



Instituto Politécnico Nacional.
Escuela Superior De Cómputo.



Materia:
Aplicaciones Para Comunicación En
Red.

Tema:
Juego del Ahorcado.
(Tarea 03)

Profesor:
Axel Ernesto Moreno Cervantes.

Alumno:
Mario Alberto Miranda Sandoval.

Grupo:
3CM5

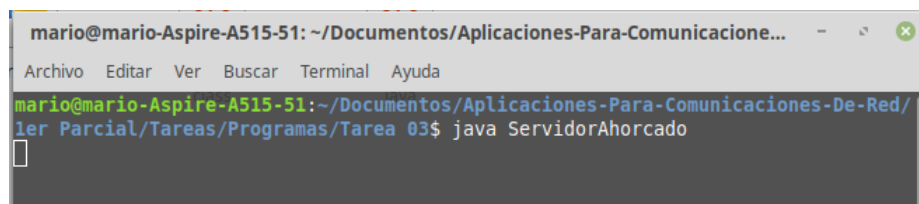
Programar el juego del ahorcado, utilizando sockets de datagrama, entre dos lenguajes de programación.

En la siguiente tarea, se pretende realizar un juego de ahorcado, donde debe existir una comunicación por sockets de datagrama entre cliente y servidor.

Por el motivo que aún no se usa concurrencia, optaremos porque el servidor solo almacene las palabras y las envíe, mientras que, en el cliente, solo exista la lógica del juego.

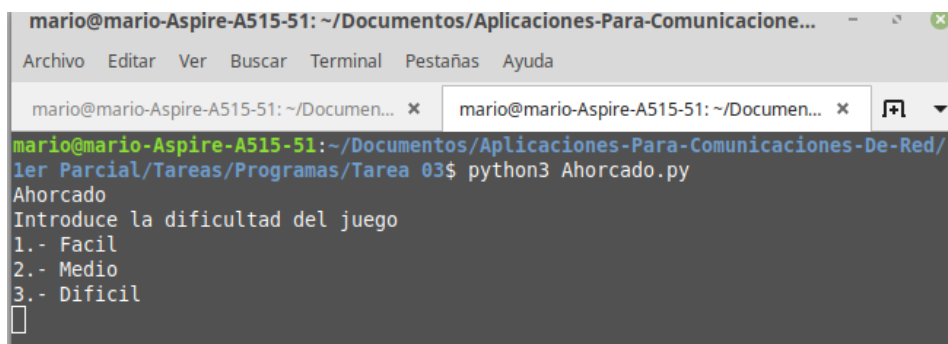
Para esta tarea, no se creo una interfaz de usuario, sino todo se lleva en el modo consola, además, que los lenguajes de programación escogidos fueron Java y Python.

Pruebas.



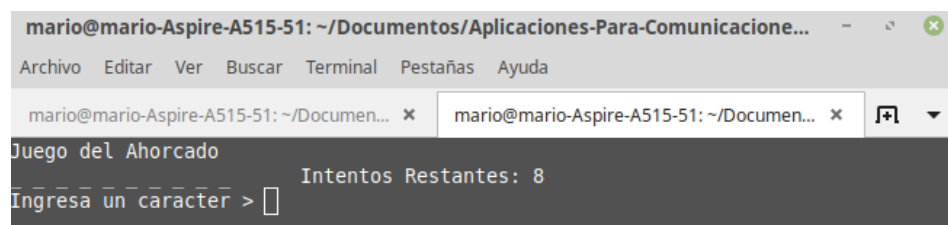
```
mario@mario-Aspire-A515-51: ~/Documentos/Aplicaciones-Para-Comunicacione...
Archivo Editar Ver Buscar Terminal Ayuda
mario@mario-Aspire-A515-51:~/Documentos/Aplicaciones-Para-Comunicaciones-De-Red/ler Parcial/Tareas/Programas/Tarea 03$ java ServidorAhorcado
```

Corriendo Servidor



```
mario@mario-Aspire-A515-51: ~/Documentos/Aplicaciones-Para-Comunicacione...
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
mario@mario-Aspire-A515-51:~/Documentos/Aplicaciones-Para-Comunicaciones-De-Red/ler Parcial/Tareas/Programas/Tarea 03$ python3 Ahorcado.py
Ahorcado
Introduce la dificultad del juego
1.- Facil
2.- Medio
3.- Dificil
```

Corriendo Cliente (Iniciando Juego).



```
mario@mario-Aspire-A515-51: ~/Documentos/Aplicaciones-Para-Comunicacione...
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
Juego del Ahorcado
Intentos Restantes: 8
Ingresa un caracter >
```

