



IIC1103 – Introducción a la Programación  
2 - 2017

## Enunciado Tarea 1

### Recordatorio:

- **Fecha de entrega:** Viernes 15 de septiembre de 2017, a las 23:50 hrs.
- **Foro de consulta:** <https://piazza.com/uc.cl/summer2017/iic1103>
- Este trabajo es **estrictamente personal**. Recuerda leer la Política de Integridad Académica del DCC disponible en <http://www.ing.uc.cl/ciencia-de-la-computacion/programas/licenciatura/politica-de-integridad-academica/>. Se usará un software **anti-plagio** para detectar similitud entre códigos.

### ¡Atención!

Ten en consideración que **no se recibirán entregas fuera del plazo**. Tampoco se evaluará tu tarea si te equivocas al entregar el archivo.

Es tu responsabilidad subir entregas parciales de tu tarea. Se revisará **solamente la última versión** que hayas subido. Revisa la sección **Entrega** para las instrucciones de entrega de tu trabajo.

### Objetivo

Aplicar conocimientos adquiridos en clases incluyendo asignación de variables, operaciones y expresiones, interacción con usuarios mediante *input* y *output*, control de flujo (**if**, **elif**, **else**), ciclos (**while**) y funciones.

### Enunciado

Dos jóvenes programadores, Hugo y Nicolás, siempre compiten entre ellos para decidir distintas cosas: quién va a ser el jefe de un grupo de trabajo, quién carrea en LOL, entre otras cosas. Sin embargo, Hugo siempre sospecha que Nicolás hace trampa, por lo tanto decidió que él iba a programar un juego donde no sea posible hacer trampas. El que gane el juego tendrá el honor de tomar el último puesto de ayudantía de introducción a la programación. Hugo eligió crear un BINGO<sup>1</sup>.

---

<sup>1</sup>Sobre todo porque el Bingo es una clara prueba de habilidad...

B	I	N	G	O	B	I	N	G	O	B	I	N	G	O

Cuadro 1: Figuras posibles en el BINGO. Cartón completo, diagonal, y letra Y

## Reglas del juego

En esta tarea tendrás que programar el funcionamiento del juego BINGO.

El BINGO es un juego de azar que consiste en que un locutor va sacando bolitas con números dentro de un cierto rango que están en una tómbola, y anunciando los números en voz alta. Los jugadores juegan usando cartones con números escritos en ellos, dentro del mismo rango de las bolitas. Si un jugador tiene dicho número en su cartón, lo marca. Una partida de BINGO continúa hasta que alguien consigue marcar en su cartón todos los números que formen alguna figura específica.

Tu programa deberá implementar una secuencia de partidas de BINGO. Ambos jugadores poseen una cantidad de dinero inicial y apuestan una cierta cantidad de dinero al iniciar cada partida (el dinero inicial y el monto de las apuestas es el mismo para ambos jugadores). Una partida termina cuando un jugador consigue completar cualquiera de las figuras indicadas en el Cuadro 1. Cada figura tiene un valor distinto de acuerdo al siguiente orden:

Cartón completo	Figura de mayor valor
Letra Y	Valor intermedio
Diagonal izquierda-derecha hacia abajo	Figura de menor valor

En caso de que un jugador consiga formar más de una figura a la vez, se considera la figura de mayor valor. Por ejemplo, si el jugador forma la Y y la diagonal al mismo tiempo, entonces se considera solo el valor de la Y.

En caso de que ambos jugadores formen una misma figura al mismo tiempo, cada uno obtiene lo que apostó al inicio de la partida, sin perder nada. Por otro lado, si ambos jugadores forman distintas figuras a la vez, el ganador será el que haya formado la figura de mayor valor. El ganador de la partida recibirá el monto acumulado de la apuesta.

Al finalizar una partida, se pregunta a los jugadores si quieren jugar de nuevo. Si es así, tendrás que iniciar una nueva partida generando nuevos cartones aleatorios y pedir nuevamente la apuesta.

El programa termina cuando, al finalizar una partida, un jugador se queda sin dinero, o bien cuando los jugadores deciden no jugar una siguiente partida. El ganador final es aquel jugador que tenga más dinero.

