



IIC2115 – Programación como Herramienta para la Ingeniería (I/2019)

## Laboratorio 1 - Programación Orientada a Objetos

### Objetivos

- Aplicar los contenidos de Programación orientada a objetos, a través de una simulación de eventos reales.

### Entrega

- **Lenguaje a utilizar:** Python 3.6
- **Lugar:** repositorio privado en GitHub. Recuerde incluir todo en una carpeta de nombre **L01**.
- **Entrega parcial:** lunes 18 de marzo a las 16:50 hrs.
- **Entrega final:** jueves 21 de Marzo a las 23:59 hrs.
- **Formato de entrega:** archivo python notebook (**.ipynb**) con la solución, ubicado en la carpeta **L01**. Suba además, en la misma carpeta, un archivo **README.md** con las instrucciones para ejecutar su tarea. No se debe subir ningún otro archivo a la carpeta. Utilice múltiples celdas de texto y código para facilitar la revisión de su tarea.
- **Descuentos:** se descontará 0.5 pts. por cada hora de atraso y fracción en la entrega final. Tareas que no cumplan el formato de entrega tendrán un descuento de 0.5 pts.
- **Entregas parciales subidas fuera de plazo no serán consideradas.**
- **Tareas con errores de sintaxis y/o que generen excepciones serán calificadas con nota 1.0.**

# Introducción

## Mejoras en la ciudad Comodoro

Bastián, el alcalde de la ciudad Comodoro está muy preocupado en mejorar la infraestructura vial de su ciudad. Para ello, necesita conocer cómo las personas toman decisiones a la hora de viajar, y de este modo, conocer el uso de los diferentes espacios. Bastián ha tomado cartas en el asunto y te ha contratado a ti para que modelés el comportamiento de los viajes en la ciudad. Tu modelo debe ayudar al alcalde a saber qué es prioritario de intervenir a fin de tener un mayor impacto.

## Modelación

A continuación, se describen los elementos que debes tener en consideración para realizar la modelación pedida. Posteriormente, se entregan detalles del comportamiento de la ciudad y sus habitantes.

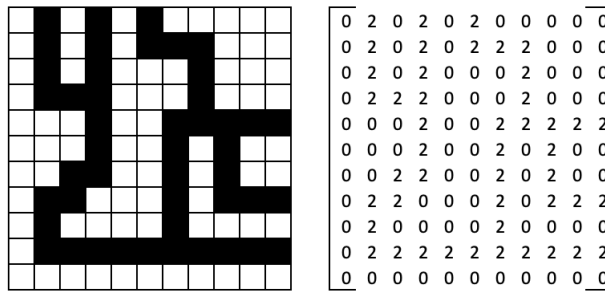
### Mapa de la ciudad

La ciudad Comodoro corresponde a una cuadrícula de  $\mathbf{m} \times \mathbf{n}$  celdas. Cada celda puede ser de **edificación**, **vialidad** o celda **vacía**. Las celdas edificación pueden ser de tipo **hogar** u **oficina** y en ambas pueden generarse viajes a la red vial. Por otro lado, las celdas vialidad son parte de la recién mencionada red vial y es por donde circulan los habitantes de la ciudad. Estas pueden ser de tipo **camino**, **calle** o **autopista** lo que incide directamente en la velocidad de circulación de las entidades por la vía.

La ciudad Comodoro puede ser representada como una matriz de  $\mathbf{m} \times \mathbf{n}$ , donde cada coeficiente representa un tipo de celda. A continuación se describe la simbología de celdas utilizada:

- **-1**: vacía
- **0**: hogar
- **1**: oficina
- **2**: camino
- **3**: calle
- **4**: autopista

En la siguiente figura se expone un ejemplo de una ciudad Comodoro de  $\mathbf{9} \times \mathbf{9}$ , con celdas hogar (grises) y celdas camino (blancas), con su representación matricial adjunta.



