

PASAPALABRA

UTILIZANDO CLIENTE-SERVIDOR

Miguel Ángel López Robles

Jaime Frías Funes

TABLA DE CONTENIDO

Contenido

Pasapalabra_____	1
Diagrama de estados_____	2
Desarrollo_____	3
Capturas del funcionamiento del programa _____	5

Pasapalabra

RESUMEN

Nos basamos en el programa popular “Pasapalabra” y trasladamos el juego a un simple paso de mensajes basado en el paso de mensajes de Cliente-Servidor. El usuario deberá completar un “rosco” indicándole definiciones de palabras que deberá de acertar. El objetivo es que este acierte todas las definiciones. Aunque no se acierte, el juego seguirá hasta que se llegue a la última letra del rosco.

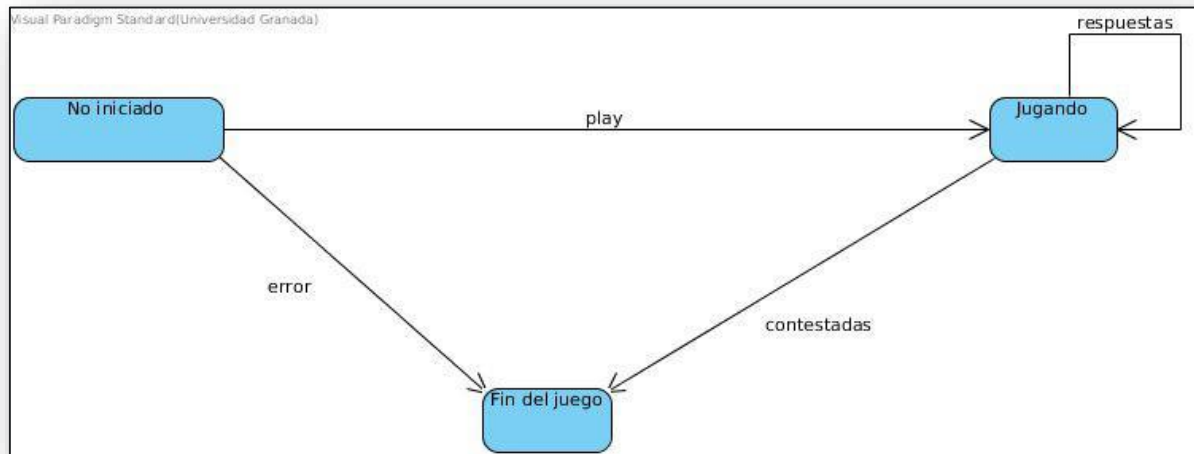
RECURSOS UTILIZADOS

Nos basamos en el paso de mensajes utilizando **TCP** ya que necesitamos que no se pierda información en el envío de paquetes ya que las definiciones deben de llegar completas.

Hemos utilizado un **servidor concurrente** ya que permite que varios jugadores completen el rosco a la vez.

El paso de mensajes se basa en **modo texto** utilizando la JDK ya que las definiciones eran demasiado largas y no podemos prever la longitud de la cadena.

Diagrama de estados



El juego se inicia y el cliente envía *play* y una vez que el servidor confirme enviando *COMENCEMOS* entra en bucle en el estado "**Jugando**" mientras se va completando el roscó enviando las palabras al servidor, si el jugador las acierta el servidor enviará *CORRECTO*, hasta que se contestan todas las letras y el servidor envía *FIN DEL JUEGO*, y se llega al estado "**Fin del juego**". También se puede llegar a este último estado directamente sin iniciar el juego mediante algún error.

Desarrollo

ESTRUCTURAS DE DATOS UTILIZADAS

Hemos utilizado tres arrays:

- **Definiciones.** Conjunto de definiciones para cada letra.
- **Respuestas.** Conjunto de palabras correctas para cada letra.
- **Contestadas.** Conjunto de enteros que representan la validez de cada respuesta del usuario.

Además, nos hemos valido de variables de enteros que representaban el puntero seleccionando cada letra durante el funcionamiento del juego, las respuestas correctas, entre otros.

SERVIDOR

El servidor recibirá la petición de los clientes y lanzará una hebra del procesador de modo que puede atender peticiones concurrentemente. El servidor solo se encargará de lanzar nuevas instancias del procesador donde se realiza realmente el juego.

PROCESADOR

Se añaden todas las definiciones y las respuestas correctas. Y el procedimiento de este se basa en recibir el mensaje del cliente para empezar el juego y enviar una confirmación. Se enviará la definición y si la respuesta es la correcta se enviará un mensaje que lo confirme, si no, lo contrario, así hasta que el número de contestadas sea el total de letras del roco. Se le permitirá usar la palabra clave 'pasapalabra' para pasar de pregunta.

CLIENTE

Se envía el código *play* al servidor y si el servidor responde correctamente se comienza el juego, y se empiezan a recibir definiciones por parte del servidor y se enviarán las palabras y se recibirá una confirmación de la validez de cada respuesta, todo esto se repetirá hasta que el servidor indique el fin del juego. En el cliente se imprimirá en cada turno el roco actual indicando con 0 que no se ha contestado correctamente esa letra y con 1 que sí, y así hasta completar el roco sin quedar letras.

