



PROGRAMACIÓN I

CS1111 - INTRODUCCIÓN A LA CIENCIA DE LA
COMPUTACIÓN

PROYECTO 1: JUEGO DEL ADIVINA QUIÉN?

LABORATORIO 3.05

June 21, 2021

1 ENUNCIADO DEL PROYECTO

Para este proyecto desarrollara un programa para jugar el popular juego: Adivina Quién?. El juego, en líneas generales, involucra dos jugadores, cada uno elige a un personaje misterio y debe adivinar el personaje del otro jugador mediante preguntas de sí o no. Cada jugador tiene una lista de posibles personajes para hacerle el trabajo más fácil.

1.1 REQUERIMIENTOS DEL PROYECTO

Las reglas generales del juego deben ser respetadas. Si un jugador decide adivinar el personaje del otro por nombre, gana si está en lo correcto y pierde si el intento es incorrecto (la condición de perder puede ser modificada en la configuración del juego). Además, las preguntas para ganar información sobre el personaje sólo deben revelar una característica a la vez. Algunas de las funcionalidades específicas para este proyecto se detallan a continuación (referirse a la Sección 2 para una muestra visual de lo que se pide):

Menus

1. El juego debe tener un menú principal con las opciones: '1. JUGAR', '2. CONFIGURACION', '3. HIGHSCORES', y '4. SALIR'.
2. En la opción 'JUGAR', aparece: 'INGRESE EL NOMBRE DEL JUGADOR 1 (INGRESE 0 PARA VOLVER ATRAS): '.
3. Al ingresar un nombre: 'INGRESE EL NOMBRE DEL JUGADOR 2: (INGRESE 0 PARA VOLVER ATRAS):'. Tanto en este punto como en el anterior, al ingresar un valor de 0, se vuelve un paso atrás.
4. En la opción 'CONFIGURACION', se debe considerar un submenu con las siguientes opciones: '1. CONFIGURAR INTENTOS', '2. REGRESAR' y '3. SALIR'; las acciones para cada una de estas opciones serían: configurar la cantidad de intentos (ver el siguiente punto), regresar al menú anterior y salir del juego, respectivamente.
5. En la opción: 'CONFIGURAR INTENTOS' se debe solicitar al usuario ingresar el número de intentos de adivinanza que tiene cada jugador en el juego, estas opciones sólo pueden ser 3: 1, 3 o 5. En caso el usuario ingrese algún valor diferente, se debe mostrar el error: 'OPCION INCORRECTA. INTENTE DE NUEVO.' y se debe solicitar de nuevo otra opción. En caso de ingresar una opción válida, se debe regresar al menú principal.
6. En la opción: 'HIGHSCORES', como parte de este avance de proyecto, sólo mostrara el mensaje: 'EN CONSTRUCCION' y solicitará ingresar una opción al usuario con el mensaje: '1. REGRESAR'.
7. Al final de cada menú y después de un salto de línea, se debe mostrar al usuario el mensaje: 'SELECCIONE UNA OPCION: '. Instantáneamente después, se debe solicitar como entrada de usuario una opción numérica en caso de elegir una opción de menú, o alfanumérica en caso de ingresar los nombres de los jugadores. De no ser así, se debe mostrar el mensaje: 'OPCION INCORRECTA. INTENTE DE NUEVO', y solicitar una nueva entrada.

Funcionalidad

- El juego se desarrolla por turnos. Al iniciar el juego, se le pide a cada jugador, por turno, elegir un personaje misterio. Para esto se mostrará una matriz inicial de personajes con sus características. El juego se desarrolla por turnos. En cada turno, un jugador tiene dos opciones de juego: '1. ADIVINAR PERSONAJE', '2. HACER PREGUNTA'.
- Una vez ingresados los personajes misterio, en cada turno se le muestra el tablero actual al jugador, y se le da dos opciones de juego: '1. ADIVINAR PERSONAJE', '2. HACER PREGUNTA'. El jugador debe elegir la opción correspondiente.
- En la opción 1, el jugador (digamos X) escribe el nombre del personaje y el juego le resta un intento de adivinanza. Luego, al jugador Y se le muestra el personaje adivinado y debe responder con una de dos opciones: '1. ES CORRECTO', '2. ES INCORRECTO'

Si el jugador Y elige la opción 1, el juego acaba y se muestra un mensaje que diga: 'FELICIDADES: EL JUGADOR X GANA'. Si el jugador Y elige la opción 2, se muestra un mensaje con la cantidad de intentos restante para el jugador X. Si se le agotaron los intentos, se muestra un mensaje 'LO SIENTO JUGADOR X. SE LE ACABARON LOS INTENTOS'. Caso contrario, el juego continúa pasando al siguiente turno.