# Implementación

## Interfaz. Módulo Tarea\_1

Con el objetivo de que esta tarea se vea más profesional, se te proporcionará un módulo **Tarea\_1** que provee una interfaz gráfica con las siguientes funciones:

- **app.colocar\_numero(fila, columna, numero, jugador):** permite colocar un número en el cartón. Recibe la posición y el número que se desea colocar, y el número del cartón (1 ó 2) al que desea colocar un número. La forma en que se especifican las coordenadas se encuentra explicada en la sección “Posiciones”.
- **app.marcar\_numero(fila, columna, opcion, jugador):** esta función permite marcar o desmarcar un número del cartón. Recibe la posición deseada, la opción **True** si desea marcar, o **False** si desea desmarcar el número, y el jugador (1 ó 2).
- **app.mostrar\_dinero(jugador, monto):** recibe el jugador (1 ó 2) y el monto que se desea mostrar en pantalla. Además guarda el dinero del jugador dentro de la interfaz. Para poder obtenerlo de nuevo se utiliza la función **app.preguntar\_monto**.
- **app.obtener\_numero(fila, columna, jugador):** dada una posición y un jugador (1 ó 2), esta función entrega el número que se encuentra en dicho lugar.
- **app.mostrar\_mensaje(mensaje):** recibe un mensaje para mostrarlo en la interfaz. Puedes usar esta función para mostrar el número extraído en el turno y, en caso de que haya un ganador, utilizarla para anunciar si ganó el jugador 1 o el jugador 2.
- **app.mostrar\_ventana(estado):** si estado es **True**, se abre la ventana; en otro caso, se guarda la ventana. Esto sirve para que el usuario sepa que tiene que interactuar con la consola cuando se cierra, y que debe interactuar con la interfaz cuando aparece.
- **app.esta\_marcado(fila, columna, jugador):** recibe la posición de un número del cartón del jugador (1 ó 2). Retorna **True** si el número está marcado, y en otro caso retorna **False**.
- **app.preguntar\_monto(jugador):** recibe el número de jugador (1 ó 2) y retorna el monto de dinero que el jugador tiene.
- **app.poner\_apuesta(monto):** recibe el monto que ambos jugadores apostaron. Puede utilizar esta función para guardar el valor de la apuesta y luego pedirla con la función **obtener\_apuesta()**.
- **app.obtener\_apuesta():** entrega el monto de la apuesta que ambos jugadores realizaron.
- **app.cerrar\_ventana():** cierra completamente la ventana. Puede utilizarla al finalizar el juego.

También dispondrás de dos funciones especiales para complementar tu tarea. Estas funciones permiten administrar un contador de números. La función **app.agregar(numero)** entrega un número al contador, y retorna un **bool** que es **False** cuando el contador ya ha recibido ese número antes, o **True** cuando no lo ha recibido. Puedes utilizar esta función para saber si una bolita ya ha sido vista en el BINGO. Finalmente, la función **app.reiniciar\_contador()** permite reiniciar el contador borrando todos los números que ha recibido.

Jugador					
	0	1	2	3	4
0	B 0	I 0	N 0	G 0	O 0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	0	0

Figura 1: Formato de un cartón en la interfaz

## Formato del cartón

En el juego del BINGO, cada columna de un cartón solamente pueden tener números en un rango definido. Para esta tarea el rango será de veinte números, de manera que en la columna “B” solamente pueden aparecer números del 1 al 20, en la columna “I”, únicamente números del 21 al 40, y de esta manera hasta la letra “O”. No pueden haber números repetidos dentro de un mismo cartón.

## Posiciones

Las coordenadas de los números del cartón se detallan en la figura 1.

Cuando utilices las funciones que reciben una posición, ésta debe ser especificada con valores `fila` y `columna`. El valor `fila` corresponde a la coordenada que se muestra en la izquierda del cartón (números en color morado del 0 al 4). El valor `columna` corresponde a la coordenada que está en la parte superior del cartón.

Como ejemplo, si deseas utilizar la esquina inferior derecha del cartón (viéndolo de frente) la posición sería la fila 4 y la columna 4. Si se desea referenciar al primer número de la columna “N”, entonces la posición es fila 0 y columna 2.

## Interacción con el usuario

Es importante mantener una interacción fluida con el usuario para que la experiencia del juego sea grata.