CÓDIGO PROCESADOR

```
if(peticion.equals("play")){
    //mandamos el mensaje de comenzamos para confirmar que atendemos la peticion
    outPrinter.println("comenzemos");

    //bucle que solo acabara cuando se hayan respondido todas
    while(contar_completas < 25){
        //imprimir el rosco
        outPrinter.println(imprimirRosco());
        // letra por la que vamos a jugar
        char letra = convierteLetra(puntero);

        //mandamos el mensaje de la pregunta
        outPrinter.println(letra +": " + definiciones.get(puntero));

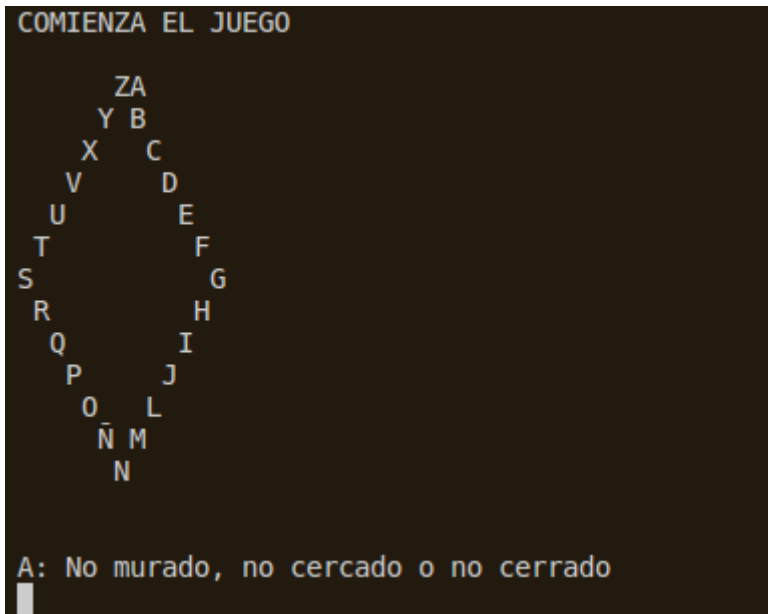
        //leemos la respuesta
        peticion=inReader.readLine();

        //comprobacion de si la respuesta es correcta o pasapalabra
        if(peticion.equals(respuestas.get(puntero))){
            contar_completas++;
            acertadas++;
            contestadas.set(puntero,1);
            if(contar_completas == 25){
                outPrinter.println("FIN DEL JUEGO");
            }
            else{
                outPrinter.println("CORRECTO!!!!!!!!!!");
            }
        }
        else if( !peticion.equals("pasapalabra")){
            contestadas.set(puntero,2);
            contar_completas++;
            if(contar_completas == 25){
                outPrinter.println("FIN DEL JUEGO");
            }
            else{
                outPrinter.println("OHHHHH ERRORRR!! La respuesta correcta era " + respuestas.get(puntero));
            }
        }
        else{
            outPrinter.println("CONTINUAMOS CON LA SIGUIENTE");
        }
        //System.out.println(contar_completas);
        //buscamos la siguiente palabra
        siguientePuntero();
    }
}
```

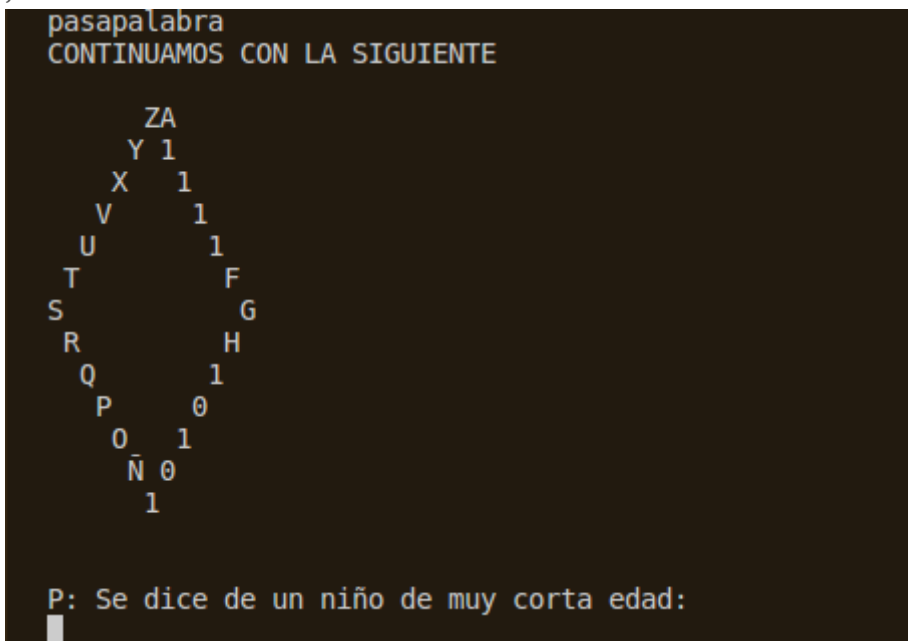
```
//pasamos el resultado del rosco
outPrinter.println(imprimirRosco());
if(acertadas == 25){
    outPrinter.println("Enhorabuena has ganado y te llevas el bote");
}
else{
    outPrinter.println("Lo siento has perdido pero has acertado " + Integer.toString(acertadas));
}
```

Capturas del funcionamiento del programa

INICIO DEL JUEGO



JUEGO EN PROCESO PASANDO PALABRA



PASAPALABRA

FINAL DEL JUEGO CON ÉXITO

```

      11
     1 1
    1   1
   1     1
  1       1
 1         1
1          1
1         1
1         1
 1       1
   1     1
    1   1
     1 1
      1

```

FINAL DEL JUEGO PERDIENDO

```

FIN DEL JUEGO
El resultado del rosco es:

  11
 1 1
1  1
 1  1
1  0
 1  0
 0  0
 0  0
 0  0
 0  0
 1  1
 1 1
  1

```