

 Proyecto Cupí2	ISIS-1205 Algorítmica y Programación II Consideraciones adicionales de diseño
Ejercicio:	n8_CentroDeConvenciones
Autor:	Equipo Cupí2
Semestre:	2017-1

Manejo de excepciones propias

Para tratar los errores específicos que se presenten 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` y `RecetaTieneIngredienteException`.

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. El mensaje asociado con la excepción describe el problema que se presentó.

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.

3. `EspacioTieneEventoFechaException`

Es la clase de excepción que se lanza cuando se intenta agregar un evento en donde, para el mismo espacio, hay otro evento en la fecha dada. El mensaje asociado a la excepción indica el nombre del espacio y el nombre del evento que se intentó agregar. Al crear la excepción, se registra el error en un archivo de log (ver documento de descripción, sección "Persistencia"). Esta clase posee constantes, atributos y métodos propios, los cuales se describen a continuación.

Se declara una constante de tipo `String` para guardar el nombre del archivo de log (registro de errores de elementos repetidos). Esta es: `LOG_FILE = "./data/error.log"`.

El atributo utilizado es el nombre del espacio, y el nombre del evento que se intentó agregar.

El método constructor, además de crear la excepción (con la instrucción `super`), debe registrar el error en el archivo de log.

