



IIC2115 – Programación como Herramienta para la Ingeniería (I/2018)

## Actividad 2

### Objetivos

- Aplicar contenidos Programación Orientada a Objetos (POO).
- Utilizar ordenación jerárquica entre entidades.

### Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** GitHub
- **Hora:** 16:55
- **Desarrollo individual**

### Recomendación

Lean la actividad completa antes de comenzar a trabajar y fíjense en los sustantivos y verbos relacionados con el problema. En general, los sustantivos dan información de los objetos a modelar y los verbos relacionados con esos sustantivos se traducen en métodos de objetos.

Al terminar la clase (y no necesariamente la actividad), escriban un comentario al inicio de su código dando a conocer su opinión y/o su percepción de la actividad.

### Introducción

EL FIN SE ACERCA. Desesperado por el enorme meteorito que golpeará la tierra a las 16:55, tienes solo 2 horas y cuarenta minutos de vida, por lo que decides disfrutarlas al máximo y hacer realidad el sueño de todo estudiante universitario: Programar un grafo de computo que trabaje con Ints y Strings (que emoción).

## Descripción

Para esta actividad deberá modelar un *gráfo de cómputo*. Este gráfo funciona mediante nodos que realizan una cierta operación, en base a un input (que viene del output de otro nodo), para así entregar el resultado hacia el nodo siguiente en el grafo.

Todo nodo debe tener como mínimo un método llamado **operar**. Existen 3 tipos de nodos <sup>1</sup>:

- Nodo de strings: El input de este nodo es un **string** y realiza una operación sobre este, cuyo resultado es el output del nodo (también un **string**). Este nodo, a su vez, puede ser de 2 tipos, variando la operación que se realiza sobre el **string**:
  - Nodo que agrega una letra predeterminada al final del **string**.
  - Nodo que agrega una letra predeterminada al principio del **string**.
- Nodo de ints: El input de este nodo es un **int** y realiza una operación sobre este, cuyo resultado es un el output del nodo (también un **int**). Este nodo, a su vez, puede ser de 2 tipos, variando la operación que se realiza sobre el **int**:
  - Nodo que suma un valor predeterminado al int ingresado.
  - Nodo que multiplica por un valor predeterminado al int ingresado.
- Nodo híbrido: Este nodo puede tener como input a cualquiera de los dos tipos de datos anteriormente mencionados. Su operación varía en base a su input:
  - Si el input es un string → lo invierte.
  - Si el input es un int → cambia el orden de sus dígitos inversamente<sup>2</sup>, le agrega una constante predeterminada y lo convierte en **string**.

## Grafos

Con los nodos modelados anteriormente, deberán crear 3 grafos de cómputo distintos. Se define un grafo de cómputo como una red de conexiones entre distintos nodos. Los grafos a modelar son los siguientes:

1. Un grafo en donde se utilicen los caracteres **antane\_\_eprogramar** (en ese orden) para que el último output de la red sea **"me\_encanta\_programar"**.
2. Un grafo en donde se utilicen los números 5, 4, 20, 60, 3, 10, 2, 46, 1 (en ese orden) para que el último output de la red sea **"667"**
3. Un grafo en el que se utilicen los 2 grafos anteriores (con los mismos inputs utilizados) para que el output de la red sea **"767ramargorp\_atnacne\_em767"**

---

<sup>1</sup>hint: Se espera que al menos exista una clase nodo que **no pueda** ser instanciada

<sup>2</sup>Ejemplo: 123 → 321

## Hints

- Se recomienda primero realizar TODA la modelación del problema, con las clases respectivas y la jerarquía asociada a ellas (esto es lo que más asignación de puntaje tendrá).
- Luego de realizada la modelación, comenzar creando los grafos para probar que el modelo funcione correctamente.

## Política de Integridad Académica

Los alumnos de la Escuela de Ingeniería deben mantener un comportamiento acorde al Código de Honor de la Universidad:

*“Como miembro de la comunidad de la Pontificia Universidad Católica de Chile me comprometo a respetar los principios y normativas que la rigen. Asimismo, prometo actuar con rectitud y honestidad en las relaciones con los demás integrantes de la comunidad y en la realización de todo trabajo, particularmente en aquellas actividades vinculadas a la docencia, el aprendizaje y la creación, difusión y transferencia del conocimiento. Además, velaré por la integridad de las personas y cuidaré los bienes de la Universidad.”*

En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los alumnos que incurran en este tipo de acciones se exponen a un procedimiento sumario. Ejemplos de actos deshonestos son la copia, el uso de material o equipos no permitidos en las evaluaciones, el plagio, o la falsificación de identidad, entre otros. Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica en relación a copia y plagio: Todo trabajo presentado por un alumno (grupo) para los efectos de la evaluación de un curso debe ser hecho individualmente por el alumno (grupo), sin apoyo en material de terceros. Si un alumno (grupo) copia un trabajo, se le calificará con nota 1.0 en dicha evaluación y dependiendo de la gravedad de sus acciones podrá tener un 1.0 en todo ese ítem de evaluaciones o un 1.1 en el curso. Además, los antecedentes serán enviados a la Dirección de Docencia de la Escuela de Ingeniería para evaluar posteriores sanciones en conjunto con la Universidad, las que pueden incluir un procedimiento sumario. Por “copia” o “plagio” se entiende incluir en el trabajo presentado como propio, partes desarrolladas por otra persona. Está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la cita correspondiente.