



IIC2115 – Programación como Herramienta para la Ingeniería (I/2023)

## Ejercicio Capítulo 1a

### Aspectos generales

- **Objetivos:** Aplicar los contenidos fundamentales de programación orientada a objetos para modelar un local de votación.
- **Lugar de entrega:** lunes 13 de marzo a las 16:30 hrs. en el buzón de tareas.
- **Formato de entrega:** archivo Python Notebook (**C1a.ipynb**) con el avance logrado durante la sesión. El archivo debe estar ubicado en la carpeta **C1a**. Utilice múltiples celdas de texto y código para facilitar el trabajo del cuerpo docente.

### Introducción

Con el fin practicar las técnicas de POO, en este ejercicio deberá modelar un local de votaciones, considerando las entidades principales que participan en su funcionamiento. Esto significa, definir las **clases** correspondientes, sus relaciones y comportamientos. Una vez finalizado esto, deberá simular el funcionamiento del local, indicando el tiempo de espera promedio por mesa.

### Descripción del problema

El local de votación opera como un sistema con  $N$  filas de espera, en donde los votantes llegan aleatoriamente con una probabilidad  $p$  y se ubican en la fila correspondiente a su mesa. Un individuo vota cuando llega al principio de la fila, y el tiempo de votación (lo que demora en votar, sin considerar la espera) depende de la edad del votante. Considere finalmente que el local de votación tiene un aforo máximo en su interior, por lo

que cuando este es superado, las filas de cada mesa deben continuar en el exterior. El criterio para manejar las filas exterior y el momento en que un individuo pueden entrar al local debe ser definido por usted.

## Modelo de clases

El modelo a implementar debe considerar los siguientes elementos:

- Clases para todas las entidades relevantes a modelar.
- La jerarquía completa de clases para individuos, que considere una clase base abstracta (primera clase de la jerarquía) que solo defina la interfaz (atributos y métodos sin implementar).
- Dos clases que hereden de otra (puede no ser la misma).
- Clases que participen como atributos en otras (composición o agregación)
- Implementaciones del método `__str__()`, que entreguen información relevante del estado de los objetos.
- Dos sobrecargas de métodos (*override* - polimorfismo)

Se recomienda fuertemente comentar el código y definir en celdas de texto qué es cada clase. Considere además que para definir las clases, sus relaciones y comportamientos, no es necesario tener desarrollada la simulación.

## Simulación

Durante la simulación debe entregar información sobre el proceso cada vez que ocurre algún evento, además de reportar el tiempo de espera promedio por mesa. No olvide utilizar clases para modelar las entidades. Si lo desea, puede considerar la siguiente secuencia de instrucciones como guía:

```
# creación de variables y objetos relevantes
for t in range(t_max):
    if llega un votante nuevo
        # agregar votante a la fila externa correspondiente
    if aforo máximo no alcanzado
        # traspasar votantes de filas externas a internas hasta llegar al aforo máximo
    if no hay individuos votando en una fila y sí hay votantes en espera para hacerlo
        # se atiende al primer votante de la fila
    # se actualiza el tiempo de espera
```