



Universidad de Costa Rica  
Facultad de Ingeniería  
Escuela de Ingeniería Eléctrica  
**IE-0117 Programación Bajo Plataformas Abiertas**



Ing. Marco Villalta Fallas - I Ciclo 2018

---

Laboratorio 9  
Programación Orientada a Objetos: Python

---

**Instrucciones Generales:**

Los laboratorios se deben de realizar de manera individual.

El laboratorio debe de entregarse a más tardar el 21 de junio antes de las 23:59.

Entregue un archivo comprimido que incluya un directorio llamado **informe** con los archivos necesarios para generar el PDF del informe (.tex, imágenes, código, entre otros) y un directorio llamado **src** con los archivos de código fuente que lleven a la solución. En la calificación de este laboratorio se tomará en cuenta el uso del control de versiones git y la documentación en Doxygen.

## 1. Enunciado

El siguiente laboratorio tiene como objetivo comprobar los conocimientos adquiridos durante el curso. Específicamente se evaluará el manejo del lenguaje Python, en su aplicación a la programación orientada a objetos (POO). Los temas a evaluar son: manejo de conceptos de la POO como herencia y polimorfismo, correcto manejo de errores, documentación y por supuesto la funcionalidad del código. Para ello se presenta como ejercicio académico la implementación de un clase **Fecha**. Esta clase hereda de dos clases: **Reloj** y **Calendario**, las cuales a su vez está compuestas de otras más simples. A continuación se expone que debe contener cada una de las clases.

**Nota:** En caso de que no se cumpla alguna de las instrucciones generales, el laboratorio tendrá una nota final de cero.

## 2. Clases

Las clases que debe de implementar se muestran en el diagrama UML de la Figura 1. Todos los atributos de las clases deben de poder asignarse desde el método constructor, ser privados, y estar disponibles a través de métodos del tipo **set** y **get**. Además se proporciona la descripción de los que debe de realizar cada uno de los métodos en el caso que su implementación no sea totalmente trivial.

### 2.1. Descripción de las clases y sus funciones

#### 2.1.1. Clase Contador

Esta clase debe de proporcionar una funcionalidad de cuenta con rebase. Esto quiere decir que debe tener la capacidad de aumentar un número en uno de manera indeterminada hasta que se alcance un valor máximo, en ese momento, la cuenta debe de volver a iniciar en cero. Esta clase le brindará la funcionalidad necesaria para que los segundos, minutos y horas aumenten y se reinicien de manera apropiada. A continuación se describen algunos de sus métodos:

- **contar:** esta función debe de aumentar en 1 el valor del atributo **cuenta** siempre que el valor actual de **cuenta** no sea igual al atributo **max**. En ese caso, la función debe de regresar a 0 la variable **cuenta** y poner en 1 el atributo **rebase**.
- **borrarRebase:** esta función debe de poner en 0 el atributo **rebase**.