Vista del juego

```
mario@mario-Aspire-A515-51: ~/Documentos/Aplicaciones-Para-Comunicacione...
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
mario@mario-Aspire-A515-51: ~/Documen... x mario@mario-Aspire-A515-51: ~/Documen... x
Juego del Ahorcado
_ _ _ _ _
_ _ _ _ _ Intentos Restantes: 7
_ _ _ _ _ Intentos Restantes: 7
Ingresa un caracter > h
_ _ _ _ _ Intentos Restantes: 6
Ingresa un caracter > y
_ _ _ _ _ Intentos Restantes: 5
Ingresa un caracter > 
```

```
mario@mario-Aspire-A515-51: ~/Documentos/Aplicaciones-Para-Comunicacione...
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
mario@mario-Aspire-A515-51: ~/Documen... x mario@mario-Aspire-A515-51: ~/Documen... x
Juego del Ahorcado
_ _ _ _ _
_ _ _ _ _ Intentos Restantes: 5
_ _ _ _ _ Intentos Restantes: 5
Ingresa un caracter > q
_ _ _ _ _ Intentos Restantes: 4
Ingresa un caracter > r
_ _ _ _ _ Intentos Restantes: 3
Ingresa un caracter > t
_ _ _ _ _ Intentos Restantes: 2
Ingresa un caracter > 
```

```
mario@mario-Aspire-A515-51: ~/Documentos/Aplicaciones-Para-Comunicacione...
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
mario@mario-Aspire-A515-51: ~/Documen... x mario@mario-Aspire-A515-51: ~/Documen... x
Juego del Ahorcado
_ _ _ _ _
_ _ _ _ _ Intentos Restantes: 5
_ _ _ _ _ Intentos Restantes: 5
Ingresa un caracter > q
_ _ _ _ _ Intentos Restantes: 4
Ingresa un caracter > r
_ _ _ _ _ Intentos Restantes: 3
Ingresa un caracter > t
_ _ _ _ _ Intentos Restantes: 2
Ingresa un caracter > m
_ _ _ _ _ Intentos Restantes: 1
Ingresa un caracter > s
_ _ _ _ _ Intentos Restantes: 0
Palabra a buscar: licantropo
mario@mario-Aspire-A515-51:~/Documentos/Aplicaciones-Para-Comunicaciones-De-Red/
ler Parcial/Tareas/Programas/Tarea 03$ 
```

Juego Perdido.

```
mario@mario-Aspire-A515-51: ~/Documentos/Aplicaciones-Para-Comunicacione...
Archivo Editar Ver Buscar Terminal Pestañas Ayuda
mario@mario-Aspire-A515-51: ~/Documen... x mario@mario-Aspire-A515-51: ~/Documen... x
Juego del Ahorcado
p u s h i t t _ t h _ l i m i t Intentos Restantes: 10
_ _ _ _ _ Intentos Restantes: 10
Ingresa un caracter > 
```

```
mario@mario-Aspire-A515-51: ~/Documentos/Aplicaciones-Para-Comunicacione...
Archivo  Editar  Ver  Buscar  Terminal  Pestañas  Ayuda
mario@mario-Aspire-A515-51: ~/Documen... x  mario@mario-Aspire-A515-51: ~/Documen... x  [?] v
Juego del Ahorcado
push it to the limit  Intentos Restantes: 8
Felicidades haz ganado
mario@mario-Aspire-A515-51:~/Documentos/Aplicaciones-Para-Comunicaciones-De-Red/
ler Parcial/Tareas/Programas/Tarea 03$
```

Juego Ganado.

La dificultad inicial, estuvo dada por el hecho que desconocía el como enviar datos mediante datagramas entre dos distintos lenguajes, ya que en los primeros intentos de envío, el mensaje llegaba de Python a Java, pero a la hora del mensaje de vuelta este no regresaba nada, así, que después de investigar la solución estuvo en colocar en el puerto al momento de enviar el datagrama lo siguiente `paquete.getPort()`, con esto se logró la comunicación bidireccional.

Los problemas siguientes, se debieron más por la lógica del juego tanto en las validaciones y el formato mostrado en consola.

Código.

ServidorAhorcado.java

```
1. import java.net.*;
2.
3. public class ServidorAhorcado {
4.     public static void main(String[] args) {
5.         String palabras [][] = {
6.             {"perro", "hijo", "coche", "hombre"},
7.             {"ferrocarril", "autobus", "desgracia", "licantropo"},
8.             {"anita lava la tina", "ese compa ya esta muerto", "push it to the lim
it", "adios mundo cruel"}
9.         };
10.        try {
11.            while(true) {
12.                DatagramSocket servidor = new DatagramSocket(9999);
13.                DatagramPacket paquete = new DatagramPacket(new byte[1024], 1024);
14.
15.                servidor.receive(paquete);
16.                int num = Integer.parseInt(new String(paquete.getData(), 0, paquet
e.getLength()));
17.                String seleccion = palabras[num - 1][numero()];
18.                byte[] s = seleccion.getBytes();
19.                DatagramPacket vuelta = new DatagramPacket(s, s.length, InetAdres
s.getByname("localhost"), paquete.getPort());
20.                servidor.send(vuelta);
21.                servidor.close();
22.            }
23.        } catch (Exception e) { e.printStackTrace(); }
24.    }
```

```

25.
26.     private static int numero() {
27.         return (int) (Math.random() * ((3 - 1) + 1) + 1);
28.     }
29. }

