



# Universidad Nacional de La Matanza

**Algoritmos y estructuras de datos**

**Trabajo práctico:**

**Quien quiere C millonario**



## Necesidad

El departamento de recursos humanos de CodeInc, una startup de sistemas, está planeando el aniversario número 22 de la empresa. Para hacerlo divertido, se les ocurrió hacer un torneo en donde cada participante tenga que contestar una serie de preguntas. Si se contesta correctamente, se van sumando puntos y al final, la persona con mayor puntaje recibirá un premio.

## Reglas del juego

Al inicio de cada juego, se van a determinar la cantidad de jugadores, la cantidad de preguntas y el tiempo máximo por round que tiene un jugador para contestar.

Suponiendo que se define que van a jugar 3 personas, durante 5 rounds, 10 segundos por round:

- Se ingresan los nombres de los jugadores
- Se ingresa el nivel de dificultad (lo que se seleccione aplica para TODOS los jugadores de esa partida)
- Se hace un sorteo para determinar el orden de los jugadores
- Empieza el primer jugador sorteado
- El sistema le muestra la primera pregunta al jugador
- El jugador tiene 10 segundos para seleccionar la respuesta correcta (A, B, C o D). NO puede ingresar otro carácter que no sean esos.
- Si se le acaba el tiempo y no contesta, automáticamente se indicará que el tiempo finalizó y se le mostrará al jugador una nueva pregunta para el segundo round
- Si contesta, se le mostrará la siguiente pregunta
- En cada round se reinician los 10 segundos
- Así hasta completar los 5 rounds
- Luego juega el segundo participante sorteado
- En cada round, le tocará las MISMAS preguntas que al primer jugador y tendrá 10 segundos por round para contestar
- Idem para el jugador 3

Al final de la partida, se analizará la respuesta que dió cada jugador y se asignarán puntos por round de la siguiente manera:

- El jugador que haya contestado más rápido y correctamente recibe **+3 puntos**
- Todos los jugadores con que hayan contestado correctamente, pero que no hayan sido los más rápidos, reciben **+1 puntos**
- Si se ingresa una respuesta incorrecta el jugador recibe **-2 puntos**
- Si no se ingresa ninguna respuesta, no se recibe puntos
- En caso de que haya varios jugadores que hayan sido los más rápidos (tardaron lo mismo en contestar), recibirán c/u **+2 puntos**

Ganará la persona con mayor cantidad de puntos. En caso de que hayan 2 personas con la mayor cantidad de puntos (ejemplo: Jugador 1 y jugador 3



hicieron 10 puntos cada uno, mientras que el jugador 2 hizo 7 puntos), entonces el jugador 1 y 3 serán los ganadores.

Nota 1: Mínimo tiene que haber 4 rounds y cómo máximo puede haber 7 rounds.

Nota 2: Los niveles de dificultad son:

- Fácil
- Medio
- Difícil

Ejemplo:

Dificultad elegida: Media

Pregunta	Respuestas	Jugador 1	Jugador 2	Jugador 3
¿Cuál es la capital de Canadá?	A) Toronto B) Montreal C) Vancouver D) <b>Ottawa</b>	5 segundos en contestar <b>D (+1)</b>	2 segundos en contestar <b>D (+3)</b>	2 segundos en contestar A (-2)
¿Qué antiguo líder es conocido como el "Padre de Europa" por su papel en la unificación de gran parte de Europa occidental durante la Edad Media?	A) Julio César B) <b>Carlomagno</b> C) Alejandro Magno D) Napoleón Bonaparte	2 segundos en contestar A (-2)	8 segundos en contestar <b>B (+3)</b>	5 segundos en contestar A (-2)
¿Qué gas es el más abundante en la atmósfera terrestre?	A) Oxígeno B) Dióxido de carbono C) <b>Nitrógeno</b> D) Argón	5 segundos en contestar <b>C (+1)</b>	No contesta 0 puntos	6 segundos en contestar <b>C (+3)</b>
¿Quién es conocido como uno de los inventores de la bombilla eléctrica?	A) Nikola Tesla B) <b>Thomas Edison</b> C) Alexander Graham Bell D) Benjamin Franklin	7 segundos en contestar <b>B (+1)</b>	2 segundos en contestar <b>B (+2)</b>	2 segundos en contestar <b>B (+2)</b>
¿Quién escribió "Cien años de soledad"?	A) Julio Cortázar B) <b>Gabriel García Márquez</b> C) Mario Vargas Llosa D) Jorge Luis Borges	5 segundos en contestar <b>B (+3)</b>	4 segundos en contestar A (-2)	5 segundos en contestar C (-2)
<b>Total</b>		<b>4 puntos</b>	<b>6 puntos</b>	<b>-1 punto</b>



Al final de cada juego se deberá generar un informe que indique las preguntas de cada round, qué contestó cada jugador, cuál era la respuesta correcta, cuál fue el puntaje obtenido por pregunta, puntaje total y quién/es ganaron el premio.

