

Instituto Politécnico Nacional

Escuela Superior de Cómputo

*Aplicaciones para comunicaciones de red*

*“Practica 2 – Sopa de letras”*

**Grupo:** 3CM15

**Integrantes:**

* Ramírez Olvera Guillermo.
* Sánchez Méndez Edmundo Josue.

**Fecha de entrega:** 26 de marzo de 2021

**Profesor:** Moreno Cervantes Axel Ernesto

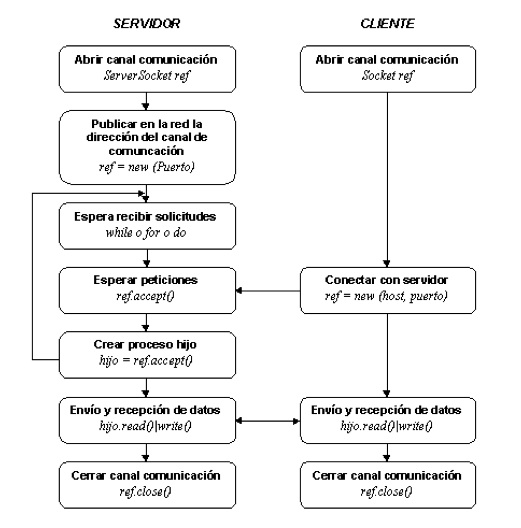


En la actualidad, muchos de los procesos que se ejecutan en una computadora requieren obtener o enviar información a otros procesos que se localizan en una computadora diferente.  Para lograr esta comunicación se utilizan los protocolos de comunicación TCP y UDP.

El protocolo TCP (Transmission Control Protocol) establece un conducto de comunicación punto a punto entre dos computadoras, es decir, cuando se requiere la transmisión de un flujo de datos entre dos equipos, el protocolo TCP establece un conducto exclusivo entre dichos equipos por el cual los datos serán transmitidos y este perdurará hasta que la transmisión haya finalizado, gracias a esto TCP garantiza que los datos enviados de un extremo de la conexión lleguen al otro extremo y en el mismo orden en que fueron enviados. Las características que posee TCP hacen que el protocolo sea conocido como un protocolo orientado a conexión.

Los sockets son una forma de comunicación entre procesos que se encuentran en diferentes máquinas de una red, los sockets proporcionan un punto de comunicación por el cual se puede enviar o recibir información entre procesos.

Los sockets tienen un ciclo de vida dependiendo si son sockets de servidor, que esperan a un cliente para establecer una comunicación, o socket cliente que busca a un socket de servidor para establecer la comunicación.



**Desarrollo de la practica:**

En esta práctica se implementará una aplicación en donde el servidor mandará una al cliente una sopa de letras que se resolverá mediante el uso de las coordenadas de las palabras y en caso de resolverla de manera exitosa su puntuación será guardada en el servidor con base en el tiempo de solución.

Para la implementación de esta práctica se realizaron 5 clases en java: Cliente\_O, LogicaJuego, Objeto, Servidor\_O y Sopa.

**Clase Objeto:**

En nuestra clase objeto tenemos dos atributos llamados palabras y sopa, las cuales nos permitirá facilitar el trabajo posteriormente.

1. import java.io.\*;
3. class Objeto implements Serializable {
5. private String palabras;
6. private char[][] sopa;

9. public Objeto(String palabras) {
10. this.palabras = palabras;
11. }
13. public Objeto(char[][] sopa) {
14. this.sopa = sopa;
15. }
16. public char[][] getSopa() {
17. return sopa;
18. }
20. public void setSopa(char[][] sopa) {
21. this.sopa = sopa;
22. }
23. public String getPalabras() {
24. return palabras;
25. }
27. public void setPalabras(String palabras) {
28. this.palabras = palabras;
29. }
30. }

**Clase LogicaJuego:**

En nuestra clase LogicaJuego tenemos todo lo necesario para el correcto funcionamiento de nuestra práctica. Primeramente, tenemos nuestro constructor en donde se creará toda nuestra sopa de letras excluyendo las letras que se colocan de manera aleatoria.

En nuestra funcion jugar tenemos la parte lógica del juego, en la cual nos permite mostrar la sopa de letras mediante el uso de la funcion mostrar y que también nos permite ingresar nuestras coordenadas de inicio y fin para poder solucionar nuestra sopa de letras. También nos permite ver que palabras nos faltan mediante la funcion encontrar y que palabras tenemos que buscar, ademas de que mediante la funcion marcar es como se encuentran nuestras palabras.