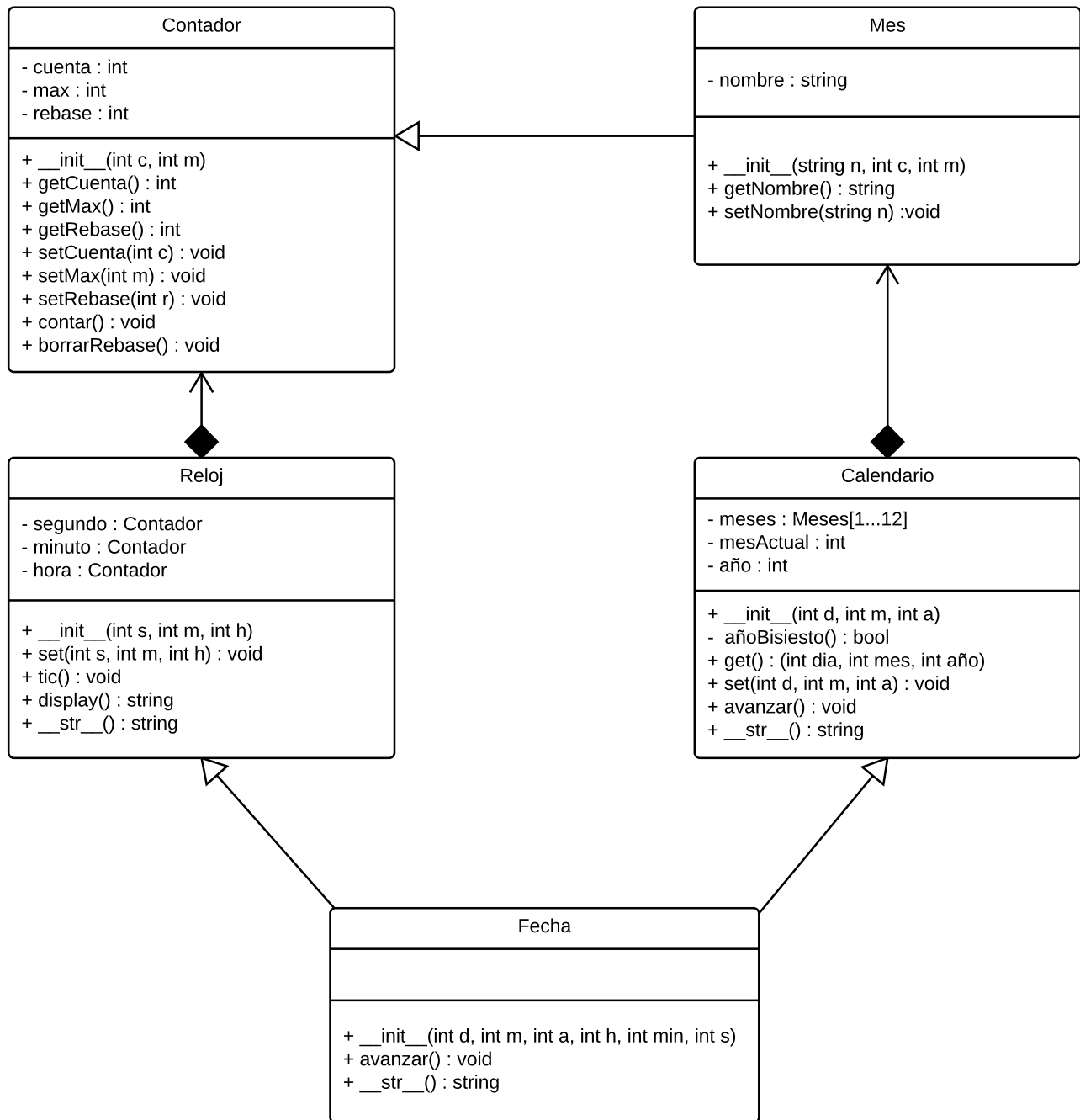


Figura 1: Diagrama UML

### 2.1.2. Clase Mes

Esta clase proporciona la utilidad para saber cuantos días tiene un mes particular, así como almacenar su nombre y día actual. También debe avisar mediante **rebase** cuando acaba el mes. Hereda de la clase **Contador**.

### 2.1.3. Clase Reloj

Clase compuesta. Tiene objetos de tipo **Contador** para llevar el control de como avanzan los segundos, minutos y horas. A continuación se describen algunos de sus métodos:

- **tic**: este método deberá realizar las operaciones necesarias para hacer que avance un segundo en el reloj.
- **display**: este método debe de regresar un string de la forma: "hh:mm:ss"
- **\_\_str\_\_**: este método retorna el mismo string que la función **display**

### 2.1.4. Clase Calendario

Clase compuesta. Tiene una lista de objetos tipo **Mes** para llevar el control del día actual del mes, y conocer de manera adecuada cuando se acaba el mes. A continuación se describen algunos de sus métodos:

- **añoBisiesto**: método privado utilizada para comprobar si el año actual es bisiesto. Se debe de llamar cada vez que el año cambia o cuando se crea el objeto para saber si el año actual es bisiesto o no, y así fijar la cantidad de días del mes de febrero. El estudiante deberá investigar y programar por su cuenta el algoritmo para determinar si es año bisiesto.
- **avanzar**: método que se utiliza para avanzar de manera adecuada al siguiente día.
- **\_\_str\_\_**: este método regresa un string de la forma: "dd/mm/aaaa'."

### 2.1.5. Clase Fecha

Esta clase hereda de Reloj y de Calendario. Aporta la funcionalidad de poder brindar una fecha completa: hora, minuto y segundo de un día, mes y año particular. Sus métodos se describen a continuación:

- **avanzar**: este método se llama para avanzar un segundo en la **Fecha** actual.
- **\_\_str\_\_**: Este método regresa un string de la forma: "dd de nombre\_del\_mes del aaaa, hh:mm:ss"

## 3. Comprobación de funcionalidad

Para comprobar la funcionalidad, haga empezar su objeto del tipo Fecha en el día "17 de noviembre del 2016, a las 19:00:00", y llame al método avanzar 1 000 000 veces. Con esto la fecha resultante que usted debe de obtener es: "29 de noviembre del 2016, 08:46:40"