## Consigna

Hacer un programa en C para jugar este juego.

Las configuraciones iniciales serán leídas de un archivo txt (config.txt) con el siguiente formato:

Rounds: 5

Tiempo por round: 10

Apenas se ingrese, deberá haber un menú de 2 opciones:

**[A] Jugar**

**[B] Salir**

Si alguien ingresa a “Jugar”, primero se le pedirá que cargue los nombres de las personas que van a jugar. Puede ingresar la cantidad de nombres que desee.

Una vez que termine de ingresarlos, tiene que elegir el nivel de dificultad del juego (fácil, medio, difícil) y entonces aparecerá por pantalla el orden en el que jugarán los jugadores (recordar que se elige aleatoriamente), la cantidad de rounds y el tiempo por round, y se le preguntará al primer jugador si está listo. En caso de que sí, inicia el juego.

Por round, se le muestra una pregunta al jugador y tiene la cantidad de segundos establecida para contestar, eligiendo la letra de la respuesta que cree correcta.

Al finalizar todos los rounds, se borrará de la pantalla todo lo que el primer jugador escribió, y se le preguntará al siguiente jugador si está listo. Cuando indique que sí, inicia su turno.

Al finalizar el turno de todos los jugadores, el sistema mostrará por pantalla quién ganó y por cuantos puntos, y a su vez va a generar un informe de que fue lo que sucedió en esa partida, indicando las preguntas de cada round, qué contestó cada jugador, cuál era la respuesta correcta, cuál fue el puntaje obtenido por pregunta, puntaje total y quién/es ganaron el premio. El nombre del archivo debe contener la fecha y la hora actual en el siguiente formato: YYYY-MM-DD-HH-mm. Ejemplo de nombre: informe-juego\_2024-02-01-12-20.txt

Además, se debe entregar un documento con diferentes lotes de prueba con el siguiente formato:

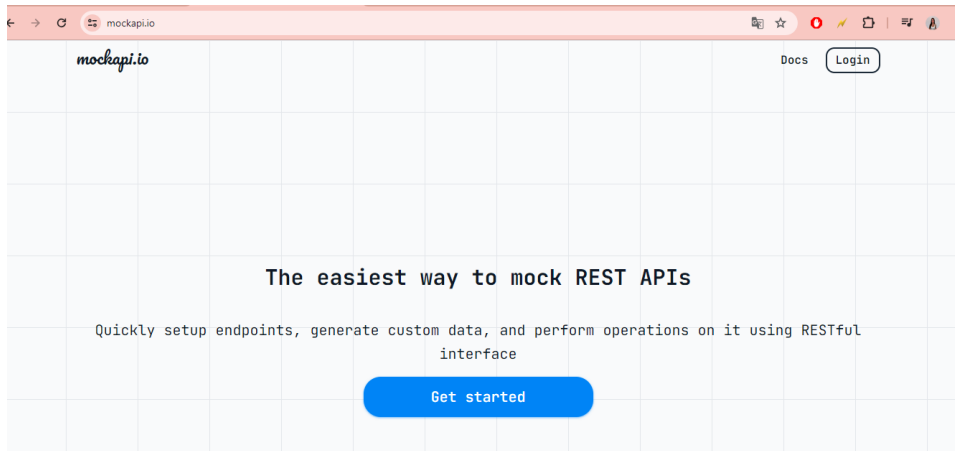


Descripción	Salida esperada	Salida obtenida
Se quiere probar qué es lo que pasaría si....	Se espera que....	La salida obtenida fue....

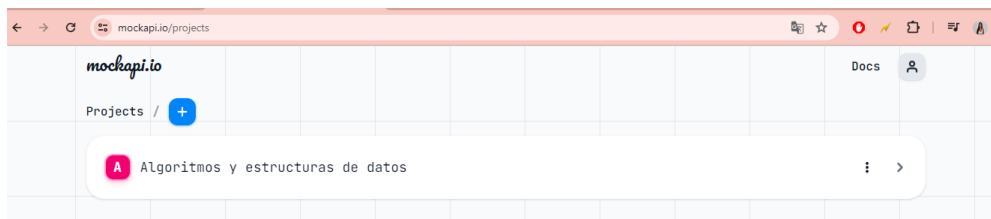
Mínimo se deben documentar 8 casos de prueba, con captura de pantalla de la salida obtenida.

