



IIC2115 – Programación como Herramienta para la Ingeniería (II/2020)

## Laboratorio 1 - Elementos básicos

### Objetivos

- Aplicar los contenidos de programación orientada a objetos, herencia y polimorfismo en la elaboración de un simulador.

### Entrega

- **Lenguaje a utilizar:** Python 3.6 o superior
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L01**.
- **Entrega: domingo 23 de Agosto a las 23:59 hrs.**
- **Formato de entrega:** archivo python notebook (**.ipynb**) y archivo python (**.py**) con la solución de este enunciado. Los archivos deben estar ubicados en la carpeta **L01**. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su tarea. Los archivos **ipynb** y **py** deben contener la misma solución.
- **Descuentos:** el descuento por atraso se realizará de acuerdo a lo definido en el programa del curso. Además de esto, tareas que no cumplan el formato de entrega tendrán un descuento de 0.5 pts.
- **Tareas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**
- Si su laboratorio es entregado fuera de plazo, tiene hasta el **lunes 24 de agosto a las 11:59AM hrs** para responder el formulario de **entregas fuera de plazo** disponible en el Syllabus.
- Las discusiones en las *issues* del Syllabus en GitHub son parte de este enunciado.

## Introducción: Invernadero inteligente

La crisis sanitaria internacional COVID-19 nos ha mantenido a muchos confinados en nuestras casas, en ocasiones, con escasa posibilidad de ir de compras para abastecerse. Frente a esta problemática, usted ha decidido construir un invernadero y plantar algunos alimentos.

Dado que ha tomado el curso de Programación como Herramienta para la Ingeniería, ha decidido programar cierta inteligencia en el invernadero con el fin de despreocuparse de los cuidados y maximizar la producción. Para poder llevar a cabo su proyecto, ha decidido encargar por internet algunos sensores y artefactos. Con ellos usted pretende medir la humedad o temperatura y programar riegos, entre otras.

Para probar su sistema previo a construirlo, ha decidido realizar una simulación para definir y probar la lógica inteligente de su proyecto.

## El invernadero

El invernadero corresponde a una estructura que alberga plantas, sensores y artefactos, en la que podrá realizar cosechas de insumos para poder alimentarse. El invernadero esta posicionado en el exterior, donde se conoce a toda hora la cantidad de luz y temperatura:

- La **luz exterior** corresponde a un indicador que vive en el intervalo  $[0, 100]$ . Se obtiene a partir de la hora del día de acuerdo a las siguientes reglas:
  - 6:00 am - 12:00 pm: Crece linealmente de 0 a 100.
  - 12:01 pm - 7:00 pm: Decrece linealmente de 100 a 0.
  - 7:01 pm - 5:59 am: Se mantiene en 0
- La **temperatura exterior** corresponde a un indicador medido en grados celsius. Se obtiene a partir de la medición de un sensor de luz  $L$ , a través de la fórmula  $L * 3/10 + aleatorio(-10, 10)$ , donde *aleatorio* corresponde a una variable aleatoria que se mueve entre -10 y 10 (investigue el uso de la librería `random`).

## Plantas

Las plantas corresponden a lo que se cultivará dentro del invernadero. Se dispone de variados tipos que presentan características diferentes. Todas deben tener el método `cosechar()` y deben estar identificadas con:

- **fecha\_plantacion:** Variable tipo *datetime* que almacena la fecha en que fue plantada. (Investigue el uso de la librería `datetime`).
- **humedad\_optima:** Variable de tipo *float* que almacena la humedad óptima de la tierra para que la planta maximice la producción.
- **humedad\_actual:** Variable de tipo *float* que almacena la humedad actual en la tierra de la planta.

