

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

IC6831 - Aseguramiento de la Calidad del Software
Grupo 40

Prof. Ericka Solano

A4 – Producción de código: Calendario Gregoriano

Estudiantes
Roger Villalobos
Samantha Arbuola
Jose Pablo Murillo

Febrero 2018, San José

Producción de código según requerimientos

1. Dominio: el calendario gregoriano

¿Qué es el calendario juliano?

Llamado así por Julio César en el año 47 antes de Cristo, es el punto de referencia romano. Era el calendario predominante en el mundo romano, y posteriormente en la mayor parte de Europa y en los asentamientos europeos en América y otros lugares, hasta que fue sustituido progresivamente por el más preciso calendario gregoriano. Se adelanta con respecto al año trópico aproximadamente un día cada 128 años lo que equivale a 11 minutos 14 segundos de exceso por año

¿Qué es el calendario gregoriano?

Para darle origen al calendario gregoriano existieron dos estudios 1515 y 1578, pero no fue hasta este segundo que se hizo caso al desfase de días que tenían en el calendario de esa época y buscar una forma de corregirlo.

Para poder conseguir esto, el papa Gregorio XIII hizo que desaparecieran 10 días que se habían acumulado desde que se instaurase el calendario juliano. Más concretamente los días del 5 al 14 de octubre de 1582. El descubrimiento de este desfase llegó desde la Universidad de Salamanca, donde se realizaron varios estudios sobre el paso de los días.

La unidad principal de tiempo en el calendario gregoriano es el día. Hay 365 días en el calendario gregoriano. Una semana consta de 7 días.

En cuanto a los meses, pueden ser períodos de 30 o 31 días, salvo Febrero que puede tener 28 días en un año común y 29 en un año bisiesto. Los meses con 30 días son: abril, junio, septiembre y noviembre. Los meses con 31 días son: enero, marzo, mayo, julio, agosto, octubre y diciembre.

¿Cuándo fue instituido el calendario gregoriano?

Vino a sustituir en 1582 d.c al calendario juliano, utilizado desde que Julio César lo instaura en el año 46 antes de Cristo

¿Cuáles deficiencias o limitaciones del calendario juliano viene a subsanar el calendario gregoriano?

Entre los años 325 y 1582, antes de la reforma gregoriana, se había ido acumulando un desfase en el calendario debido a un cálculo inexacto del número de días del año trópico. El calendario juliano establecía un año bisiesto cada cuatro años, contando que el año tenía 365,25 días, en lugar de los 365,242189 que tiene en realidad. A consecuencia de este desfase de más de 11 minutos por año, en 1582 el calendario civil llevaba un retraso de 10 días con respecto al calendario astronómico.

Determinar la relación entre las fechas en que murieron los escritores Miguel de Cervantes y William Shakespeare

El 23 de abril de 1616 muere Miguel de Cervantes en Madrid y William Shakespeare en Statford-upon-Avon; el primero una prominente figura de la literatura inglesa y uno de los grandes novelistas en la historia de la literatura mundial, y el segundo considerado el más brillante autor inglés y el dramaturgo más influyente de la historia.

El 23 de abril se celebra el Día del Libro en conmemoración de la muerte de ambos.

En 1582, el Papa Gregorio XII adelantó 10 días el calendario, de modo que el día 5 de octubre pasó a ser el 15 del mismo mes. Sólo Francia, Italia y España adoptaron inmediatamente este cambio. Inglaterra no lo hizo hasta 1752. Así, para los ingleses, Shakespeare murió el 23 de abril, pero para los países católicos ocurrió el 3 de mayo.

2. Requerimientos funcionales generales

Con las propiedades del calendario gregoriano que Ud. investigó y el supuesto de que las fechas por tratar siempre están en una fecha igual o posterior a su entrada en vigencia en Roma, transforme los siguientes requerimientos generales en requerimientos funcionales más precisos. Todos los requerimientos tienen como contexto el dominio de las fechas en el calendario gregoriano.

R0: Todas las fechas serán presentadas como tuplas de tres números enteros positivos (ternas), en este orden: (año, mes, día).