## Preguntas

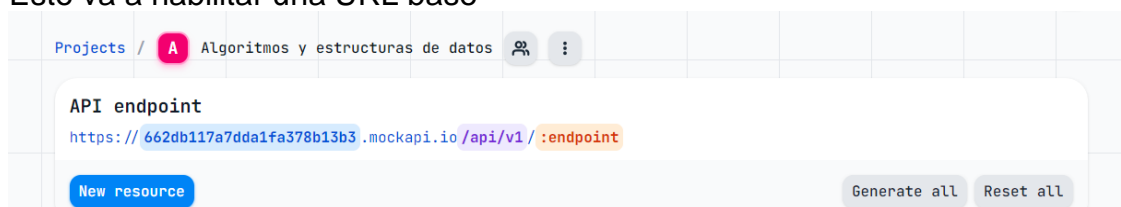
Para armar las preguntas que va a usar el juego, van a utilizar una API propia. Para esto, se van a crear una cuenta en la siguiente página: <https://mockapi.io/> y van a crear un endpoint que retorna todas las preguntas con las opciones de las respuestas + respuesta correcta + el nivel de dificultad de cada pregunta.



### 1. Crearse un proyecto



### 2. Esto va a habilitar una URL base



### 3. Hay que crear un “New Resource”, para así tener un endpoint y poder configurar lo que va a retornar



EDIT RESOURCE - PREGUNTAS

Schema

Define Resource schema, it will be used to generate mock data.

Field	Type
id	Object ID
pregunta	String
resp_correcta	String
opcion_1	String
opcion_2	String
opcion_3	String
nivel	Number

#### 4. Hay que cargar los datos

RESOURCE DATA - PREGUNTAS

Resource data

Edit/replace data for preguntas resource. Data must be an array and a valid JSON.

```
[
  {
    "pregunta": "¿Cuál es la capital de Francia?",
    "resp_correcta": "Paris",
    "opcion_1": "Londres",
    "opcion_2": "Madrid",
    "opcion_3": "Berlín",
    "nivel": 1,
    "id": "1"
  },
  {
    "pregunta": "¿Quién escribió 'Don Quijote de la Mancha'?",
    "resp_correcta": "Miguel de Cervantes",
    "opcion_1": "William Shakespeare",
    "opcion_2": "Gabriel García Márquez",
    "opcion_3": "Franz Kafka",
    "nivel": 1,
    "id": "2"
  }
]
```

#### 5. Si se hace un request a una URL, se devuelve los datos cargados

662db117a7dda1fa378b13b3.mockapi.io/api/v1/preguntas

Dar formato al texto ☒

```
[
  {
    "pregunta": "¿Cuál es la capital de Francia?",
    "resp_correcta": "Paris",
    "opcion_1": "Londres",
    "opcion_2": "Madrid",
    "opcion_3": "Berlín",
    "nivel": 1,
    "id": "1"
  },
  {
    "pregunta": "¿Quién escribió 'Don Quijote de la Mancha'?",
    "resp_correcta": "Miguel de Cervantes",
    "opcion_1": "William Shakespeare",
    "opcion_2": "Gabriel García Márquez",
    "opcion_3": "Franz Kafka",
    "nivel": 1,
    "id": "2"
  },
  {
    "pregunta": "¿Qué elemento químico tiene el símbolo 'O'?",
    "resp_correcta": "Oxígeno",
    "opcion_1": "Oro",
    "opcion_2": "Osmio",
    "opcion_3": "Olmo",
    "nivel": 2,
    "id": "3"
  },
  {
    "pregunta": "¿Cuál es el río más largo del mundo?",
    "resp_correcta": "Amazonas",
    "opcion_1": "Nilo",
    "opcion_2": "Yangtsé",
    "opcion_3": "Misisipi",
    "nivel": 2,
    "id": "4"
  }
]
```



Entonces, desde el código hacen una llamada HTTP a esa URL para traerse las preguntas, las filtran por el nivel de dificultad elegido, y se quedan con la cantidad de preguntas correspondientes según la cantidad de rounds. Ejemplo: Si el endpoint retorna 100 preguntas, tengo que quedarme con aleatoriamente 6 preguntas del nivel seleccionado (si es que se van a jugar 6 rounds)

### **Condiciones básicas para aprobar**

- ✓ Código prolijo, dividido en funciones
- ✓ Funciones lo más genéricas posibles
- ✓ Nombres significativos de variables
- ✓ 0 errores 0 warnings
- ✓ Que funcione mínimo para todos los casos de prueba que presentan