Al comienzo de tu programa, debes saludar al usuario y preguntar por el monto inicial que se cargará a ambos jugadores. Durante todo el juego deberás mostrar en la interfaz la cuenta de los montos disponible para cada jugador.

```

from Tarea_1 import *

root = tk.Tk()
root.geometry('{}x{}'.format("550", "650"))
app = Application(master=root)

def turno():
    # AQUI DENTRO EMPIEZA CADA TURNO
    pass # eliminar esta linea una vez hayan escrito algo aqui

# Aqui empieza su programa

# ESTO NO SE TOCA
app.button.config(command=turno)
app.mainloop()

```

Figura 2: Código de inicio `main.py`

Al inicio de cada partida debes preguntar qué cantidad de dinero se desea apostar, y luego generar un cartón distinto para cada jugador. Al final de cada partida debes mostrar un mensaje en la interfaz indicando quién fue el ganador (si lo hay), y entregar el dinero que le corresponde.

Cada turno se ejecuta mediante un botón de la interfaz llamado “Siguiente turno”. Al presionar este botón, se ejecuta la función `turno()` que se encuentra en el archivo `main.py` que se te entregó. Dentro de esta función debes programar lo que sucede en cada turno, tal como se indica en la figura 2.

En cada turno deberás mostrar un número al azar (sacar una bolita) entre 1 y 100 que no haya salido antes durante la partida. Cuando algún jugador tenga en su cartón el número que salió, entonces debes marcarlo en la interfaz gráfica. Si algún jugador completa una figura, debes terminar la partida y determinar al ganador. Luego se pregunta si los jugadores desean seguir jugando o no. En caso de que elijan que sí, se les pregunta de nuevo el monto a apostar y se generan cartones con **nuevos** números para cada uno, y empieza una nueva partida. En el caso de que no quieran seguir jugando o alguno de los jugadores no tenga más dinero, el juego se despide y el programa termina.

Para efectos de la tarea, suponga que el usuario<sup>2</sup> utilizará correctamente la interfaz y, por ejemplo, no seguirá presionando el botón cuando haya terminado el juego.

## Consideraciones adicionales

- Los números del Bingo van del 1 al 100.
- Durante una misma partida no puede salir un número repetido de la tómbola. Para esto considere utilizar las funciones `app.agregar(numero)` y `app.reiniciar_contador()`.

---

<sup>2</sup>y los ayudantes al momento de corregir

- Un jugador no puede apostar más dinero del que posee <sup>3</sup>. Si el usuario intenta apostar una cantidad que no tiene, debe avisar por la consola o por la interfaz y preguntar de nuevo el monto a apostar.
- Sin desmedro de lo anterior, puedes suponer que los ayudantes probarán correctamente tu tarea, usando valores válidos como *input* y utilizarán correctamente la interfaz.

## Entrega

Debes guardar tu tarea en un archivo de nombre `tarea1_numero_alumno.py`, donde debes reemplazar `numero_alumno` con tu número de alumno. Por ejemplo, si tu número de alumno es 12345678 el nombre de la tarea sería `tarea1_12345678.py`. **Si no sigues estas instrucciones de entrega, el ayudante corrector tiene el deber de descontar 5 décimas a tu tarea.**

La entrega se realiza a través de un cuestionario en el SIDING disponible en la página web del curso hasta el Viernes 15 de septiembre de 2017 a las 23:50 hrs. **No se recibirán entregas atrasadas ni entregadas por otro medio.**

## Indicaciones generales

1. Si utilizas el sistema operativo Linux y no tienes instalada la librería `tkinter` puedes utilizar el siguiente comando en la terminal para instalarlo si tu distribución es Ubuntu o Debian:

```
sudo apt-get install python-tk
```

Ante cualquier problema que suceda con respecto a esta librería, siéntete libre de preguntarlo en el foro del curso.

2. Para esta tarea **NO debes modificar** la librería `Tarea_1.py`, pues los ayudantes utilizarán la librería original, sin cambios. Tampoco debes entregar la librería `Tarea_1.py`.
3. Recuerda que la tarea es **estrictamente individual**. Cualquier situación de copia será sancionada severamente según dicta el código de honor de la universidad, el cual se detalla al final de este documento.
4. **NO hagas la tarea a última hora**, pues el cuestionario se cierra automáticamente cuando se cumple el plazo. **NO se aceptarán entregas atrasadas ni entregadas por otros medios.**
5. Revisa bien lo que entregas, y prueba tu código antes de enviar la versión definitiva. **Puedes enviar tu tarea todas las veces que estimes conveniente, pero solo se almacenará y revisará la última entrega.** Si el ayudante corrector no logra hacer correr tu tarea, es bastante probable que ésta sea evaluada con una nota baja o cercana a 1,0.

## 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*

---

<sup>3</sup>No tenemos un sistema de préstamos implementado

*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.