Cabe destacar que en la ciudad Comodoro, la vialidad es siempre de una celda de ancho y no pueden existir vías paralelas adyacentes, es decir, deben haber celdas de infraestructura entre ellas. Además, no puede existir vialidad perimetral en las celdas de los bordes de la ciudad. Es decir, la vialidad solo puede comenzar o finalizar en el borde, no recorrerlo. Los mapas con los que usted va a trabajar no presentan loops. Es decir, mapas como el de la figura no será necesario de resolver.

## Habitantes y sus modos de transporte

Los habitantes de la ciudad se pueden mover por la vialidad de la ciudad Comodoro en 4 modos de transporte: caminata, bicicleta, auto y bus.

- **Caminata:** Equivale a una persona.
- **Bicicleta:** Pueden existir bicicletas individuales o de dos personas.
- **Auto:** Tiene una capacidad máxima de 5 personas y mínima de 1.
- **Bus:** Tiene una capacidad máxima de 40 personas.

Para determinar el nivel de ocupación en cada uno de los modos (a excepción del modo caminata) basta con utilizar una **distribución uniforme** (de números enteros) entre los valores posibles de capacidad. Para ello utilice la función disponible en la librería descrita más abajo.

## Eventos y movilidad

- **Generación de viajes:** Los viajes son generados de forma aleatoria en todas las entradas de la red (vialidad en el borde) y en los hogares u oficinas que colindan con la vialidad (que sean adyacentes por una arista con la vía). Una vez que el viaje es generado, se debe determinar el modo de ese viaje. Para ello considere que hay un 15% de probabilidad que sea en el modo caminata, 10% en bicicleta, 25% en auto y 50% en bus. Posteriormente, si corresponde, genere la cantidad de personas con la

función ya descrita. Si el viaje es generado en un hogar, este ingresa a la celda vialidad de acuerdo a la **preferencia**: arriba, abajo, izquierda y derecha. Es decir, si una celda hogar tiene vialidad a la izquierda y abajo, entonces el viaje comienza por abajo.

- **Desplazamiento:** Las entidades se desplazan por la vialidad a cierta velocidad (explicada más abajo). Cada vez que una entidad se enfrenta a más de un camino, debe decidir estocásticamente cuál tomará (ver función más abajo). Si la distancia cubierta en un turno es mayor a la distancia hasta el siguiente cruce, el modo debe detenerse en el cruce, decidir, y luego continuar desplazandose la distancia restante.
- **Fin de los viajes:** Los viajes finalizan al salir por una celda vialidad del borde. En resumen, los viajes pueden ocurrir entre dos bordes o entre una edificación y un borde.
- **Velocidad de desplazamiento:** La velocidad de cada modo de transporte dependerá de sus características propias y del tipo de vialidad. Para cada posible combinación, la siguiente tabla presenta la velocidad en celdas/turno:

		Vialidad		
		Camino	Calle	Autopista
Modo	Caminata	1	1	1
	Bicicleta	2	3	3
	Bus	2	4	5
	Auto	3	5	7

**Hint:** Para la modelación del problema, considere que cada modo de transporte es el encargado de calcular la velocidad en cada turno.

## Simulación

Usted deberá construir una simulación que muestre los viajes que se producen en la ciudad Comodoro. Es decir, deberá modelar la situación descrita y representar los viajes que se generan en la ciudad.

Deberá simular turnos donde ocurren desplazamientos, generación de viajes y decisiones de ruta. En paralelo a la matriz de la ciudad, debe generar una matriz de igual dimensión ( $\mathbf{m} \times \mathbf{n}$ ), que describe el estado de la red mediante la cuenta del número de personas presentes en cada celda. Es decir, deberá actualizar en cada turno el estado de esta matriz. En caso de finalizar un viaje, el conteo de personas que se encontraban en el modo de transporte debe agregarse a la celda del borde donde terminó el viaje.

