

## TAREA SIMULACIÓN DE COLAS DE ESPERA – MODELAMIENTO Y SIMULACIÓN 6°

**Nombres:** Anthony Bladimir Vélez Villavicencio

**Objetivo:** Desarrollar un Simulador de colas de espera con ajuste de números aleatorios con MonteCarlo

### Procedimiento:

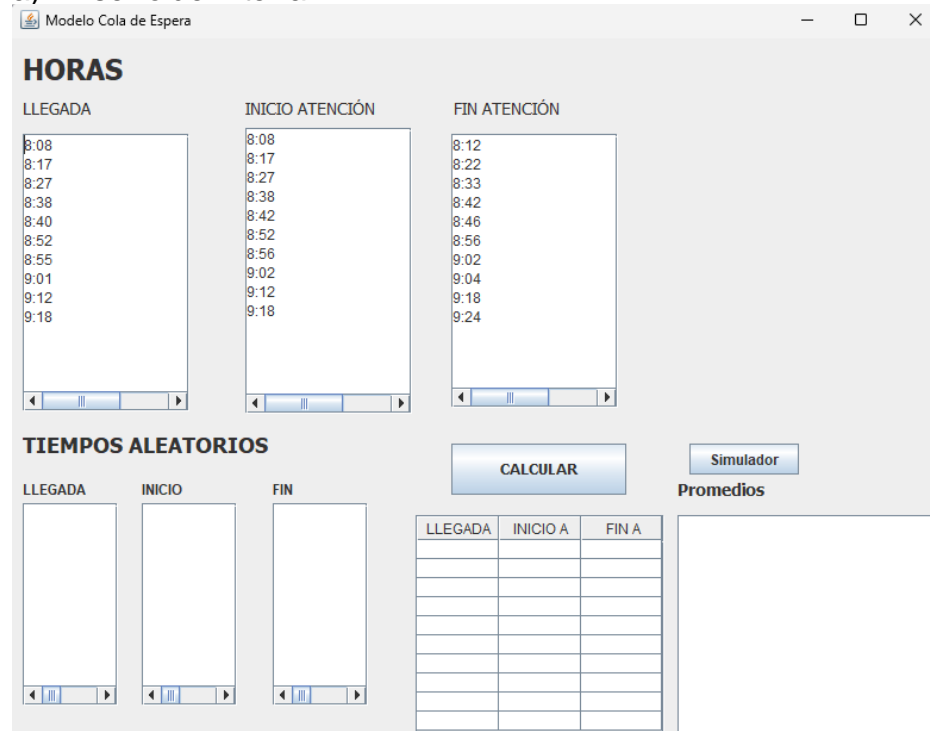
1. Implementar un formulario con el modelo de colas de espera, con el ingreso de los datos para el uso de MonteCarlo
2. Implementar un formulario con un simulador que genere los promedios de la ejecución de 10000 veces el modelo

Guardar el archivo en formato pdf y subir antes de la fecha límite de entrega

### Ejecución

1. Capturas del formulario con el modelo de colas de espera

#### a) Diseño de interfaz



#### b) Código para ejecutar el modelo

```
cModelo ejecutaModelo() {
    cModelo oM = new cModelo(entreda1, entrada2, entrada3);
    oM.Calcular();
    return oM;
}
```

### c) Código de MonteCarlo

```

Source History
15 public class cMC {
16     String vDat[];
17     List<cDat> LMC;
18 }
19 /**
20  * Constructor de la clase MonteCarlo
21  * @param pDat Enviar un vector de datos tipo String
22  */
23 public cMC(String[] pDat) {
24     this.vDat = pDat;
25     LMC=new ArrayList<>();
26     double FA=0;
27     for (String d : vDat) {
28         cDat nveD= new cDat();
29         nveD.dat=d;
30         nveD.rep=1;
31         nveD.fre=1.00/(double)vDat.length;
32         nveD.freAcu=(FA+=nveD.fre);
33         LMC.add(nveD);
34     }
35 }
36 /**
37  * Método que retorna un dato aleatorio ajustado con Montecarlo
38  * @return Un dato aleatorio tipo String
39  */
40 public String DatAle(){
41     double na=Math.random();
42     for (cDat filTab : LMC) {
43         if(na<=filTab.freAcu){
44             return filTab.dat;
45         }
46     }
47     return "error";
48 }