R1: (bisiesto): Dado un año perteneciente al rango permitido, determinar si este es bisiesto. El resultado debe ser un valor booleano, True o False.

R2: (fecha_es_valida): Dada una fecha, determinar si ésta es válida. El resultado debe ser un valor booleano, True o False.

R3: (dia_siguiente): Dada una fecha válida, determinar la fecha del día siguiente. El resultado debe ser una fecha válida (tupla de tres números enteros positivos que

corresponde a una fecha en el Calendario gregoriano.

R4: (días_desde_primero_enero): Dada una fecha válida, determinar el número de días transcurridos desde el primero de enero de su año (el número de días transcurridos entre el primero de enero y el primero de enero, dentro de un mismo año, es 0). El resultado debe ser un número entero.

R5: (dia_primero_enero): Dado un año perteneciente al rango permitido, determinar el día de la semana que le corresponde, con la siguiente codificación: 0 = domingo, 1 = lunes, 2 = martes, 3 = miércoles, 4 = jueves, 5 = viernes, 6 = sábado. El resultado debe ser un número entero, conforme a la codificación indicada.

3. Restricciones no funcionales y técnicas

- Diseñe y programe todas las funciones que implementan los requerimientos funcionales del apartado anterior. No es válido reutilizar funciones o métodos de bibliotecas. Toda programación por entregar debe ser producto de su propio trabajo. Todo cálculo y lógica del programa debe ser explicado, sea en comentarios o en documentación externa al programa.
- Ud. tiene libertad para extender los requerimientos en cuanto a la detección de casos de error y la manera en que estos serán señalados vía programación.
- No nos interesa construir ni evaluar una interfaz de usuario amigable (en este momento).
- Resuelva todos los requerimientos en el lenguaje de programación Python (versión 3.4 o superior).
- Construya su programa de manera que el código sea legible.
- Pueden trabajar en grupos de hasta tres (3) miembros comunicando de antemano la conformación del grupo para su respectiva creación en la plataforma virtual.
- Su solución debe ser entregada en una carpeta comprimida, que comprenda: portada (que identifique a los miembros del grupo), requerimientos funcionales, resumen de hallazgos respecto del calendario gregoriano, decisiones de diseño tomadas, código fuente de su solución, evidencias de las pruebas realizadas, análisis de resultados obtenidos. El código fuente debe estar en un solo archivo aparte, dentro de la misma carpeta comprimida.

