

Asignación 2

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computación
Aseguramiento de la Calidad del Software
Prof. Ignacio Trejos Zelaya

Franco Quiros, Carnet 2013029890

Bryan Mena, Carnet 2016112933

Pablo Brenes, Carnet 2016250460

1 EL CALENDARIO GREGORIANO

Es de origen europeo, llegó a sustituir el calendario Juliano en 1582, incorporó los años bisiestos al agregar un día extra a febrero en ciertas fechas. El propósito de su implementación era eliminar el desfase que hacía que la Pascua no se celebrara en el momento correcto. Un error de decimales en el cálculo del calendario Juliano hizo que entre los años 325 y 1582 se acumulara un error de aproximadamente 10 días, para evitar este desfase en el calendario gregoriano se establecieron una serie de reglas que contrarrestaban el efecto del redondeo:

- Cada 4 años es bisiesto, se agrega un día a Febrero
- Si el año es múltiplo de 100 no es bisiesto
- Si el año es múltiplo de 400, es bisiesto

El desfase antes mencionado se contrarrestó cambiando la fecha 5 de Octubre de 1582 por 15 de Octubre de 1582.

2 RELACIÓN ENTRE LA DEFUNCIÓN DE CERVANTES Y SHAKESPEARE

El día 23 de abril de cada año se celebra el Día del Libro en conmemoración a la muerte de William Shakespeare y Miguel de Cervantes en 1616, sin embargo las muertes no sucedieron el mismo día, existía un desfase entre el calendario Inglés y el Español, para los ingleses Shakespeare murió el 23 de abril pero para los españoles murió el 3 de mayo, esto dado que Inglaterra adoptó el calendario Gregoriano hasta 1752 y España lo hizo inmediatamente.

3 REQUERIMIENTOS FUNCIONALES

Para el desarrollo de esta asignación se siguen los requerimientos dados en la especificación de la Asignación 2, a estos requerimientos se les añadieron conceptos de validación de datos entre otros como se menciona a continuación:

- R0(fecha_es_tupla): Dado el uso del lenguaje de programación Java, el objetivo de este requerimiento pasa a ser la validación de la creación correcta de un objeto tipo Fechas a partir del argumento dado en la línea de comandos.

- R7 Validación de fechas: Como se menciona en el resumen de la investigación acerca del calendario gregoriano se requiere la validación de la excepción existente en el calendario (Salto del 4 de Octubre de 1582 al 15 de Octubre de 1582).
- R8 Validación del año: Se requiere que el año que se esta evaluando en funciones como *bisiesto* sea valido dentro del contexto del programa, esto es: año \geq 1582

4 TRABAJO FINAL

4.1 Decisiones de Diseño Tomadas

Para este proyecto es importante mencionar las siguientes decisiones de diseño tomadas por el equipo de trabajo:

- Se decide utilizar varias clases dentro del programa con el fin de mantener el principio de *Single Responsibility* beneficiando así la mantenibilidad del código
 - Clase Fecha: Manejo de fechas según el requerimiento R0 de la especificación, y el parseo de las mismas a partir de un argumento en la línea de comandos
 - Clase Mes: Almacenamiento de la información de un mes, su nombre, el calendario asociado a ese mes y el número de días que tiene el mes
 - Clase Utilitarias: Contiene las funciones mencionadas en la especificación de esta Asignación 2, además de otras funciones auxiliares
 - Enum Meses: Almacenar el par (nombreMes, numeroMes) para una búsqueda sencilla dentro de los algoritmos
- Velar por el principio de *Single Responsibility* creando una función *crearCalendario* con la lógica para crear un calendario y otra función *imprimir_4x3* que imprima el calendario deseado
- Uso de una matriz para almacenar los días del calendario de un mes en específico con el fin de fácil acceso y la fácil modificación de los datos
- Implementación de un menú por línea de comandos para la facilidad de uso del programa

4.2 Pruebas Realizadas

Para finalizar con el trabajo se realizaron una serie de pruebas para verificar los requerimientos funcionales dados en la especificación:

```

Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
fecha_es_tupla 2018 04 12
true
Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
fecha_es_tupla 18 04 12
false
Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
bisiesto 2004
true
Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
bisiesto 1997
false
Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
fecha_es_valida 1809 11 05
true
Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
fecha_es_valida 1809 13 32
false
Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
dia_siguiente 2004 02 29
(Año: 2004, Mes: 2, Día: 29)

Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
dia_siguiente 2005 02 29
(Año: 2005, Mes: 3, Día: 1)

Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
dia_desde_primero_enero 2018 09 09
251
Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
dia_primero_enero 2018
1
  
