

Nombre:	
Código:	Fecha:

**Unidad 4: Algoritmos de Ordenamiento y Búsqueda**

Al finalizar esta unidad, el estudiante estará en capacidad de:

OE4.1 Implementar algoritmos clásicos de ordenamiento de datos en estructuras de datos lineales y aplicarlos en la solución de un problema.

OE4.2 Implementar algoritmos clásicos de búsqueda de información en estructuras de datos lineales y aplicarlos en la solución de un problema.

OE4.3 Reconocer la diferencia entre orden natural y orden parcial de los objetos por medio de la descripción de utilidades de las interfaces Comparable y Comparator.

**Unidad 3: Persistencia**

OE3.3 Hacer persistir el estado del modelo de solución del problema durante la ejecución de un programa y restaurarlo cuando se requiera.

**Enunciado**

La Feria de Cali es uno de los eventos más importantes de la ciudad, considerándose uno de los principales destinos turísticos de Colombia y Latinoamérica entre el 25 y 30 de diciembre de cada año. Los recursos destinados para esta feria son administrados por Corfecali, la empresa que ha solicitado de tu asesoría para gestionar específicamente la información de los artistas. Por ahora, la compañía sólo necesita su apoyo en el desarrollo de métodos de ordenamiento y persistencia. Para esto, te ha entregado el siguiente diagrama de clases, el cual permite modelar la información que se requiere hasta el momento.

Su labor como ingeniero de sistemas consiste en el desarrollo de los siguientes puntos:

**1. [50%] Comparación por orden natural.**

Se ha definido que el orden natural de los artistas consiste en dos criterios: el nombre y la identificación. Es decir, los artistas se deben comparar con base en el nombre. En el caso en que los nombres sean iguales, entonces se debe hacer la comparación con base en la identificación teniendo en cuenta que este valor será diferente en cada artista. Con base en esto, realiza lo siguiente:

- a. [20%] Completa el diagrama de clases, de tal manera que se pueda definir el orden natural a la clase artista.
- b. [30%] Implementa el método en la clase artista que permite realizar la comparación. Recuerda que este método compara los artistas en primer lugar según su nombre. Si los nombres son iguales, entonces realiza la comparación por el id.

Retorna un valor negativo:

- Si el artista con el que se compara tiene un nombre MAYOR.
- Si tiene el mismo nombre y tiene un id MAYOR.

Retorna un valor positivo:

- Si el artista con el que se compara tiene un nombre MENOR.
- Si tiene el mismo nombre y tiene un id MENOR.

Retorna 0:

- Si el artista con el que se compara tiene el mismo nombre y el mismo id.

**2. [30%] Ordenamiento por selección**

Ordena los artistas usando la estrategia de ordenamiento selección. Tenga en cuenta que este ordenamiento debe hacerse con base en el orden natural de los objetos Artista. Para esto puedes basarte en el siguiente contrato.

```
/**
```

```
* Clase: Desfile <br>
```