```

Ahorcado.py (Usar Python 3)

```

1. import socket
2. import sys
3. import os
4.
5. word = ""#palabra recibida
6. intentos = 0
7. caracteresQuemados = []
8. posiciones = []
9. impresion = []
10.
11. def llenarLista():
12.     global impresion
13.     global word
14.
15.     for i in word:
16.         impresion.append("0")
17.
18.     #print(impresion[:])
19.
20. def dificultad(dificultad):
21.     try:
22.         global word
23.         cliente = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
24.         cliente.sendto(bytes(str(dificultad), "utf-8"), ("localhost", 9999))
25.         palabraRecibida, addr = cliente.recvfrom(1024)
26.         word = palabraRecibida.decode("utf-8")
27.     finally:
28.         cliente.close()
29.
30. def establecerIntentos(decision):
31.     global intentos
32.
33.     if decision == 1:
34.         intentos = 5
35.     elif decision == 2:
36.         intentos = 8
37.     elif decision == 3:
38.         intentos = 10
39.
40. def formatoInicial():
41.     global word
42.     global intentos
43.
44.     os.system("clear")
45.
46.     print("Juego del Ahorcado")
47.
48.     for i in word:
49.         if i != " ":
50.             print("_", end = " ")
51.         else:
52.             print(" ", end = " ")

```

```

53.
54.     print("      Intentos Restantes: " + str(intentos))
55.
56. def checarQuemados(caracter):
57.     global caracteresQuemados
58.     #print(caracteresQuemados[:])
59.     return (caracter in caracteresQuemados)
60.
61.
62. def formato(caracter):
63.     global word
64.     global posiciones
65.     global impresion
66.     global intentos
67.     #print(posiciones[:])
68.     contador = 0
69.     cont = 0
70.     j = 0
71.
72.     os.system("clear")
73.
74.     print("Juego del Ahorcado")
75.
76.     for i in word:
77.         if cont == posiciones[contador] :
78.             impresion.pop(cont)
79.             impresion.insert(cont, caracter)
80.             if contador == len(posiciones) - 1:
81.                 contador -= 1
82.                 contador += 1
83.             cont += 1
84.
85.     #print(impresion[:])
86.     for i in word:
87.         if i == impresion[j]:
88.             print(impresion[j], end = " ")
89.         else :
90.             if i != " ":
91.                 print("_", end = " ")
92.             else:
93.                 print(" ", end = " ")
94.         j += 1
95.
96.     posiciones.clear()
97.     print("      Intentos Restantes: " + str(intentos))
98.
99. def validar():
100.     global word
101.     global impresion
102.
103.     #print(impresion[:])
104.     listaFinal = []
105.
106.     for i in impresion:
107.         if i != "0":
108.             listaFinal.append(i)
109.         else:
110.             listaFinal.append(" ")
111.
112.     #print(listaFinal[:])
113.     cadena = "".join(listaFinal)

```

```

114.         #print(cadena)
115.
116.         j = 0
117.         bandera = True
118.         for i in word:
119.             if i != cadena[j] :
120.                 bandera = False
121.                 break
122.             j += 1
123.
124.         if bandera:
125.             print("Felicidades haz ganado")
126.             sys.exit(0)
127.
128.     def logica():
129.         global word
130.         global intentos
131.         global caracteresQuemados
132.         global posiciones
133.
134.         #os.system("clear")
135.
136.         bandera = False
137.         posicion = 0
138.
139.         caracter = str(input("Ingresa un caracter > "))
140.         caracter = caracter.lower()
141.
142.         if len(caracter) == 0 or len(caracter) > 1 :
143.             print("Caracter no valido")
144.             sys.exit(-1)
145.
146.         if not checarQuemados(caracter):
147.             for i in word:
148.                 if caracter == i:
149.                     posiciones.append(posicion)
150.                     posicion += 1
151.                     bandera = True
152.                 else :
153.                     posicion += 1
154.             else:
155.                 bandera = False
156.
157.         if bandera:
158.             formato(caracter)
159.             intentos += 1
160.             validar()
161.
162.         caracteresQuemados.append(caracter)
163.         intentos -= 1
164.         print("      Intentos Restantes: " + str(intentos))
165.
166.     def main():
167.         global word
168.         global intentos
169.         print("Ahorcado")
170.         decision = int(input("Introduce la dificultad del juego \n1.- Facil\n2
.- Medio\n3.- Dificil\n"))
171.         dificultad(decision)
172.         #print(word)
173.         establecerIntentos(decision)

```

```
174.         print("Comienza el juego")
175.         print(intentos)
176.         llenarLista()
177.         formatoInicial()
178.
179.         while intentos != 0:
180.             logica()
181.
182.         print("Palabra a buscar: " + str(word))
183.
184.     main()
```