Además, las plantas pueden ser:

- **lechuga:** Corresponde a una planta de lechuga que además incorpora la variable **n\_hojas** que guarda el número de hojas de la lechuga y la variable **tamano** que puede ser ‘*grande*’ (sobre 40 hojas), ‘*mediana*’ (entre 10 y 40 hojas) o ‘*chica*’ (menos de 10 hojas). El método cosechar extrae el número de hojas indicado en el método y debe actualizar correctamente el nuevo **tamano**.
- **tomate:** Corresponde a una planta de tomates que además incorpora la variable **raza** que puede ser ‘*normal*’ o ‘*cherry*’, la variable **n\_tomates** que guarda el número de tomates en la planta y la variable **acidez** que guarda el porcentaje de acidez del tomate. El método cosechar extrae el número de tomates indicado en el método y si se trata de un tomate cherry no debe permitir sacar tomates antes de medio día.
- **cilantro:** Corresponde a una planta de cilantro que además incorpora la variable **n\_ramas** que guarda el número de ramas del cilantro. El método cosechar extrae el número de ramas indicado en el método.

## Sensores

Los sensores se instalan en el invernadero o en cada planta para medir parámetros. Todos los sensores siempre almacenan la **ultima\_medición** y tienen el método `medir()`. Los tipos de sensores son:

- **Humedad:** Este sensor se instala en cada planta para medir el nivel de humedad de la tierra. Si se trata de un sensor de humedad, el método `medir()` debe retornar la humedad entre 0 y 100. Para simular la medición de humedad, reste un número aleatorio entre 1 y 30 de la **ultima\_medición**.
- **Temperatura:** Este sensor se instala a nivel de invernadero para medir la temperatura en el aire de este. Si se trata de un sensor de temperatura, el método `medir()` debe retornar la temperatura entre -30 y 50 celsius. Para simular la medición de temperatura, sume o reste un número aleatorio entre 1 y 5 para acercarse a la temperatura exterior.

- Luz: Este sensor se instala a nivel de invernadero para medir la luz dentro del invernadero. Si se trata de un sensor de luz, el método `medir()` debe retornar la intensidad de luz entre 0 y 100. Para simular la medición de luz, reste un número aleatorio entre 5 y 10 de la luz exterior.

## Regador

Instrumento para regar una planta cuando se alcance una humedad límite en base a condiciones. Por ejemplo, una planta se debe regar a plena luz del día (luz superior a 60), cuando la temperatura sea superior a 10 grados y la humedad este por debajo de 12. En este caso, cree su propia lógica para regar una planta hasta su humedad óptima.

## Crecimiento

Defina el método `crecer()` dentro de una planta, que aumente bajo algún criterio el tamaño de sus plantas, por ejemplo, cada 5 días. Entiéndase por tamaño, el número de hojas, tomates y ramas.

## Simulación

Su objetivo es programar mediante un enfoque de Programación Orientada a Objetos una simulación que reproduzca la situación descrita y le permita cosechar sus productos.

Para ello, básiense en la estructura de a continuación:

```
# Defina su invernadero
# Defina sus plantas y sensores
h = 0 # Variable que lleva el conteo de la hora
dias_a_simular = 30 # Total de dias que va a simular
while h <= 30*24:
    # Determine los parametros externos
    # Evalúe los parametros
    # Coseche cuando se pueda
```

Dentro de la simulación debe realizar impresiones de lo que sucede sujeto a los siguientes criterios:

- Al iniciar la simulación se debe imprimir cuantas plantas hay de cada tipo y la cantidad de hojas, tomates o ramas que cada una tiene.

- Cuando se produce un riego se debe imprimir un mensaje de qué planta se ha regado, la humedad actual antes de regar, la temperatura y la cantidad de luz.
- Cuando se cosecha, se debe indicar: la planta cosechada, las condiciones (temperatura, luz y humedad) y el estado final del planta (cantidad de hojas, tomates o ramas)
- Al finalizar se debe indicar el total cosechado para cada planta.

## Corrección

Para la corrección de este laboratorio, se revisará el modelo de clases y la implementación de la lógica descrita. Para que obtenga una buena calificación, se le exigirá el uso de las siguientes estructuras y contenidos:

- Listas y diccionarios
- Herencia
- Polimorfismo

## Política de Integridad Académica

*“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.