Persona`. Se entiende que dos personas son iguales si tienen el mismo dni.

Implementa el *orden natural* en la clase `Persona` (interfaz `Comparable`) atendiendo al orden alfabético de los nombres y en caso de empate se ordenan por dni.

Un **resultado** representa el número de votos que ha recibido una determinada opción de voto registrada en el proceso de votación telemático. Se corresponde con un tipo de datos que define objetos inmutables caracterizados por dos atributos: 1) la opción de voto (cadena de texto) y 2) el número de votos que ha recibido. Esta clase ofrece dos constructores, uno que recibe como parámetro el valor inicial de los dos atributos y otro en el que se omite el número de votos y se entiende, en ese caso, que el número de votos es 0.

Un **proceso de televoto** representa un proceso de votación telemática iniciado en una organización. Estos procesos pueden encontrarse en distintas **fases** (enumerado): “configuración”, “votación” y “resultados” (ver figura en la página siguiente). La funcionalidad de la clase que se pueda ejecutar dependerá de la fase del proceso en la que se encuentre. Por ejemplo, en la fase de “votación” no se puede añadir ninguna opción de voto. Se tiene que gestionar el control de estados en la implementación de la funcionalidad de la clase como parte del control de precondiciones.

Las propiedades que caracterizan a un proceso de televoto son:

- Descripción. Cadena de texto. El valor establecido en la construcción no puede cambiar.
- Organizador: Cadena de texto con el nombre de la organización que inicia el proceso de televoto. El valor de esta propiedad no puede cambiar una vez establecido.
- Fase en la que se encuentra el proceso de televoto.
- Activo: Valor booleano que indica si el proceso de votación todavía no se ha cerrado, esto es, no está en la fase “resultados”. - Censo: **colección ordenada** de personas que tienen derecho a votar en el proceso. El censo se establece en la construcción y no se puede modificar.
- Número de electores: se corresponde con el número de personas del censo.
- Votos: mapa que asocia cada opción de voto (cadena de texto) con un entero que se corresponde con el contador del número de votos que ha recibido. Este atributo es un atributo de implementación que no se puede consultar.
- Opciones de voto: Colección de cadenas de texto con las opciones que se pueden votar.
- Votantes: colección de personas que ya han votado.

La clase ofrece un constructor que recibe como parámetro la descripción, el organizador y el censo. La colección que registra los votantes se inicializa a vacío, mientras que el mapa de votos se inicializa con la opción de voto “abstención” a la que se le asocia el contador a 0. La fase del proceso se inicializa a “configuración”.

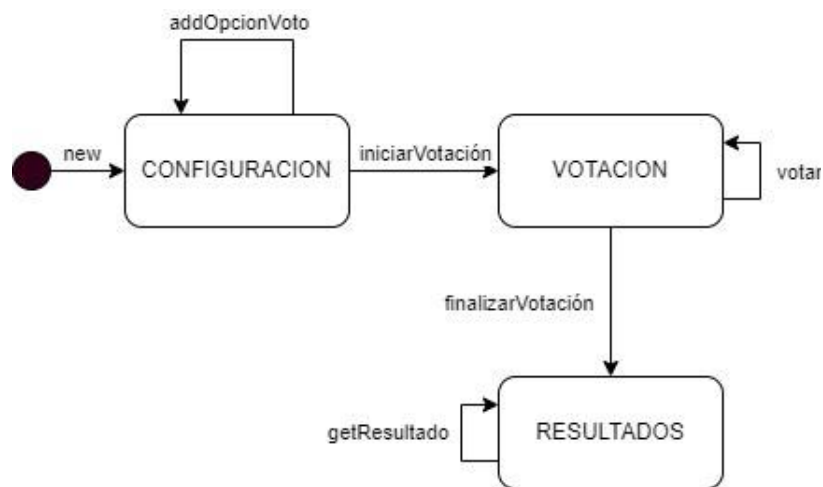
La clase ofrece la siguiente funcionalidad:

- Obtener una persona del censo: El método recibe el dni de la persona y devuelve el objeto Persona, si está en el censo, o el valor nulo si ese dni no se corresponde con ninguna de las personas censadas.
- Consultar si una persona está en el censo. El método recibe el dni de la persona y devuelve `true` si existe una persona en el censo con ese dni y `false` en caso contrario.
- Añadir opción de voto: Recibe como parámetro una cadena de texto con la opción de voto que se presenta, y devuelve un valor booleano para indicar si la presentación ha sido correcta o no. Solo se puede ejecutar este método si la fase en la que se encuentra el proceso de televoto es “configuración”. Si la opción de voto no existe en el mapa de votos, se añade la entrada en el mapa de votos asociando la nueva

opción de voto con el contador a 0 y devuelve `true`, en caso contrario devuelve `false`.

- Iniciar votación: devuelve un valor booleano que indica si se ha iniciado la nueva fase o no. Solo se puede ejecutar este método si la fase en la que se encuentra el proceso de televoto es “configuración”, en ese caso, si hay al menos dos opciones de voto (incluida la “abstención”), cambia la fase del proceso a “votación” y devuelve el valor `true`. En caso contrario devuelve `false`.
- Votar. Método que recibe como argumentos dos cadenas de texto: dni de la persona que vota (votante) y la opción de voto. Sólo se puede ejecutar este método si la fase en la que se encuentra el proceso es “votación”. En ese caso, se tiene que cumplir que: 1) el votante está en el censo, 2) no ha votado todavía y 3) el voto se corresponde con una de las posibles opciones de voto. Si se cumplen estas condiciones se contabiliza el voto emitido, esto es, se incrementa el contador asociado a la opción de voto, se registra al votante y devuelve `true`. En caso contrario devuelve `false`.
- Versión sobrecargada del método votar en el que se omite la opción de voto y se entiende que la opción a la que se está votando es “abstención”.
- Finalizar votación: Sólo se puede ejecutar este método si la fase en la que se encuentra el proceso es “votación”, en ese caso cambia la fase del proceso de televoto a “resultados”.
- Consultar el resultado. El método de consulta del resultado del proceso de televoto devuelve una lista con los resultados de la votación ordenados de mayor a menor número de votos. Este método sólo se puede ejecutar si la fase en la que se encuentra el proceso es “resultados”. Es un requisito implementar este método haciendo uso del *procesamiento basado en streams*.