- En la opción 2, el jugador (digamos X) se muestra un mensaje diciendo: 'INGRESE UNA PREGUNTA: '. El jugador la ingresa y al jugador Y se le muestra la pregunta. Se le pide ingresar una de dos opciones: '1. SI', '2. NO'.

Se muestra un resultado de la opción del jugador Y. El jugador X ingresa una lista de nombres a remover de su tablero, y luego se pasa el turno del jugador Y.

- En cada turno, se le muestra su tablero actual a cada personaje. Este consiste en una tabla de nombres con características del personaje. Personajes que han sido removidos por el jugador se muestran como una X.

2 EJEMPLOS DE ENTRADAS Y SALIDAS

2.1 ENTRADAS

- Ejemplo de Menú principal:

```
Bienvenido a Adivina Quien?
1. JUGAR
2. CONFIGURACION
3. HIGHSCORES
4. SALIR
SELECCIONE OPCION:
```

- Ejemplo de submenu de nombres

```
INGRESE EL NOMBRE DEL JUGADOR 1 (INGRESE 0 PARA VOLVER ATRAS):
Jose
INGRESE EL NOMBRE DEL JUGADOR 2 (INGRESE 0 PARA VOLVER ATRAS):
Jorge
```

- Ejemplo de Pantalla de eleccion de personaje al haber elegido Jose de nombre

```
['Jaime', 'sombrero'] ['Juan', 'pelo largo'] ['Carlos', 'pelo corto']
['Maria', 'casco'] ['Penelope', 'pelo rubio'] ['Juana', 'pelo corto']
Elige tu personaje misterio, Jose
```

- Ejemplo de CONFIGURACION

```
1. CONFIGURAR INTENTOS
2. REGRESAR
3. SALIR
SELECCIONE OPCION:
```

2.2 SALIDAS

- Ejemplo de Pantalla inicial al haber elegido a Jaime como personaje misterio

```
Es tu turno, Jose
| Jaime sombrero          | | Juan pelo largo        | | Carlos pelo corto      |
| Maria casco             | | Penelope pelo rubio    | | Juana pelo corto       |
|                           | | Jaime sombrero         | |                          |

1. ADIVINAR PERSONAJE
2. HACER PREGUNTA
SELECCIONE OPCION:
```

- Ejemplo de intento de adivinar personaje de forma correcta

```
Es tu turno, Jose
| Jaime sombrero          | | Juan pelo largo        | | Carlos pelo corto      |
| Maria casco             | | Penelope pelo rubio    | | Juana pelo corto       |
|                           | | Jaime sombrero         | |                          |

1. ADIVINAR PERSONAJE
2. HACER PREGUNTA
SELECCIONE OPCION: 1
INGRESE UN PERSONAJE A ADIVINAR: Maria
Su contrincante esta tratando de adivinar su personaje
Su personaje actual es: ['Maria', 'casco'] . El personaje adivinado es Maria
1. ES CORRECTO
2. ES INCORRECTO
SELECCIONE OPCION: 1
FELICIDADES Jose USTED GANA!
```

- Ejemplo de perder por adivinar personaje de forma incorrecta con un solo intento restante

```

Es tu turno, Jose
| Jaime sombrero          | | Juan pelo largo        | | Carlos pelo corto      |
| Maria casco             | | Penelope pelo rubio    | | Juana pelo corto       |
|                           | | Jaime sombrero        | |                          |

1. ADIVINAR PERSONAJE
2. HACER PREGUNTA
SELECCIONE OPCION: 1
INGRESE UN PERSONAJE A ADIVINAR: Jaime
Su contrincante esta tratando de adivinar su personaje
Su personaje actual es: ['Maria', 'casco'] . El personaje adivinado es Jaime
1. ES CORRECTO
2. ES INCORRECTO
SELECCIONE OPCION: 2
LE QUEDAN 0 INTENTOS
LO SIENTO Jose . SE LE ACABARON LOS INTENTOS.

Process finished with exit code 0