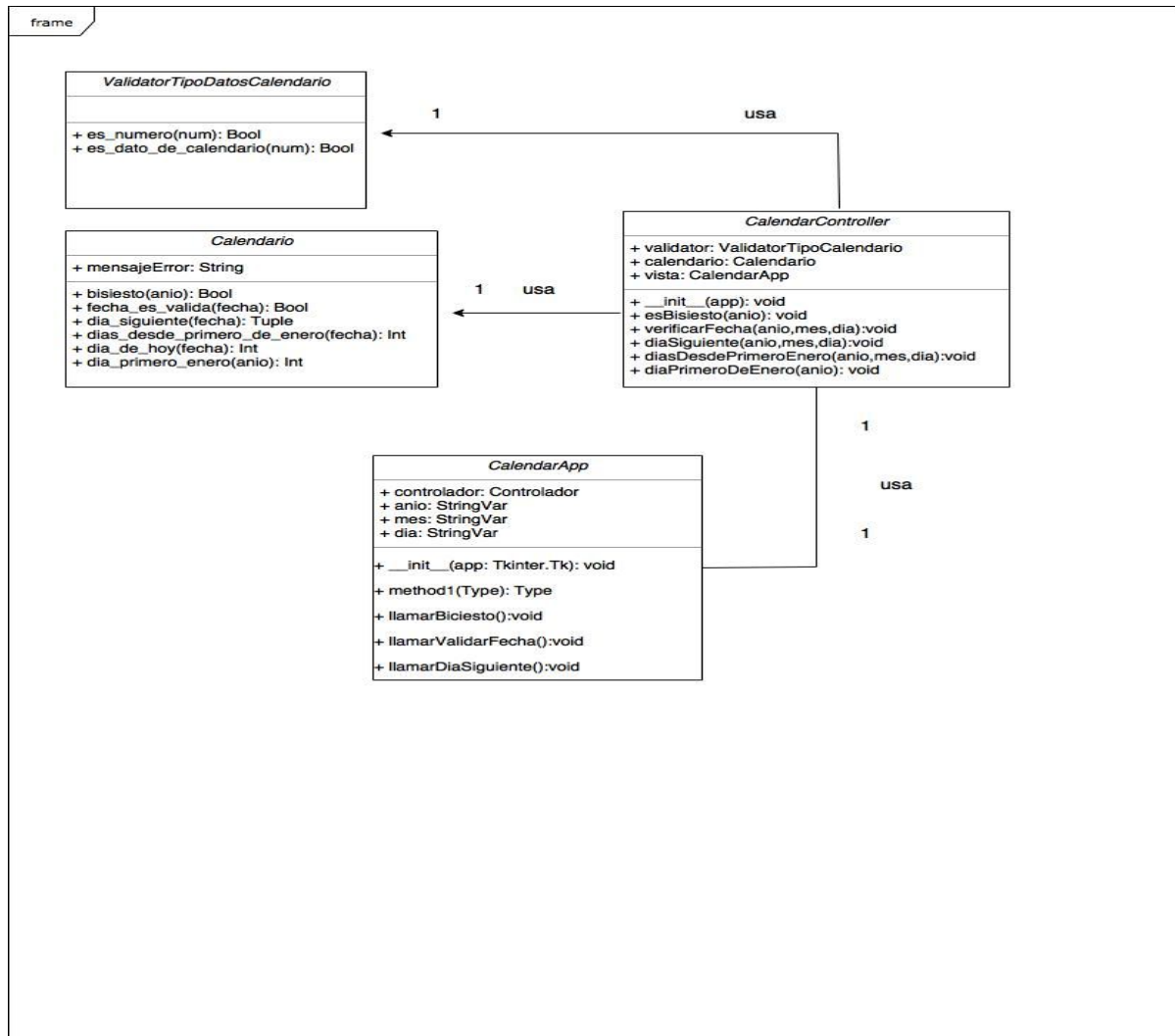
Decisiones de diseño

A. Manejo de fecha

Para la elaboración de este programa se decidió que se manejarían tuplas numéricas. El uso de datos numéricos ayuda a que los requerimientos funcionales se enfoquen solo en el proceso y validación de datos numéricos. La validación y conversión de datos que se reciben de la interfaz como hileras son tareas que va a realizar el controlador y validator respectivamente. Al grupo le parecía dejar la validación y operación de datos en entidades distintas para así asignar una sola responsabilidad a la clase. Asignar una sola responsabilidad ayuda a tener un código mantenible y más fácil de detectar errores.

Los datos que retornan los métodos funcionales retornan tuplas o un entero como es especificado en la descripción de la asignación. Sin embargo nos parecía más agradable poder presentar los resultados como un mensaje que muestra la pantalla porque esto hace el resultado más entendible y así se pueden mostrar errores más exactos.

B. Diagrama de clases:



Se decidió usar el paradigma Orientado a Objetos ya que esto ayuda a que solo una clase se encargue de hacer tareas específicas y el requerimiento funcional sea alcanzado a través de la comunicación entre clases. Además se pueden usar las clases **ValidatorTipoDatosCalendario** y **Calendario** para otros entornos ya que solo contienen la lógica; las clases **CalendarController** y **CalendarApp** si están ligadas al entorno específico de Python. A continuación se da un breve resumen de lo que hace cada clase:

ValidatorTipoDatosCalendario: Se encarga de validar el tipo de datos de entrada y que sea un dato válido de un calendario (mes, año o día)

Calendario: Se encarga de realizar consultas y cálculos respectivos a fechas de un calendario

CalendarController: Clase que se encarga de usar la clase ValidatorTipoDatosCalendario, Calendario y CalendarApp para hacer los requerimientos funcionales especificados y mostrar en pantalla los resultados correspondientes.