La generación de viajes se rige por una distribución de **Poisson**. Cada vez que un viaje es generado, debe determinar la próxima salida por ese punto. Considere que se generan **en promedio** 1 viaje por turno en los terminales de camino, 1 viaje cada 2 turnos en hogares y 1 viaje cada 3 turnos en oficinas.

La generación de eventos en la simulación terminará una vez que se hayan cumplido  $t$  turnos, donde  $t$  es una constante predefinida, entregada a la simulación. Una vez que hayan transcurrido los  $t$  turnos, la simulación debe continuar hasta que **todos los viajes hayan terminado**.

En cada turno de la simulación, deberá imprimir cada uno de los eventos generados, además del estado actual de la matriz de conteo de personas. Para esto último, utilice la función entregada en el Syllabus junto con el enunciado del laboratorio.

**Hint:** si la simulación es demasiado rápida y no puede ver la impresión, agregue luego de cada turno un llamado a la función `time.sleep(s)` del módulo `time`, donde `s` es la cantidad de segundos (puede ser una fracción) que el programa se detendrá antes de ejecutar la siguiente instrucción.

## Librería Comodoro

Usted tendrá a disposición una librería llamada Comodoro (`comodoro.py`). Esta debe ser importada en el proyecto y no debe ser modificada. Además, se cuenta con una carpeta `mapas` que contiene 4 mapas para desarrollar el laboratorio. Tanto la librería como la carpeta deben estar en la misma ubicación que su archivo `.ipynb`. Además, debe incluir en su archivo la línea “**import comodo**” al inicio de su trabajo para acceder a todas las funciones disponibles. A continuación se describen los métodos que incluye la librería.

1. **comodoro.generar\_mapa(nMapa):** Recibe un número de mapa (actualmente de 1 a 4) y genera una matriz que describe un mapa de la ciudad Comodoro.
2. **comodoro.imprimir\_mapa(personas, vialidad):** Recibe una matriz donde cada celda es el número de personas en ella y la matriz de vialidad generada por la función `generar_mapa()`. Esta imprime un gráfico de calor con el estado de la ciudad.
3. **comodoro.imprimir\_mapa(personas, vialidad):** Recibe una matriz donde cada celda es el número de personas en ella y la matriz de vialidad generada por la función `generar_mapa()`. Esta imprime un gráfico de calor con el estado de la ciudad.
4. **comodoro.uniforme(a,b):** Retorna un número aleatorio entre los valores “a” y “b”.
5. **comodoro.exponencial(1/λ):** Retorna un número aleatorio exponencial donde el parámetro es el inverso de la tasa ( $1/\lambda$ ).

6. **comodoro.ruta(a,b,i,d)**: Recibe 4 bools que indican las rutas posibles que puede seguir una entidad a la hora enfrentar una intersección. Considerando que “a” es arriba, “b” es abajo, “i” es izquierda y “d” es derecha. Se deben indicar como “True” todas las direcciones disponibles a seguir. Por ejemplo, si solo se puede seguir hacia abajo y a la izquierda, los parámetro de la función deben ser (False, True, True, False). La función luego retorna la decisión con un string del tipo “a”, “b”, “i” o “d”. En el ejemplo podría retornar “b” o “i”.

## Corrección

Para la corrección de este laboratorio, se revisarán las clases definidas, la relación entre estas y los métodos implementados. Es decir, cómo diseñó su modelo orientado a objetos. Además, su programa debe ser capaz de imprimir el estado de la red en cada turno.

### Avance parcial (final del la clase)

1. Modelo de clases completo.
2. Impresión de un estado cualquiera de la red.

### Entrega final

1. Modelo de clases completo.
2. Simulación de la situación.
3. Impresión de eventos y estados por turno.

Para revisar el correcto funcionamiento de la simulación se comparará el estado final de la red, con lo que debiera resultar luego de la ejecución. En base a esto, usted puede tenerla perfecta, regular o incorrecta.

## 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.