



## Práctica 2. Clases y Objetos

### Objetivos

- Conocer la forma en que se crean nuevas clases en C++ y se instancian los objetos de dichas clases.
- Declarar, implementar y utilizar diferentes tipos de constructores en las clases.
- Declarar, implementar y utilizar funciones auxiliares como complemento para el correcto uso de las clases declaradas.
- Gestión de errores con excepciones que son clases.

### Índice

[Contexto de la práctica](#)

[Material de partida](#)

[Ejercicios](#)

### Contexto de la práctica

*Segismundo Barcetes es un joven de cuarenta-y-muy-pocos años de nuestra provincia. Sus amigos, si los tuviera, le llamarían Segis. Segismundo no es especialmente agraciado en lo relativo a sus relaciones con los demás. De hecho, es especialmente des-agraciado en sus relaciones con los demás. Nunca lo sabremos, pero quizá sea este el motivo por el que su vida social es prácticamente nula... al menos en lo que se refiere al mundo real, pues en los círculos virtuales de Internet dicha vida social es inexistente.*

*Sobra decir que Segismundo estudia el Grado en Ingeniería Informática. Sobra decir que sueña, como no podía ser de otra forma, con ser guapo, joven y rico como sus profesores de Programación Orientada a Objetos, a los cuales idolatra... como hacen el resto de sus compañeros. Por supuesto, este sentimiento no tiene nada que ver con el hecho de que sean ellos los que lo van a examinar. Como él suele decir: “esta última circunstancia es meramente*



*circunstancial”, o también: “esta última coyuntura es meramente coyuntural”, sin olvidar: “este último hecho es meramente hechal”.*

*¿Qué decir de las aficiones del joven Barcetes? Que no tiene aficiones, solo afición: Segismundo desea fervientemente ser un famoso DJ y está convencido de que algún día Avicii y David Guetta se enzarzarán en una acalorada discusión a través de twitter para ver cuál de los dos consigue ser su telonero. Por eso, cada fin de semana (una vez que ha terminado todos los ejercicios de las relaciones de ídem, amén de los cuadernillos y guiones de prácticas), nuestro protagonista cambia su nombre por otro mucho más cañero (DJ Segismundo Barcetes Rodríguez), coge sus bártulos y se desplaza a los garitos habituales en los que, durante horas, solo vive por y para su público. Horas que, no es necesario decir, se hacen interminables... para su público.*

*Por motivos que serían largos de contar, DJ Segismundo Barcetes Rodríguez solo tiene permitido actuar por el momento en dos garitos: el bar de su padre, Felipe Sánchez (esto también sería largo de contar) y el hogar del pensionista de su pueblo. Por cierto que el pleno del ayuntamiento ya aprobó hace tiempo la realización de un estudio estadístico para determinar si existía relación entre la fecha en que DJ Segismundo Barcetes Rodríguez comenzó a actuar en este último lugar y la fecha en la que empezaron a multiplicarse las funerarias en el municipio. Si bien el presupuesto (algo más de 150.000 euros) ya está aprobado, un hecho puntual retrasa la realización de este estudio. Concretamente el hecho puntual de que la hija del concejal de urbanismo solo haya aprobado 3 asignaturas de la Diplomatura en Estadística que empezó hace 15 años.*

*Nuestro protagonista está convencido de que solo utilizando tecnología de última generación podrá conseguir las altas cotas de éxito que tanto anhela. Es por este motivo que ha decidido hacer un programa en C++ que se ejecute desde línea de órdenes con el cual controle cada una de sus actuaciones. De este modo, podrá disponer de información actualizada acerca de qué temas ha ido utilizando en cada sesión y en qué medida va respondiendo su público. Un análisis automatizado de los datos recopilados le permitirá ir confeccionando su personal lista de temazos infalibles. Además, ha decidido que posiblemente lo comercialice, por lo que ya ha reservado el dominio:*

*[www.el-programa-que-utiliza-dj-segismundo-barcetes-rodriguez-para-almacenar-la-informacion-de-cada-una-de-sus-sesiones-cuando-va-por-los-garitos-cada-fin-de-semana-una-vez-que-ha-terminado-todos-los-ejercicios-de-las-relaciones-de-ejercicios-amen-de-los-cuadernillos-y-guiones-de-practicas.org](http://www.el-programa-que-utiliza-dj-segismundo-barcetes-rodriguez-para-almacenar-la-informacion-de-cada-una-de-sus-sesiones-cuando-va-por-los-garitos-cada-fin-de-semana-una-vez-que-ha-terminado-todos-los-ejercicios-de-las-relaciones-de-ejercicios-amen-de-los-cuadernillos-y-guiones-de-practicas.org).*

*Por desgracia, el .com ya estaba pillado.*

*Durante las próximas sesiones, desarrollaremos este programa de gestión de información sobre sesiones de DJ poniéndonos en la piel del muchacho Barcetes.*

*Bien pensado, con entender lo que necesita posiblemente sea más que suficiente.*

## Material de partida

- En esta práctica y la siguiente trabajaremos la definición de clases e instanciación de objetos.
- Al ser la primera sesión del bloque, no existe material de partida.