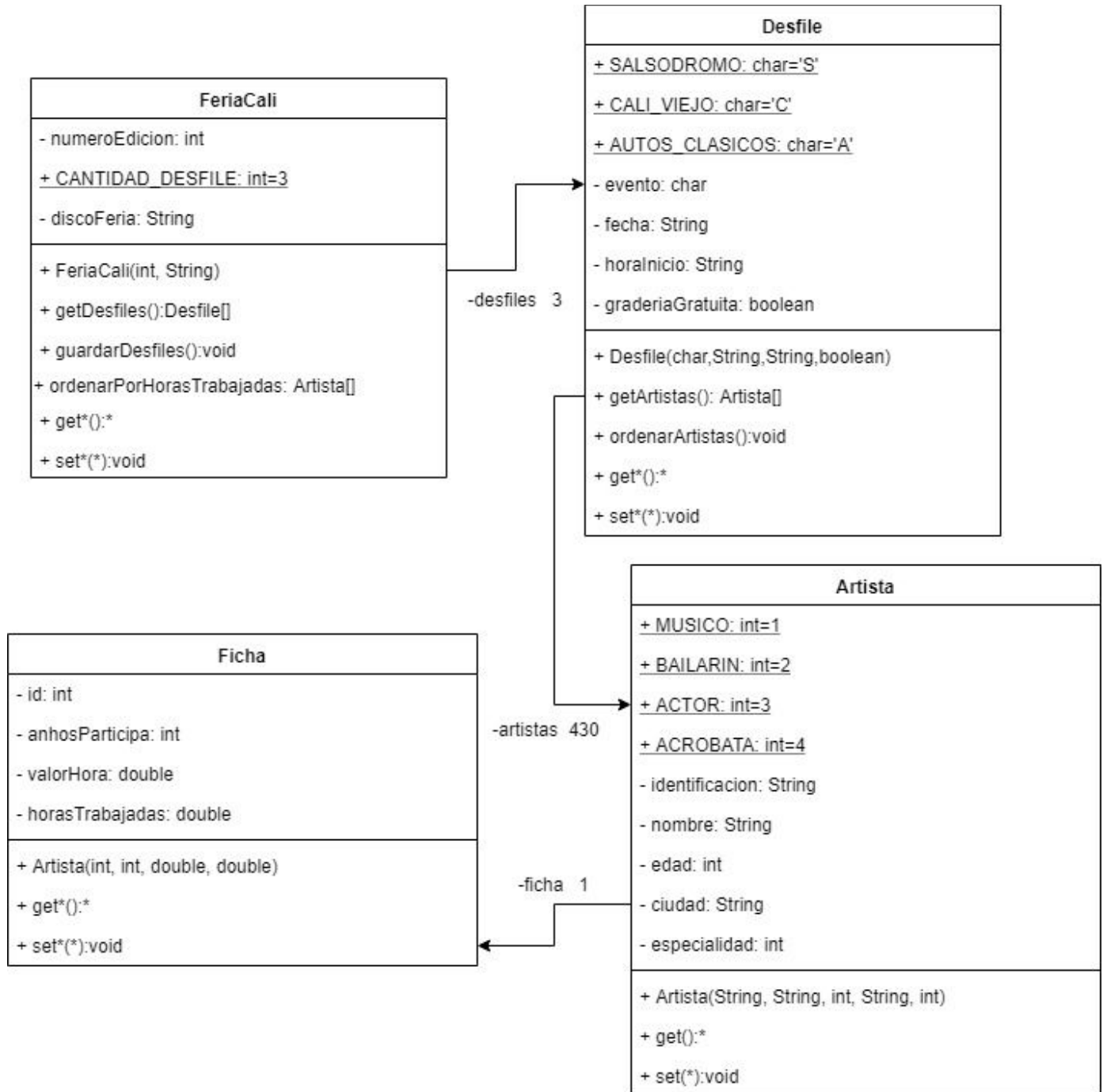
```
* Método: ordenarArtistas(): void <br>
```

```
* Este método organiza ascendentemente el arreglo de
```

```
* artistas según su orden natural <br>
```

```
* <b>post: </b> El arreglo de artistas está ordenado según los criterios dados.
```

```
*/
```



### 3. [20%] Persistencia

Construye el método que permite **serializar** los desfiles de la feria de Cali, de tal manera que sea posible guardar el estado de los objetos de tipo Desfile. Para realizar dicho método, debe tener como referencia el siguiente contrato.

/\*\*

\* Clase: FeriaCali

\* Método: GuardarDesfiles(): void

\* Este método permite dar persistencia a los objetos de tipo Desfile. Por lo tanto, el

\* estado de estos objetos son guardado en archivos binarios. <br>

\* <b>post: </b> Los objetos de tipo Desfile son guardados en archivos binarios. \*/