1. import java.io.Serializable;
2. import java.util.ArrayList;
3. import java.util.Arrays;
4. import java.util.List;
5. import java.util.Random;
6. import java.util.Scanner;
7. import sun.net.idn.Punycode;
9. public class LogicaJuego implements Serializable{
10. private char[][] sopa;
11. private String[] palabras;
12. private List<String> encontradas = new ArrayList<String>();
13. LogicaJuego(char[][] sopa,String[] palabras) {
14. Random r = new Random();
15. String alfabeto="abcdefghijklmnopqrstuvwxyz";
16. int num;
17. this.sopa=sopa;
18. this.palabras=palabras;
19. for (int i = 0; i < this.sopa.length; i++) {
20. for (int j = 0; j < this.sopa[i].length; j++) {
21. if(this.sopa[i][j]=='.'){
22. num = r.nextInt(25) + 1;
23. this.sopa[i][j]=alfabeto.charAt(num);
24. }
25. }
26. }
27. }
28. public double jugar(){
29. long inicio = System.currentTimeMillis();
30. long fin;
31. double tiempo;
32. int[] cordenadas={0,0,0,0};
33. String aux1,aux2;
34. String[] aux;
35. String enc;
36. Scanner escaneo = new Scanner(System.in);
37. while(palabras.length!=encontradas.size()){
38. mostrar();
39. System.out.println("Ingrese las cordenada 1");
40. aux1=escaneo.nextLine();
41. System.out.println("Ingrese las cordenada 2");
42. aux2=escaneo.nextLine();
43. aux=aux1.split(",");
44. cordenadas[0]=Integer.parseInt(aux[0]);
45. cordenadas[1]=Integer.parseInt(aux[1]);
46. aux=aux2.split(",");
47. cordenadas[2]=Integer.parseInt(aux[0]);
48. cordenadas[3]=Integer.parseInt(aux[1]);
49. enc=encontrar(cordenadas);
50. System.out.println(enc);
51. for(int i=0;i<palabras.length;i++){
52. if(palabras[i].equals(enc)){
53. marcar(cordenadas);
54. encontradas.add(enc);
55. }
56. }
57. }
58. System.out.println("Palabras por encontrar: "+Arrays.toString(palabras));
59. System.out.println("Palabras encontradas: "+encontradas);
60. System.out.println("Felicidades, has ganado");
61. fin = System.currentTimeMillis();
62. tiempo = (double) ((fin - inicio)/1000);
63. System.out.println("Su puntaje es: "+ tiempo);
64. return tiempo;
65. }
66. public void marcar(int[] cordenadas){
67. char letra;
68. if(cordenadas[0]==cordenadas[2]&&cordenadas[1]<cordenadas[3]){//derecha
69. for(int i=0;i<cordenadas[3]-cordenadas[1]+1;i++){
70. letra=sopa[cordenadas[0]][cordenadas[1]+i];
71. if(letra>96&&letra<123){
72. sopa[cordenadas[0]][cordenadas[1]+i]=(char) (letra-32);
73. }
74. }
75. }else if(cordenadas[1]==cordenadas[3]&&cordenadas[0]<cordenadas[2]){//abajo
76. for(int i=0;i<cordenadas[2]-cordenadas[0]+1;i++){
77. letra=sopa[cordenadas[0]+i][cordenadas[1]];