## Ejercicios

1. Define la clase *Temazo* que tiene los siguientes atributos:

- *Título*: cadena de caracteres
- *Intérprete*: cadena de caracteres
- *Duración en segundos*: entero
- *Puntuación del público*: entero

Y los siguientes métodos:

- Constructor por defecto
- Constructor parametrizado con título, intérprete y duración
- Constructor de copia
- Destructor
- *Getters* y *setters*... salvo *setPuntuacion*, que no existirá.

2. Define la clase *Garito* con los atributos:

- *Nombre*: cadena de caracteres
- *Dirección*: cadena de caracteres

Siendo sus métodos:

- Constructor por defecto
- Constructor parametrizado
- Constructor de copia
- Destructor
- *Getters* y *setters* para los dos atributos.

3. Define la clase *Fecha*, cuyos atributos son:

- *Día, mes y año*: de tipo entero

Y sus métodos:

- Constructor por defecto que asigna la fecha actual<sup>1</sup> al objeto
- Constructor parametrizado
- Constructor de copia

---

<sup>1</sup> <http://stackoverflow.com/questions/997946/how-to-get-current-time-and-date-in-c>

- *Getters y setters*

**4.** Crea la clase *ParametroNoValido*. Esta clase nos servirá para lanzar excepciones de este tipo en los ejercicios que vendrán a continuación. La clase debe tener los siguientes atributos:

- *Origen*: cadena de caracteres. Nombre de la función o clase::método desde el que se realiza el throw.
- *Descripción*: cadena de caracteres. Una descripción explicando qué parámetro no es válido y por qué

Además del constructor parametrizado, añadir a la clase un método llamado *queOcurre()*, que devuelva una cadena de caracteres concatenando los atributos *Origen* y *Descripción*, e.g. "[Garito::setNombre]: el nuevo nombre no puede ser una cadena vacía"

**5.** Modifica los setters de la clase Garito, de forma que se lance una excepción de tipo *ParametroNoValido* con un mensaje de texto (*std::string*) si el valor que se intenta asignar al atributo correspondiente es una cadena vacía

**6.** Crea un módulo de funciones auxiliares llamado *djutils*, que contenga:

- una función para mostrar por pantalla los datos de un Temazo,
- una función para mostrar por pantalla los datos de un Garito,
- una función para mostrar por pantalla los datos de una Fecha
- una función que reciba como parámetro (por referencia) un objeto de clase Garito; dentro de la función se pedirá al usuario que escriba un nombre y una dirección. La función debe modificar el objeto recibido por referencia, asignándole los nuevos datos válidos aportados por el usuario. Incluir en la especificación de la función las excepciones que puede lanzar si algún dato proporcionado para el garito no es válido.

**7.** Crea una función *main* (en el fichero *main.cpp*) en la que:

- Se instancien:
  - 2 Temazos: uno usando el constructor por defecto y otro usando el parametrizado
  - 2 Garitos: uno usando el constructor parametrizado y otro usando el constructor de copia
  - 2 Fechas: una usando el constructor por defecto y otra el constructor parametrizado.
- A continuación, usando las funciones del módulo auxiliar, escribir los datos de los Temazos, los Garitos y las Fechas en pantalla.
- Finalmente, modificar los datos del 2º Garito (el que hemos creado por copia) y luego escribir en pantalla los nuevos datos del mismo, usando para ello las funciones declaradas en el módulo de funciones auxiliares. Capturar las excepciones que se puedan producir durante la modificación de los datos y mostrar su información por la salida de error, *std::cerr*.

**8.** (Opcional) Crear un menú que permita al usuario decidir si quiere:

- Mostrar los datos de los Temazos.
- Mostrar los datos de los Garitos.

- c. Mostrar los datos de las Fechas.
- d. Modificar los datos del segundo Garito.
- e. Terminar el programa.