```

Fig. 1. Pruebas de varias de las funcionalidades del proyecto

```

Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
Uso del programa:
fecha_es_tupla aaaa mm dd
    -> Verifica si una fecha se puede representar en el programa
bisiesto aaaa
    -> Retorna verdadero o falso dependiendo si el año dado es bisiesto o no
fecha_es_valida aaaa mm dd
    -> Verifica que la fecha sea válida
dia_siguiente aaaa mm dd
    -> Retorna la fecha siguiente a la fecha dada
dia_desde_primero_enero aaaa mm dd
    -> Retorna la cantidad de días transcurridos entre el primero de enero del año dado a la fecha dada
dia_primero_enero aaaa
    -> Retorna el número del día en que cae el 1 de Enero del año dado
imprimir_dia aaaa
    -> Imprime el calendario en una matriz del año dado
  
```

Fig. 2. Muestra del comando de ayuda presente dentro del programa

Capitulum 480 1582

Calendario 1582

Enero							Febrero							Marzo								
D	L	K	M	J	V	S	D	L	K	M	J	V	S	D	L	K	M	J	V	S		
					1	2			1	2	3	4	5	6			1	2	3	4	5	6
3	4	5	6	7	8	9	7	8	9	10	11	12	13	7	8	9	10	11	12	13		
10	11	12	13	14	15	16	14	15	16	17	18	19	20	14	15	16	17	18	19	20		
17	18	19	20	21	22	23	21	22	23	24	25	26	27	21	22	23	24	25	26	27		
24	25	26	27	28	29	30	28							28	29	30	31					
31																						

Abril							Mayo							Junio								
D	L	K	M	J	V	S	D	L	K	M	J	V	S	D	L	K	M	J	V	S		
					1	2							1			1	2	3	4	5	6	
4	5	6	7	8	9	10	2	3	4	5	6	7	8	6	7	8	9	10	11	12		
11	12	13	14	15	16	17	9	10	11	12	13	14	15	13	14	15	16	17	18	19		
18	19	20	21	22	23	24	16	17	18	19	20	21	22	20	21	22	23	24	25	26		
25	26	27	28	29	30		23	24	25	26	27	28	29	27	28	29	30					
							30	31														

Julio							Agosto							Setiembre								
D	L	K	M	J	V	S	D	L	K	M	J	V	S	D	L	K	M	J	V	S		
					1	2			1	2	3	4	5	6	7			1	2	3	4	5
4	5	6	7	8	9	10	8	9	10	11	12	13	14	5	6	7	8	9	10	11		
11	12	13	14	15	16	17	15	16	17	18	19	20	21	12	13	14	15	16	17	18		
18	19	20	21	22	23	24	22	23	24	25	26	27	28	19	20	21	22	23	24	25		
25	26	27	28	29	30	31	29	30	31					26	27	28	29	30				

Octubre							Noviembre							Diciembre							
D	L	K	M	J	V	S	D	L	K	M	J	V	S	D	L	K	M	J	V	S	
					1	2						1	2	1	2	3	4	5	6	7	8
3	4	5	6	7	8	9	3	4	5	6	7	8	9	8	9	10	11	12	13	14	15
10	11	12	13	14	15	16	10	11	12	13	14	15	16	15	16	17	18	19	20	21	22
17	18	19	20	21	22	23	17	18	19	20	21	22	23	22	23	24	25	26	27	28	29
24	25	26	27	28	29	30	24	25	26	27	28	29	30	29	30	31					