78. if(letra>96&&letra<123){
79. sopa[cordenadas[0]+i][cordenadas[1]]=(char) (letra-32);
80. }
81. }
82. }else if(cordenadas[0]==cordenadas[2]&&cordenadas[3]<cordenadas[1]){//izquierda
83. for(int i=0;i<cordenadas[1]-cordenadas[3]+1;i++){
84. letra=sopa[cordenadas[0]][cordenadas[1]-i];
85. if(letra>96&&letra<123){
86. sopa[cordenadas[0]][cordenadas[1]-i]=letra=(char) (letra-32);
87. }
88. }
89. }else if(cordenadas[1]==cordenadas[3]&&cordenadas[2]<cordenadas[0]){//arriba
90. for(int i=0;i<cordenadas[0]-cordenadas[2]+1;i++){
91. letra=sopa[cordenadas[0]-i][cordenadas[1]];
92. if(letra>96&&letra<123){
93. sopa[cordenadas[0]-i][cordenadas[1]]=(char) (letra-32);
94. }
95. }
96. }
97. }
98. public String encontrar(int[] cordenadas){
99. String aux;
100. char letra;
101. List<String> letters = new ArrayList<String>();
102. if(cordenadas[0]==cordenadas[2]&&cordenadas[1]<cordenadas[3]){//derecha
103. for(int i=0;i<cordenadas[3]-cordenadas[1]+1;i++){
104. letra=sopa[cordenadas[0]][cordenadas[1]+i];
105. letters.add(letra+"");
106. }
107. }else if(cordenadas[1]==cordenadas[3]&&cordenadas[0]<cordenadas[2]){//abajo
108. for(int i=0;i<cordenadas[2]-cordenadas[0]+1;i++){
109. letra=sopa[cordenadas[0]+i][cordenadas[1]];
110. letters.add(letra+"");
111. }
112. }else if(cordenadas[0]==cordenadas[2]&&cordenadas[3]<cordenadas[1]){//izquierda
113. for(int i=0;i<cordenadas[1]-cordenadas[3]+1;i++){
114. letra=sopa[cordenadas[0]][cordenadas[1]-i];
115. letters.add(letra+"");
116. }
117. }else if(cordenadas[1]==cordenadas[3]&&cordenadas[2]<cordenadas[0]){//arriba
118. for(int i=0;i<cordenadas[0]-cordenadas[2]+1;i++){
119. letra=sopa[cordenadas[0]-i][cordenadas[1]];
120. letters.add(letra+"");
121. }
122. }
123. aux=Arrays.toString(letters.toArray());
124. aux=aux.toLowerCase();
125. aux=aux.replace("[","");
126. aux=aux.replace("]","");
127. aux=aux.replace(",","");
128. aux=aux.replace(" ","");
129. return aux;
130. }
131. public void mostrar() {
132. System.out.println("Palabras por encontrar: "+Arrays.toString(palabras));
133. System.out.println("Palabras encontradas: "+encontradas);
134. for (int i = 0; i < sopa.length; i++) {
135. if (i < 10) {
136. System.out.print("0" + i + "|");
137. } else {
138. System.out.print(i + "|");
139. }
140. for (int j = 0; j < sopa[0].length; j++) {
141. if (j != sopa[0].length) {
142. System.out.print(" " + sopa[i][j]);
143. }
144. }
145. if (i < 10) {
146. System.out.println("|" + "0" + i);
147. } else {
148. System.out.println("|" + i);
149. }
150. }
151. }
152. }