CalendarApp: Clase tipo GUI que se encarga de capturar los datos de entrada y llamar los métodos de la clase controlador para realizar la operación.

C. Uso de interfaz

Se decidió hacer un uso de interfaz para poder facilitar las pruebas de los requerimientos funcionales. Sin embargo no se tomó en cuenta el diseño de la interfaz orientado a ser detallada y atractiva sino funcional. Esta decisión fue tomada tomando en consideración el tiempo en el que se disponía para realizar la aplicación.

La interfaz consta de una serie de botones para que el usuario pueda escoger la operación a realizar. Cada botón toma los datos que requiere de los campos de texto y luego muestra un mensaje de resultado o error debajo de estos.



Pruebas realizadas

Aquí se muestra un listado de pruebas realizadas por los desarrolladores para los requerimientos funcionales

Desarrollador	Requerimiento funcional	Datos usados	Resultado esperado	Resultado actual
José Pablo Murillo Máquina: Macbook OSX 10.12	R1: (bisiesto)	"virus"	"Debe ingresar un número entero"	"Debe ingresar un número entero"
		2014	"No es bisiesto"	"No es bisiesto"
		2004	"Si es bisiesto"	"Si es bisiesto"
	R2: (fecha_es_valida)	(2014,2,30)	"Fecha ingresada no válida: febrero tiene 28 días en un año no bisiesto"	"Fecha ingresada no válida: febrero tiene 28 días en un año no bisiesto"
		(2014,2,5)	"Fecha ingresada es valida"	"Fecha ingresada es valida"
		(-2000,null,null)	"Debe ingresar un año entero"	"Debe ingresar un año entero"
	R3: (dia_siguiente)	(2018,12,31)	"Dia siguiente: 1 del mes 1 del 2019"	"Dia siguiente: 1 del mes 1 del 2019"
		(2018,12,null)	"Debe ingresar un día entero entre 1 y 31"	"Debe ingresar un día entero entre 1 y 31"
		(2014,2,28)	"Dia siguiente: 1 del mes 3 del 2014"	"Dia siguiente: 1 del mes 3 del 2014"
	R4: (días_desde_primero_enero):	(2018,1,1)	"Han pasado 0 días desde el 1ero de enero de 2018"	"Han pasado 0 días desde el 1ero de enero de 2018"

		(2018,1,15)	"Han pasado 14 días desde el 1ero de enero de 2018"	"Han pasado 14 días desde el 1ero de enero de 2018"
		(2018,2,30)	"Febrero tiene 28 días en un año no bisiesto"	"Febrero tiene 28 días en un año no bisiesto"
	R5: (dia_primer_enero)	2018	"Día 1: Lunes"	"Día 1: Lunes"
		2017	"Día 0: Domingo"	"Día 0: Domingo"
		"maldad"	"Debe ingresar un año entero"	"Debe ingresar un año entero"
Samantha Arburola Máquina: Windows 10	R1: (bisiesto)	1389	"Debe ingresar un año mayor a 1582"	"Debe ingresar un año mayor a 1582"
		1640	"Si es bisiesto"	"Si es bisiesto"
		-1994	"Debe ingresar un año mayor a 1582"	"Debe ingresar un número entero"
	R2: (fecha_es_valida)	(623, 12,3)	"Debe ingresar un año mayor a 1582"	"Debe ingresar un año mayor a 1582"
		(1650, 13,1)	"Fecha ingresada no válida: Mes inválido"	"Fecha ingresada no válida: Día y Mes inválido"
		(1789, 7,32)	"Debe ingresar un día entero entre 1 y 31"	"Debe ingresar un día entero entre 1 y 31"
	R3: (dia_siguiente)	(2016,2,28)	"Día siguiente: 29 del mes 2 del 2016"	"Día siguiente: 29 del mes 2 del 2016"
		(1813, 12,31)	"Día siguiente: 1 del mes 1 del 1814"	"Día siguiente: 1 del mes 1 del 1814"
		(1994, 7, 30)	"Día siguiente:	"Día siguiente:

			31 del mes 7 del 1994"	31 del mes 7 del 1994"
	R4: (días_desde_primero_enero):	(1643,8,15)	"Han pasado 226 días desde el 1ero de enero de 1643"	"Han pasado 226 días desde el 1ero de enero de 1643"
		(1807,9,10)	"Han pasado 252 días desde el 1ero de enero de 1807"	"Han pasado 252 días desde el 1ero de enero de 1807"
		(1990,4,24)	"Han pasado 113 días desde el 1ero de enero de 1990"	"Han pasado 113 días desde el 1ero de enero de 1990"
	R5: (dia_primero_enero)	1640	"Día 0: Domingo"	"Día 0: Domingo"
		1582	"Día 5: Viernes"	"Día 5: Viernes"
		2000	"Día 6: Sábado"	"Día 6: Sábado"
Roger Villalobos Máquina: Linux Ubuntu 16.04	R1: (bisiesto)	2018	"No es bisiesto"	"No es bisiesto"
		2020	"Si es bisiesto"	"Si es bisiesto"
		4002020	"Si es bisiesto"	"Si es bisiesto"
	R2: (fecha_es_valida)	2018,2,25	"Fecha ingresada es válida"	"Fecha ingresada es válida"
		2018,6,31	"Fecha ingresada no válida: Dia y mes inválido"	"Fecha ingresada no válida: Dia y mes inválido"
		2018,31,6	"Fecha ingresada no válida: Dia y mes inválido"	"Fecha ingresada no válida: Dia y mes inválido"
	R3: (dia_siguiente)	2018,2,28	"Día siguiente: 1 del mes 3 del 2018"	"Día siguiente: 1 del mes 3 del 2018"

		2020,2,28	"Día siguiente: 29 del mes 2 del 2018	"Día siguiente: 29 del mes 2 del 2018
		2018,2,29	"Febrero tiene 28 días en un año no bisiesto"	"Febrero tiene 28 días en un año no bisiesto"
	R4: (días_desde_pri mero_enero):	2018,2,25	"Han pasado 55 días desde el 1ero de enero de ese año"	"Han pasado 55 días desde el 1ero de enero de ese año"
		2018,2,29	"Febrero tiene 28 días en un año no bisiesto"	"Febrero tiene 28 días en un año no bisiesto"
		2020,12,31	"Han pasado 365 días desde el 1ero de enero de ese año"	"Han pasado 365 días desde el 1ero de enero de ese año"
	R5: (día_primero_en ero)	2020	"Día 3: Miércoles"	"Día 3: Miércoles"
		1995	"Día 0: Domingo"	"Día 0: Domingo"
		4002020	"Día 3: Miércoles"	"Día 3: Miércoles"

Análisis de resultados obtenidos:

De acuerdo a las pruebas se puede evidenciar que el programa tiene correctitud en cuanto a los procesos y también es robusto ya que maneja algunas excepciones de acuerdo a el formato de fechas y el formato de los datos como tal. También hay rasgos de portabilidad ya que no hubo ningún problema al correr el mismo programa en 3 ambientes diferentes.

Referencias

¿Shakespeare y Cervantes murieron el mismo día?. (2018). MuyHistoria.es.

Revisado 24 de febrero de 2018, desde

<https://www.muyhistoria.es/curiosidades/preguntas-respuestas/ishakespeare-y-cervantes-murieron-el-mismo-dia>

ALMANAQUES Y CALENDARIOS - CALENDARIO JULIANO. (2018).

Elalmanaque.com. Revisado 24 de febrero de 2018, desde

<http://www.elalmanaque.com/Calendarios/juliano.htm>

arrizabalaga, m. (2018). *Los diez días que no existieron.* ABC.es. Revisado 24 de febrero de 2018, desde

<http://www.abc.es/20121006/archivo/abci-diez-dias-existieron-201210031518.html>

Diferencia entre el calendario juliano y el gregoriano. (2018).

Historyaybiografias.com. Revisado 24 de febrero de 2018, desde

<https://historyaybiografias.com/calendario/>

Calendario gregoriano. (n.d.). accesado Febrero 25, 2018, desde

https://www.ecured.cu/Calendario_gregoriano#D.C3.ADa.2C_semana.2C_mes