



Universidad Nacional de La Matanza

Algoritmos y estructuras de datos

Trabajo práctico:

Quasi tutti frutti



Necesidad

El departamento de marketing de la empresa Codeinc quiere armar una competencia entre sus empleados para estimular su lado más creativo. Les gustaría hacer un torneo en donde cada participante tenga que escribir la palabra más larga que se le ocurra con la letra que se le indique en cada round. Al final, se sumarán los puntos y 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 rounds y el tiempo 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 hace un sorteo para determinar el orden de los jugadores
- Empieza el primer jugador sorteado
- El sistema le muestra aleatoriamente una letra (Ejemplo: M)
- El jugador tiene 10 segundos para escribir la palabra más larga que se le ocurra con M (solo 1 palabra)
- Si se le acaba el tiempo y no contesta, automáticamente se indicará que el tiempo finalizó y se le mostrará al jugador una nueva letra para el segundo round
- Si contesta una palabra, se le mostrará una nueva letra (Ejemplo A) para que escriba la palabra más larga que se le ocurra con la letra A
- 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 letras que al primer jugador y tendrá 10 segundos por round para contestar
- Idem para el jugador 3

Al final de la partida, se analizará lo que escribió cada jugador y se asignarán puntos de la siguiente manera:

- El jugador con la palabra válida más larga (por round) recibe **+3 puntos**
 - o Entiéndase por válida que sea una palabra que exista en el diccionario y que empiece con la letra correcta.
- Todos los jugadores con palabras válidas que no se repitan (y que no sean las más largas), reciben **+1 punto**
- Si se ingresa una palabra inválida (que no exista en el diccionario o que no empiece con la letra correspondiente), el jugador recibe **-2 puntos**
- Si no se ingresa ninguna palabra, no se recibe puntos
- En caso de que se repita la palabra, cada jugador recibe **-1 punto** a excepción del primero que la haya ingresado, ese recibe los puntos correspondientes



- En caso de que la palabra sea diferente pero la longitud sea la misma, cada jugador recibe **+2 puntos**

Nota: Si por ejemplo la letra es 'A' y la palabra ingresada es "árbol" o "arbol", la palabra se cuenta cómo válida. No influyen las tildes para considerarla válida o no.

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.

Ejemplo:

Letra	Jugador 1	Jugador 2	Jugador 3
M	Manzana (+1)	Municipalidad (+3)	Mano (+1)
A	Árbol (+1)	Arándano (+3)	Aveja (-2)
C	Casamiento (+3)	Casas (+1)	Casa (+1)
T	Tijera (+1)	Tatuaje (+3)	Tatuaje (-1)
P	Pinza (+2)	Perro (+2)	- (0)
Total	7 puntos	12 puntos	-1 punto

Al final de cada juego se deberá generar un informe que indique las letras de cada round, qué contestó cada jugador, cuál fue el puntaje obtenido por palabra, 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, 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 letra aleatoria y tiene la cantidad de segundos establecida para contestar. La letra no se puede repetir entre los diferentes rounds.

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 letras de cada round, qué contestó cada jugador, cuál fue el puntaje obtenido por palabra, 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.

Validación de palabras

Para comprobar que una palabra sea válida, la misma tiene que empezar con la letra correspondiente y además ser una palabra que exista en el diccionario español. Es decir, si la letra es M y la palabra que escriben es “Mother”, se tendrían que restar puntos ya que mother no existe en el diccionario.

Para hacer dicha validación van a utilizar una API. Una API es una interfaz que permite a 2 componentes de software comunicarse entre sí. (No se pierdan la explicación en clase, en donde se les va a explicar con un poco más en profundidad). En el mundo laboral, usar APIs o construirlas es algo de todos los días (si están en la parte de desarrollo).

En este caso, TODOS van a usar la API que ofrece PONS (<https://en.pons.com/p/online-dictionary/developers/api>), en donde la misma les va a permitir usarla hasta 1000 veces gratis en el mes. (Si superan ese límite, creen otra cuenta)



Getting started is easy

- Already registered on PONS.com? Within your account you just need to fill out the [API registration form](#)
- Confirm that you agree to our [Terms and Conditions](#)
- Use our [documentation](#) for an easy implementation of the PONS Online Dictionary API
- The results are presented via our PONS Online Dictionary (POD) API in the look and feel of your website
- The POD-API is provided to you directly with a free quota of 1000 reference queries per month

H

If y
on
lice
pri
on
ou

1. Se tienen que registrar en el sistema
(<https://account.pons.com/es/public/signup>)
2. Tienen que crear una cuenta de API
(https://en.pons.com/open_dict/public_api?logged=1)
3. Lo de arriba les va a generar un secret, que lo van a usar en los “requests” que hagan

- 4.
5. Usan ese secret para los llamados http
6. Link a la documentación:
<https://en.pons.com/p/files/uploads/pons/api/api-documentation.pdf>

Haciendo una prueba rápida con Postman (un programa que se usa para llamar APIs mediante una interfaz gráfica), veo que si mando “mother” me devuelve estado 204 (no content), pero que si mando mamá, me devuelve algo (status code 200) y eso ya me indica que es una palabra válida que existe en el diccionario



UNLaM / PONS

GET <https://api.pons.com/v1/dictionary?l=dees&q=mother> Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

<input type="checkbox"/> Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> l	dees			
<input checked="" type="checkbox"/> q	mother			
<input type="checkbox"/> in	es			
<input type="checkbox"/> Key	Value	Description		

Body Cookies (5) Headers (13) Test Results

Status: 204 No Content Time: 413 ms Size: 863 B Save as example

Pretty Raw Preview Visualize JSON

UNLaM / PONS

GET <https://api.pons.com/v1/dictionary?l=dees&q=mama> Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

Query Params

<input type="checkbox"/> Key	Value	Description	...	Bulk Edit
<input checked="" type="checkbox"/> l	dees			
<input checked="" type="checkbox"/> q	mama			
<input type="checkbox"/> in				
<input type="checkbox"/> Key	Value	Description		

Body Cookies (5) Headers (14) Test Results

Status: 200 OK Time: 1016 ms Size: 8.21 KB Save as example

Pretty Raw Preview Visualize JSON

```
1 [
2   {
3     "lang": "es",
4     "hits": [
5       {
6         "type": "entry",
7         "opendict": false,
8         "xoms": [
9           {
10            "headword": "mama",
```

Cualquier duda que tengan con respecto a este punto, la pueden consultar por el foro, que algunos de los docentes les vamos a responder

Condiciones básicas para aprobar

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



Suma puntos extra

- ✓ Hacer árbol binario en donde sea lo más óptimo 😊