```

- Ejemplo de pantallas al hacer una pregunta

```

Es tu turno, Jose
| Jaime sombrero          | | Juan pelo largo        | | Carlos pelo corto      |
| Maria casco             | | Penelope pelo rubio    | | Juana pelo corto       |
|                           | | Jaime sombrero        | |                          |

1. ADIVINAR PERSONAJE
2. HACER PREGUNTA
SELECCIONE OPCION: 2
INGRESE UNA PREGUNTA: Tiene pelo largo?
| Jaime sombrero          | | Juan pelo largo        | | Carlos pelo corto      |
| Maria casco             | | Penelope pelo rubio    | | Juana pelo corto       |
|                           | | Maria casco           | |                          |

Su contrincante le ha hecho la siguiente pregunta:
Tiene pelo largo?
1. SI
2. NO
SELECCIONE OPCION: 1
| Jaime sombrero          | | Juan pelo largo        | | Carlos pelo corto      |
| Maria casco             | | Penelope pelo rubio    | | Juana pelo corto       |
|                           | | Jaime sombrero        | |                          |

EL JUGADOR HA RESPONDIDO: SI
Ingrese una lista de personajes a remover de su tablero: Carlos, Juana

```

- Ejemplo de tablero luego de haber removido algunos personajes

```

| Jaime sombrero          | | Juan pelo largo        | | X                        |
| Maria casco             | | Penelope pelo rubio    | | X                        |
|                           | | Jaime sombrero        | |                          |

```

2.3 CONTROL DE ERRORES

Los siguientes errores deben ser controlados en el proyecto:

- En caso no se ingrese una opción válida en alguno de los menús, se debe mostrar el mensaje: 'OPCION INCORRECTA. INTENTE DE NUEVO', hasta que se ingrese una opción válida.

- Si en el menú CONFIGURAR INTENTOS, no se ingresa una configuración válida (1,3,5) se debe mostrar el error: 'OPCION INCORRECTA. INTENTE DE NUEVO.' y se debe solicitar de nuevo otra opción

3 PARTES DEL PROYECTO

3.1 PROGRAMACIÓN/REQUISITOS TÉCNICOS

- El programa debe estar hecho en lenguaje Python.
- Debe usar condicionales y bucles para el manejo de menús y para el manejo de los turnos del juego.
- Debe usar listas y/o matrices para dimensionar el tablero.
- Debe usar listas y/o matrices para almacenar la lista de personajes.
- Puede usar funciones de formateo de strings para dar mayor visibilidad a los caracteres.
- Debe separar las funciones que use en al menos 3 módulos de la siguiente forma:
 - Menus.py**, tendrá todas las funciones relacionadas al manejo de menús.
 - Cadenas.py**, tendrá todas las funciones relacionadas al formateo de los caracteres.
 - Main.py**, tendrá El funcionamiento general del juego y deberá llamar a funciones de los módulos anteriores.
- Debe definir al menos 10 funciones en total.
- En las 4 primeras líneas del módulo Main.py se debe colocar en forma de comentario, el número de grupo y los integrantes del mismo (Debe incluir nombres, apellidos y código de estudiante).
- No se deben usar clases, diccionarios o librerías no usadas durante las clases. Pueden usarse funciones relacionadas a strings que no se hayan usado en clases.

3.2 DEMOSTRACIÓN Y PRESENTACIÓN DEL PROYECTO

El proyecto deberá ser presentado de acuerdo a la siguiente dinámica y duración:

- Se realizará una presentación en forma de video por los integrantes del grupo, demostrando el funcionamiento de su implementación, y explicando los detalles técnicos de la solución. El video no podrá pasar de 5 minutos.
- Cualquier estudiante del grupo deberá presentar al equipo y compartir la pantalla para la evaluación. Deberá situarse en las primeras líneas del módulo Main.py.
- El estudiante presentando debe realizar al menos 2 demostraciones señalando que el programa funciona con normalidad y sin defectos. Prepare de antemano los casos necesarios para la demostración.
- Otro estudiante del grupo deberá comentar sobre lo aprendido con Proyecto 1.