**Clase Sopa:**

En nuestra clase Sopa nos permite generar la sopa de letras completas, es decir, incluyendo los caracteres que se usan como relleno.

1. import java.io.Serializable;
2. import java.util.\*;
4. public class Sopa implements Serializable {
6. Scanner sc = new Scanner(System.in);
7. private char[][] sopa;

10. private String[] palabras;
11. private int[] sitio = new int[]{-1, -1, -1};
12. String[] animales = {"oso", "abeja", "ballena", "perro", "caballo", "camello", "gato", "cerdo", "elefante", "foca","leon","tigre","lagartija","mono","lobo"};
13. String[] frutas = {"uva", "lima", "limon", "cereza", "arandano", "platano", "manzana", "sandia", "fresa", "naranja"};
14. String[] colores = {"rojo", "naranja", "verde", "negro", "blanco ", "cafe ", "amaraillo", "rosa ", "gris ", "azul"};
15. String[] prueba={"memo","jose","sui"};
16. Sopa(int nivel) {
17. this.sopa = new char[16][16];
18. for (int i = 0; i < sopa.length; i++) {
19. for (int j = 0; j < sopa[i].length; j++) {
20. sopa[i][j] = '.';
21. }
22. }
23. if (nivel == 1) {
24. this.palabras = animales;
25. } else if (nivel == 2) {
26. this.palabras = frutas;
27. } else if (nivel == 3) {
28. this.palabras = colores;
29. }else if (nivel == 4) {
30. this.palabras = prueba;
31. }
32. }
33. public char[][] getSopa() {
34. return sopa;
35. }
37. public void setSopa(char[][] sopa) {
38. this.sopa = sopa;
39. }
40. public void impPalabras(){
41. System.out.println(Arrays.toString(palabras));
42. }
43. public void cargar() {
44. Random r = new Random();
45. int xran;
46. int yran;
47. int ban;
49. for (int i = 0; i < palabras.length; i++) {
50. ban = 0;
51. String aux = palabras[i];
53. while (ban == 0) {
54. yran = r.nextInt(15) + 1;
55. xran = r.nextInt(15) + 1;
56. if (aux.length() < (sopa[0].length - xran)) {//derecha
57. if (cruza(yran, xran, aux, 1)) {
58. //System.out.println("aceptada1");
59. for (int h = 0; h < aux.length(); h++) {
60. sopa[yran][xran + h] = aux.charAt(h);
61. }
62. ban = 1;
63. }
64. } else if (aux.length() < (sopa.length - yran)) {//abajo
65. if (cruza(yran, xran, aux, 2)) {
66. //System.out.println("aceptada2");
67. for (int h = 0; h < aux.length(); h++) {
68. sopa[yran + h][xran] = aux.charAt(h);
69. }
70. ban = 1;
71. }
72. } else if ((sopa.length - yran) < aux.length()) {//izquierda
73. if (cruza(yran, xran, aux, 3)) {
74. //System.out.println("aceptada3");
75. for (int h = 0; h < aux.length(); h++) {
76. sopa[yran][xran - h] = aux.charAt(h);
77. }
78. ban = 1;
79. }
80. } else if ((sopa.length - yran) < aux.length()) {//arriba
81. if (cruza(yran, xran, aux, 4)) {
82. //System.out.println("aceptada4");
83. for (int h = 0; h < aux.length(); h++) {
84. sopa[yran - h][xran] = aux.charAt(h);
85. }
86. ban = 1;
87. }
88. }
89. }
90. }
91. }
93. public boolean cruza(int y, int x, String pal, int tipo) {
94. boolean retorno = true;
95. int aux = tipo;
96. if (tipo == 1) {
97. for (int i = 0; i < pal.length(); i++, x++) {
98. if (sopa[y][x] == '.' || (sopa[y][x] == pal.charAt(i))) {
99. continue;
100. } else {
101. retorno = false;
102. break;
103. }
104. }
105. } else if (tipo == 2) {
106. for (int i = 0; i < pal.length(); i++, y++) {
107. if (sopa[y][x] == '.' || (sopa[y][x] == pal.charAt(i))) {
108. continue;
109. } else {
110. retorno = false;
111. break;
112. }
113. }
114. } else if (tipo == 3) {
115. for (int i = 0; i < pal.length(); i++, x--) {
116. if (sopa[y][x] == '.' || (sopa[y][x] == pal.charAt(i))) {
117. continue;
118. } else {
119. retorno = false;
120. break;
121. }
122. }
123. } else if (tipo == 4) {
124. for (int i = 0; i < pal.length(); i++, y--) {
125. if (sopa[y][x] == '.' || (sopa[y][x] == pal.charAt(i))) {
126. continue;
127. } else {
128. retorno = false;
129. break;
130. }
131. }
132. }
133. return retorno;
134. }
136. public void mostrar() {
137. for (int i = 0; i < sopa.length; i++) {
138. if (i < 10) {
139. System.out.print("0" + i + "|");
140. } else {
141. System.out.print(i + "|");
142. }
143. for (int j = 0; j < sopa[0].length; j++) {
144. if (j != sopa[0].length) {
145. System.out.print(" " + sopa[i][j]);
146. }
147. }
148. if (i < 10) {
149. System.out.println("|" + "0" + i);
150. } else {
151. System.out.println("|" + i);
152. }
153. }
154. }
155. }

Finalmente tenemos nuestras clases Servidor\_O y Cliente\_O, en las cuales son muy similares a las clases que hemos estado usando, por parte del servidor obtenemos el nivel a usar para la sopa, mandamos nuestra sopa de letras al cliente y finalmente recibimos el puntaje si es necesario y lo escribimos en un archivo de texto. Por parte del cliente es mas simple ya que mandamos el nivel y al recibir la sopa de letras hacemos uso de la clase LogicaJuego para que el usuario pueda resolver la sopa creado y para finalizar se manda el puntaje obtenido.

