

 Proyecto Cupi2	ISIS-1205 Algorítmica y Programación II Consideraciones adicionales de diseño
Ejercicio:	n8_CupiPalooza
Autor:	Equipo Cupi2
Semestre:	2017-2

Manejo de excepciones propias

Para tratar los errores específicos que se presentan en la ejecución de la aplicación y poder asociar más información al error (no solo un mensaje de texto), clasificar errores en grupos y aplicar diferentes estrategias de manejo de error, se ha decidido utilizar clases propias de excepción. Estas son: `PersistenciaException`, `FormatoArchivoException`, `ElementoExistenteException` y `CupoMaximoException`.

1. `PersistenciaException`

Es la clase de excepción que se lanza cuando se presenta un error de serialización/deserialización, al leer o escribir el archivo binario con la información del estado del mundo, o al leer o escribir un archivo de texto plano. El mensaje asociado con la excepción describe el problema que se presentó. Cuando esta excepción ocurre se registra en el archivo de log, acompañada de la fecha y hora en la que se generó la excepción (ver documento de descripción, sección “Persistencia”).

2. `FormatoArchivoException`

Es la clase de excepción que indica que hubo un problema procesando el archivo con la información de los espacios. El mensaje asociado con la excepción describe el problema que se presentó. Cuando esta excepción ocurre se registra en el archivo de log, acompañada de la fecha y hora en la que se generó la excepción (ver documento de descripción, sección “Persistencia”).

3. `ElementoExistenteException`

Es la clase de excepción que se lanza cuando se intenta agregar un escenario con un patrocinador que ya está financiando otro escenario, o se intenta agregar una banda a un escenario pero ya existe en este una banda con el mismo nombre. El mensaje asociado a la excepción indica el nombre repetido y si este pertenece a una banda o a un patrocinador, ambas cualidades deben almacenarse en los atributos de la clase (Ver modelo conceptual). Para la creación de este mensaje haga uso del operador condicional. Al crear la excepción, se registra el error en el archivo de log con la fecha y hora de ocurrencia (ver documento de descripción, sección “Persistencia”).

4. `CupoMaximoException`

Es la clase de excepción que se lanza cuando se intenta agregar un escenario pero se ha alcanzado el límite de escenarios en el festival, o se intenta agregar una banda a un escenario pero este no posee espacio o presupuesto suficiente. El mensaje asociado a la excepción indica si el límite se alcanzó en el festival o en uno de los escenarios, y si el límite se alcanzó por cupos o por presupuesto, para la generación de este mensaje debe hacer uso del método `generarMensaje()` (Ver modelo conceptual). Esta información debe quedar consolidada en los atributos de la clase, utilizando las constantes de la



misma (Ver modelo conceptual). Al crear la excepción, se registra el error en el archivo de log con la fecha y hora de ocurrencia (ver documento de descripción, sección “Persistencia”).

Escritura en el Archivo de Log

Dado que todas las excepciones se registran en el mismo log debe tenerse en cuenta la forma para extender un archivo de texto ya creado. Esto se hace enviando por parámetro al constructor del objeto `PrintWriter` un objeto `BufferedWriter`, que a la vez en su constructor por parámetro recibe un objeto de tipo `FileWriter`. Para que permita continuar el archivo de texto en vez de reemplazarlo, al objeto `FileWriter` se le envía por parámetro la ruta del archivo y **true**, para indicar que se desea anexar y no reescribir. De esta forma, siempre que se escriba en el archivo las líneas quedarán anexadas al final.

Manejo de fechas

Todas las fechas, tanto en el reporte como en el log de excepciones, deben tener el siguiente formato: <día de la semana> <mes> <día> <hora> COT <año> (ej: Thu Aug 17 14:45:29 COT 2017).

Para poder crear un `String` de la fecha con esta formato debe utilizar el método `toString`, como se muestra a continuación:

```
String cadenaFecha = fecha.toString();
```