El siguiente diagrama resume las fases o estados en los que se puede encontrar un proceso de votación telemática y las operaciones aplicables en cada una de ellas. Los métodos de consulta de las propiedades se pueden ejecutar en cualquiera de las fases.



## 2. Tipos de procesos de votación telemática: Elecciones y Referéndum.

Un **proceso electoral** es un proceso de televoto en el que se presenta uno o más candidatos (opciones de voto) de entre los que saldrá elegido el más votado. Por tanto, en este tipo de proceso hay que tener en cuenta que sólo se puede añadir una opción de voto si la opción representa el dni de una de las personas del censo. La nueva funcionalidad que ofrece la clase es:

- Consultar los candidatos. Devuelve una colección de las personas candidatas. Es un requisito implementar este método haciendo uso del *procesamiento basado en streams*.  
**Nota:** Tened en cuenta que entre las opciones de voto se incluye la “abstención” que no se corresponde con un candidato válido.
- Consultar ganadores. Devuelve una colección que incluye la persona/s que más votos han recibido.
- Consultar si ha habido empate. Hay empate si hay más de un ganador.

Un **referéndum** es un proceso de televoto en el que se somete a votación una única propuesta que es la que se establece en la *descripción*, por ejemplo, “Modificación de la normativa de TFG/TFM de la FIUM”. En este tipo de proceso de votación telemática las únicas opciones de voto posibles son: “SI”, “NO” y “abstención” que deben quedar establecidas en la construcción, donde se inicia automáticamente la fase de votación.

Esta clase añade una nueva propiedad para registrar el instante de tiempo (`LocalDateTime`) en el que se realizan los votos. Así, cada vez que se vota, se tiene que registrar en la colección de tiempos el instante actual.

La funcionalidad que ofrece esta clase es consultar si el referéndum ha sido aprobado, es decir, devuelve un valor booleano indicando si el número de votos de la opción “SI” se corresponde con la mayoría. La mayoría es, al menos, mayor que la mitad del número de electores (personas en el censo). Este método de consulta sólo se puede ejecutar si la fase en la que se encuentre el proceso de referéndum es “resultados”.

## 3. Métodos de `Object`

Implementa el método `toString` en todas las clases.

Implementa el método `clone` en la jerarquía de los procesos de televoto siguiendo las recomendaciones de la asignatura. Teniendo en cuenta que la copia de un proceso de televoto mantiene las opciones de voto y el censo, pero borra la información relacionada con los votos. La fase inicial del proceso copiado debe ser “votación”.

## 4. Programa

Implementa un **Programa** con la siguiente funcionalidad:

- Crea los siguientes objetos Persona que constituyen el censo de los procesos de televoto que se van a declarar:

```
Persona julio = new Persona("34.814.689-V", "Julio",  
                             LocalDate.of(1980, 6, 15));  
Persona matias = new Persona("17.234.567-K", "Matías",  
                             LocalDate.of(1976, 2, 5));  
Persona andrea = new Persona("23.654.785-A", "Andrea",  
                             LocalDate.of(1975, 4, 21));  
Persona vega = new Persona("35.835.285-P", "Vega",  
                           LocalDate.of(1970, 3, 9));
```

- Crea una lista de procesos de votación telemática y añade: 1) un proceso electoral con la descripción “Elecciones a director de departamento” y 2) un referéndum con descripción “Normativa Fiestas de San Alberto”. En ambos casos el organizador es “DIS” y el censo está constituido por las personas creadas en el paso anterior.
- Proceso electoral:
  - o Añade como opción de voto a Vega y a Andrea.
  - o Inicia la votación.
  - o Vega se vota a re.
  - o Andrea vota a Vega.
  - o Matías se abstiene.
  - o Pedro con dni 25.789.345-C intenta votar.
- Referéndum:
  - o Vega vota “SI”.
  - o Andrea vota “Me da igual” (voto nulo).
  - o Julio vota “NO”.
  - o Matías se abstiene.
- Recorre la lista de procesos y para cada uno de ellos:
  - o Si el proceso está activo finalizar la votación
  - o Mostrar en la consola la descripción del proceso
  - o Mostrar los resultados del proceso
  - o Si es un referéndum consultar si se ha aprobado y mostrar el resultado en la consola.
- En el proceso electoral consultar si existe algún candidato que tenga más de 60 años utilizando el procesamiento basado en streams.

## 5. Métodos de utilidad

- a) En una clase `Utilidad`, implementa un **método genérico** encargado de poblar una colección con los objetos proporcionados por un iterador. Por tanto, este método tiene como parámetros una colección y un iterador. Prueba este método en el programa principal creando una lista vacía e introduciendo los objetos que proporcione un iterador de la lista de procesos de votación.
- b) El método `parseDouble` de la clase `Double` convierte una cadena de texto en un número real. Este método notifica la excepción `NumberFormatException` si la cadena no representa un número. Implementa en la clase `Utilidad` un método que reciba como parámetro un argumento variable de cadenas y retorne una lista de números reales resultado de la conversión de esas cadenas. En caso de que alguna cadena no pueda ser convertida, el método finalizará notificando la excepción comprobada `java.text.ParseException`. Realiza una prueba del método en el programa principal.