**Clase Servidor\_O**

1. import java.net.\*;
2. import java.io.\*;
3. import java.util.Random;
5. public class Servidor\_O {
7. public static void main(String[] args) throws Exception {
9. try {
10. ServerSocket ss = new ServerSocket(3000);
11. System.out.println("Servidor iniciado");
12. for (;;) {
13. String puntaje;
14. String categoria="";
15. Socket cl = ss.accept();
16. ObjectOutputStream oos = new ObjectOutputStream(cl.getOutputStream());
17. ObjectInputStream ois = new ObjectInputStream(cl.getInputStream());
18. Objeto ob = (Objeto) ois.readObject();
19. int nivel = Integer.parseInt(ob.getPalabras());
20. if (nivel == 1) {
21. categoria = "animales";
22. } else if (nivel == 2) {
23. categoria = "frutas";
24. } else if (nivel == 3) {
25. categoria = "colores";
26. } else if (nivel == 4) {
27. categoria = "prueba";
28. }
29. System.out.println(categoria);
30. Sopa sopita = new Sopa(nivel);
31. sopita.cargar();
32. sopita.mostrar();
33. Sopa copia = sopita;
34. Objeto ob2 = new Objeto(sopita.getSopa());
35. oos.writeObject(ob2);
36. oos.flush();
37. Objeto ob3 = (Objeto) ois.readObject();
38. //recibiendo
39. puntaje = ob3.getPalabras();
40. ois.close();
41. oos.close();
42. cl.close();
43. //Archivo
45. BufferedWriter bw = null;
46. FileWriter fw = null;
47. File file = new File("Puntaciones.txt");
48. if (!file.exists()) {
49. file.createNewFile();
50. }
51. fw = new FileWriter(file.getAbsoluteFile(), true);
52. bw = new BufferedWriter(fw);
53. String data = "--------\n" + "Direccion: " + cl.getInetAddress() + " En el puero"+cl.getPort()+" categoria: " + categoria + " puntuacion: " + puntaje + "\n";
54. bw.write(data);
55. System.out.println("información agregada!");
56. bw.close();
57. fw.close();
59. }
60. } catch (Exception e) {
61. e.printStackTrace();
62. }
63. }
65. }

**Clase Cliente\_O**

1. import java.net.\*;
2. import java.io.\*;
3. import java.util.Scanner;
5. public class Cliente\_O {
7. public static void main(String[] args) {
8. int vidas = 4;
9. String[] animales = {"oso", "abeja", "ballena", "perro", "caballo", "camello", "gato", "cerdo", "elefante", "foca","leon","tigre","lagartija","mono","lobo"};
10. String[] frutas = {"uva", "lima", "limon", "cereza", "arandano", "platano", "manzana", "sandia", "fresa", "naranja"};
11. String[] colores = {"rojo", "naranja", "verde", "negro", "blanco ", "cafe ", "amaraillo", "rosa ", "gris ", "azul"};
12. String[] prueba={"memo","jose","sui"};
13. String[] palabras={};
14. try {
15. Scanner escaneo = new Scanner(System.in);
16. Socket cl = new Socket("localhost", 3000);
17. System.out.println("Conexion con servidor exitosa..");
18. ObjectOutputStream oos = new ObjectOutputStream(cl.getOutputStream());
19. ObjectInputStream ois = new ObjectInputStream(cl.getInputStream());
20. System.out.println("Seleccione la categoria \n 1)animales\n 2)frutas\n 3)colores\n 4)prueba");
21. int nivel = escaneo.nextInt();
22. Objeto ob = new Objeto(nivel + "");
23. oos.writeObject(ob);
24. oos.flush();
25. Objeto ob2 = (Objeto) ois.readObject();
26. char[][] sopa = ob2.getSopa();
27. double puntaje;
28. if (nivel == 1) {
29. palabras = animales;
30. } else if (nivel == 2) {
31. palabras = frutas;
32. } else if (nivel == 3) {
33. palabras = colores;
34. } else if (nivel == 4) {
35. palabras = prueba;
36. }
37. LogicaJuego juego = new LogicaJuego(sopa,palabras);
38. puntaje=juego.jugar();
39. Objeto ob3 = new Objeto(puntaje+"");
40. oos.writeObject(ob3);
41. oos.flush();
42. ois.close();
43. oos.close();
44. cl.close();
45. } catch (Exception e) {
46. e.printStackTrace();
47. }
48. }
50. }