- En una clase subsiguiente, el profesor tomará 5 minutos para realizar preguntas al grupo, pudiendo preguntar de manera específica a cualquier integrante.
- Tome en cuenta mucho el tiempo, dado que si se acaba el tiempo y no llegó a presentar lo necesario (sea por temas técnicos que pudieron planificarse) no se tomará en cuenta el resto del video, pudiendo afectar la nota del grupo.

Adicionalmente, tome en cuenta las siguientes consideraciones:

- La asistencia a la clase subsiguiente de preguntas es mandatoria para poder considerar una nota grupal. Exceptuando faltas justificadas a través de counter alumnos, bienestar estudiantil o directamente al profesor con un motivo de fuerza mayor (ej. Salud con descanso médico).
- Tome en cuenta que no hay recuperación sobre esta entrega, debe planificar con tiempo la entrega de este hito.
- Si durante las preguntas, alguien del equipo no conoce el código presentado, la evaluación de todo el grupo se verá perjudicada. No es admisible decir 'Esa parte la hizo mi compañero'

3.3 ENTREGA DEL PROYECTO

El código de proyecto debe ser entregado por Gradescope: <https://www.gradescope.com/courses/266376>. Opcionalmente y altamente sugerido subir el código a Github: <https://github.com/UTEC-CS1100>.

La entrega (video y código) será el Domingo 18 de Julio (Semana 13). Posteriormente, se dedicará la clase del Viernes 23 de Julio a las preguntas por parte del profesor.

4 CRITERIOS DE EVALUACIÓN

Criterio	Excelente (5pts)	Adecuado (4pts)	Mínimo (2pts)	Insuficiente (0pts)
Programación (5 pts)	Selecciona las estructuras de datos adecuadas e implementa los algoritmos necesarios para satisfacer el 100 % de los requerimientos técnicos del programa.	Selecciona las estructuras de datos adecuadas e implementa los algoritmos necesarios para satisfacer al menos el 80 % de los requerimientos técnicos del programa.	Selecciona las estructuras de datos adecuadas e implementa los algoritmos necesarios para satisfacer al menos el 65 % de los requerimientos técnicos del programa.	Selecciona las estructuras de datos adecuada e implementa menos del 65 % de los requerimientos técnicos del programa.
Funcionamiento (5 pts)	El proyecto cumple con el 100% de las funcionalidades esperadas y sin errores	El proyecto cumple con al menos el 80% de las funcionalidades esperadas y no presenta errores que alteren significativamente el resultado.	El proyecto cumple con al menos el 65% de las funcionalidades esperadas. El programa presenta errores que afectan el resultado.	El proyecto cumple con menos del 65% de las funcionalidades esperadas. El programa presenta errores que afectan el resultado de manera significativa.
Legibilidad y documentación (5pts)	El código del proyecto está distribuido en al menos dos módulos. Mas del 90% de código está documentado y es legible.	El código del proyecto está distribuido en al menos dos módulos. Mas del 80% de código está documentado y es legible.	El código del proyecto está distribuido en al menos dos módulos. Mas del 65% de código está documentado y es legible.	El código del proyecto no está distribuido en al menos dos módulos. Menos del 50% de código está documentado y no es legible.
Presentación (5pts)	Los principales algoritmos utilizados, son explicados de manera clara y precisa. Contestan al 100% de las preguntas planteadas y reconocen las ventajas y limitaciones de las herramientas y recursos utilizados.	Los principales algoritmos utilizados, son explicados de manera clara y precisa. Contestan al 80% de las preguntas planteadas y reconocen las ventajas y limitaciones de las herramientas y recursos utilizados.	Los principales algoritmos utilizados, son explicados de manera clara y precisa. Contestan al 65% de las preguntas planteadas y reconocen las ventajas y limitaciones de las herramientas y recursos utilizados.	Los principales algoritmos utilizados, no son explicados de manera clara y precisa. Contestan solo al 50% de las preguntas planteadas y no reconocen las ventajas y limitaciones de las herramientas y recursos utilizados.