Fig. 3. Calendario de 1582

Además de las pruebas antes mostradas se implementa dentro del programa una función para realizar pruebas basadas en un archivo de texto (Véase archivo test.txt incluido en la carpeta de Código del proyecto como ejemplo del formato del archivo), un ejemplo del comando mencionado se puede apreciar en la fig 4.

```

Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
probar_archivo test.txt
fecha_es_tupla 2018 04 12 -> true
fecha_es_tupla 2017 01 01 -> true
fecha_es_tupla 2016 08 31 -> true
fecha_es_tupla 2015 12 01 -> true
fecha_es_tupla 2024 11 30 -> true
fecha_es_tupla 1581 01 01 -> false
fecha_es_tupla 01 01 01 -> false
fecha_es_tupla 2018 19 24 -> false
fecha_es_tupla 2018 01 32 -> false
fecha_es_tupla 2018 01 -1 -> false
bisiesto 1600 -> true
bisiesto 2004 -> true
bisiesto 2008 -> true
bisiesto 2010 -> false
bisiesto 1990 -> false
bisiesto 1197 -> false
fecha_es_valida 2018 11 11 -> true
fecha_es_valida 1998 08 25 -> true
fecha_es_valida 1967 10 20 -> true
fecha_es_valida 1700 01 01 -> true
fecha_es_valida 0000 01 01 -> false
fecha_es_valida 2018 99 01 -> false
fecha_es_valida 2018 01 99 -> false
dia_siguiente 2018 01 01 -> (Año: 2018, Mes: 1, Día: 2)
  
```

Fig. 4. Uso del comando *probar_archivo*

4.3 Diagrama de Clases

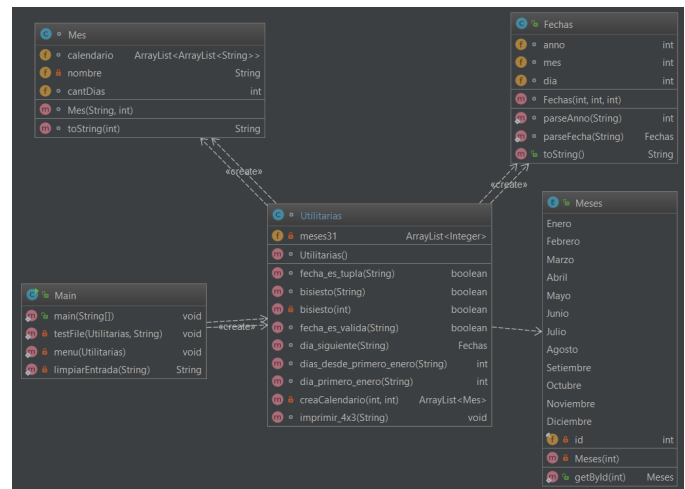


Fig. 5. Diagrama de Clases del proyecto finalizado con sus atributos y métodos

5 INSTRUCCIONES DE USO

Para el uso del programa se tienen los siguientes comandos:

- **-h**: Comando de ayuda muestra una pequeña guía con las funcionalidades del programa
- **fecha_es_tupla aaaa mm dd**: Verificar si una fecha se puede representar en el programa
- **bisiesto aaaa**: Retorna verdadero o falso dependiendo si el año dado es bisiesto o no
- **fecha_es_valida aaaa mm dd**: Verifica que la fecha sea válida

- *dia_siguiente aaaa mm dd*: Retorna la fecha siguiente a la fecha dada
- *dias_desde_primero_enero aaaa mm dd*: Retorna la cantidad de días transcurridos entre el primero de enero del año dado a la fecha dada
- *dia_primero_enero aaaa*: Retorna el número del día en que cae el 1 de Enero del año dado
- *imprimir_ 4x3 aaaa*: Imprime el calendario en una matriz 4x3 del año dado
- *probar_archivo nombre*: Ejecuta cada comando por línea en el archivo de prueba (bajo el mismo formato de los comandos) (Vease *test.txt* incluido en la carpeta de código como un ejemplo del formato para el archivo de pruebas)
- *salir*: Termina la ejecución del programa

6 ANÁLISIS DE RESULTADOS

Al final del desarrollo de este proyecto terminamos con una pieza de software que se considera ideal y se adapta a los requerimientos funcionales dados en la especificación original del proyecto, a pesar de esto dado el lenguaje de implementación (java) y la manera en que se programó la función *fecha_es_valida* esta admite valores de punto flotante, aunque a la hora de convertirlo a un objeto tipo fecha, el punto flotante se ignora como se puede ver en la fig 6.

```

Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa
Fecha en punto: 2018.10.10
Fecha Resultante: (Año: 2018, Mes: 10, Día: 10)

true
Inserte un comando, para ayuda digite "-h", digite "salir" para salir del programa

```

Fig. 6. Parsing de una fecha con decimales y su resultado