

Memoria EJ 5

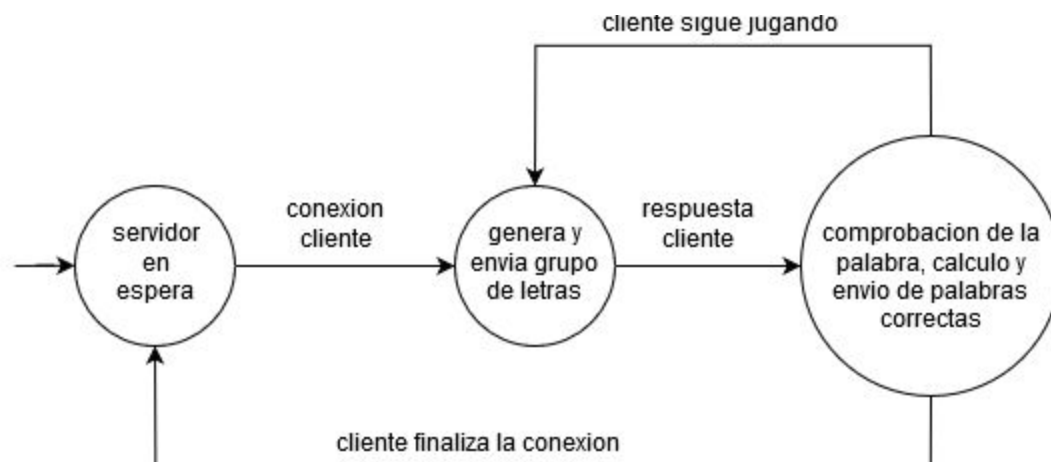
Noviembre del 2019

Descripción de la aplicación

La aplicación se basa en el paradigma cliente/servidor para dar soporte a un juego, en el que el servidor genera un conjunto de letras, que indica al cliente, y éste le devuelve una palabra que pueda ser formada con el conjunto de letras. El servidor procesará la palabra, indicando si es correcta, en cuyo caso, indicará también su puntuación, o si la palabra es incorrecta.

Finalmente, el servidor indicará también las soluciones que ha encontrado y por último mostrará la mejor solución encontrada. Finalmente preguntará el cliente si desea jugar de nuevo, en cuyo caso, el proceso comenzará de nuevo.

Diagrama de estados del servidor



Mensajes que intervienen

SERVIDOR

CÓDIGO	CUERPO	DESCRIPCIÓN
001	GROUP + [GRUPO DE LETRAS]	Mensaje enviado al cliente con las letras generadas
002	ERROR + "PALABRA INCORRECTA"	Mensaje de error cuando la palabra recibida no es correcta
003	ANSWER + [GRUPO PALABRAS]	Mensaje enviado al cliente con las palabras calculadas
004	AGAIN? + "JUGAR DE NUEVO?"	Mensaje enviado al cliente para preguntar si juega de nuevo
020	WORD + [PALABRA + PUNTUACIÓN]	Mensaje enviado al cliente con su palabra y la puntuación conseguida

CLIENTE

CÓDIGO	CUERPO	DESCRIPCIÓN
101	PLAY	Mensaje enviado al servidor para indicar que quiere jugar (al principio o para jugar de nuevo)
102	WORD + [palabra]	Mensaje enviado al servidor para especificar la palabra solución encontrada por el cliente
103	FIN	Mensaje enviado al servidor para indicar que no se quiere jugar otra vez

Capturas + evaluación de la aplicación

```
roman@XPS15-Román:/mnt/c/Users/roman/Desktop/EJ5$ java ClienteTCP
Quiero jugar
Estas son las letras con las que juegas:
[ A , P , S , S , F , E , A , R ]          Cual es tu palabra?
```

El cliente indica al servidor que quiere jugar, y el servidor devuelve la cadena de letras con la que formar la palabra. Mientras tanto, en el servidor:

```
roman@XPS15-Román:/mnt/c/Users/roman/Desktop/EJ5$ java ServidorIterativo
Leyendo palabras...
HECHO
```

El socket se ha abierto, cargando el diccionario de palabras asignado, se ha creado una instancia del programa y se ha calculado la bolsa de letras, sacando una lista char (letras) que envía al cliente.

```
Estas son las letras con las que juegas:
[ A , P , S , S , F , E , A , R ]          Cual es tu palabra?
afear
afear    Puntuacion: 5
```

El cliente especifica la palabra que forma con dichas letras y la envía al servidor. Éste devuelve la palabra junto a su puntuación si es correcta, o un error si es incorrecta.

```
Estas son las letras con las que juegas:
[ E , E , M , C , T , I , A , G ]          Cual es tu palabra?
garganta
*****PALABRA INCORRECTA*****
```

El servidor envía las palabras que ha encontrado con esas posibilidades en el diccionario, y el cliente las muestra:

```
Estas son las soluciones del servidor:
a, afe, afear, afer, afra, apea, apear, apres, apresa, ara, area, arepa, ares, arpa, as, asa, asar, asear, asesar, aspa,
aspar, aspera, e, ea, epa, era, es, esa, esas, fa, far, fara, fares, farpa, farsa, fas, fase, fe, fea, fer, frase, fres,
fresa, paf, par, para, paresa, pasa, pasar, pase, pasear, pasera, pe, pea, per, pera, persa, pesa, pesar, pre, prea, p
resa, rafa, rafe, rapa, rape, ras, rasa, raspa, re, rea, reps, res, sapa, sarape, se, ser, sera, serpa, ses,
La mejor solución es:
sarape    Puntuacion: 6
```

También envía la mejor solución que ha encontrado. Finalmente el servidor pregunta si se va a jugar de nuevo, y el cliente responde. Si es una negativa, el cliente cierra la conexión y acaba el proceso, en caso contrario se vuelve a jugar de nuevo:

```
La mejor solución es:
emetica Puntuacion: 7
Quieres jugar de nuevo? [S/n]
s
Has decidido jugar de nuevo
Quiero jugar
Estas son las letras con las que juegas:
[ U , D , B , T , L , T , E , O ]          Cual es tu palabra?
```