```

### d) Ejecución del modelo con datos (usar los datos que se encuentran anexos)

Modelo Cola de Espera

HORAS

LLEGADA

8:08  
8:17  
8:27  
8:38  
8:40  
8:52  
8:55  
9:01  
9:12  
9:18

INICIO ATENCIÓN

8:08  
8:17  
8:27  
8:38  
8:42  
8:52  
8:56  
9:02  
9:12  
9:18

FIN ATENCIÓN

8:12  
8:22  
8:33  
8:42  
8:46  
8:56  
9:02  
9:04  
9:18  
9:24

TIEMPOS ALEATORIOS

LLEGADA

12  
11  
11  
11  
11  
3  
11  
3  
2  
0

INICIO

0  
1  
0  
0  
0  
0  
0  
0  
1  
0

FIN

4  
5  
6  
6  
4  
6  
4  
5  
2  
4

CALCULAR

Simulador

Promedios

Promedio tiempo llegada: 8.4  
Promedio tiempo inicio: 0.2  
Promedio tiempo fin: 4.6  
Tiempo libre cajero: 30

LLEGADA	INICIO A	FIN A
08:12	08:12	08:16
08:23	08:24	08:29
08:34	08:34	08:40
08:45	08:45	08:51
08:56	08:56	09:00
08:59	09:00	09:06
09:10	09:10	09:14
09:13	09:14	09:19
09:15	09:19	09:21
09:24	09:24	09:28

2. Implementar un formulario con un simulador que genere los promedios de la ejecución de 10000 veces el modelo

a) Diseño de interface

Simulador

# HORAS

LLEGADA	INICIO ATENCIÓN	FIN ATENCIÓN
8:08	8:08	8:12
8:17	8:17	8:22
8:27	8:27	8:33
8:38	8:38	8:42
8:40	8:42	8:46
8:52	8:52	8:56
8:55	8:56	9:02
9:01	9:02	9:04
9:12	9:12	9:18
9:18	9:18	9:24

**CALCULAR**

jLabel5                      jLabel6                      jLabel7

Posición modelo 1...1000  **Mostrar resultado modelo**

jLabel9                      jLabel10                      jLabel11

b) Código de variables generales

```

    * @author Anthony Bladimir Vélez Villavicencio
    */
    public class cSimulador {
        List<cModelo> listaM = new ArrayList<cModelo>();
        String entreda1, entrada2, entrada3;
        public double pSL=0, pSI=0, pSF=0;
        public double rml=0, rml=0, rmlF=0;

        public cSimulador(String entreda1, String entrada2, String entrada3) {
            this.entreda1 = entreda1;
            this.entrada2 = entrada2;
            this.entrada3 = entrada3;
        }
    }

```

### c) Código del modelo

```

10  * @author Anthony Bladimir Vélez Villavicencio
11  */
12  public class cModelo {
13      String entreda1, entrada2, entrada3;
14      public double ProLle, ProIni, ProFin;
15
16  public cModelo(String entreda1, String entrada2, String entrada3) {
17      this.entreda1 = entreda1;
18      this.entrada2 = entrada2;
19      this.entrada3 = entrada3;
20  }
21
22  //Método 1
23
24  String[] SepararString(String Cadena) {
25      String[] vs = Cadena.split("\n");
26      return vs;
27  }
28
29  //Método 2
30  int SepararHora(String hora) {
31      String[] V = hora.split(":");
32      return Integer.parseInt(V[0]);
33  }
34
35  //Método 3
36  int SepararMinutos(String minutos) {
37      String[] V = minutos.split(":");
38      return Integer.parseInt(V[1]);
39  }
40
41  //Método 4
42  int TotalMinutos(int h, int m) {
43      int total = h * 60 + m;
44      return total;
45  }
46
47  //Método 5
48  int[] NumerosAleatorios(String VecAle[]) {
49      cMC omc = new cMC(VecAle);
50      int[] vs = new int[10];
51      for (int i = 0; i < vs.length; i++) {
52          vs[i] = Integer.parseInt(omc.DatAle());
53      }
54      return vs;
55  }
56
57  //Método 6
58  String[] VecMinaHor(int[] vn) {
59      String[] vH = new String[vn.length];
60      for (int i = 0; i < vn.length; i++) {
61          int h = vn[i] / 60;
62          int m = vn[i] % 60;
63          vH[i] = String.format("%02d:%02d", h, m);
64      }
65      return vH;
66  }
67  }

```

```

68
69 String MinaHor(int num) {
70     int h = num / 60;
71     int m = num % 60;
72     return String.format("%02d:%02d", h, m);
73 }
74
75 //Método 7
76 int[][] UnirMatriz(int[] v1, int[] v2, int[] v3) {
77     int[][] M = new int[3][v1.length];
78     for (int i = 0; i < 10; i++) {
79         M[0][i] = v1[i];
80         M[1][i] = v2[i];
81         M[2][i] = v3[i];
82     }
83     return M;
84 }
85
86 //Método 8
87 int[][] AjustarTiempos(int[][] M) {
88     for (int i = 0; i < 9; i++) {
89         if (M[1][i + 1] > M[2][i]) {
90             int dif = M[1][i + 1] - M[2][i];
91             M[1][i + 1] -= dif;
92             M[2][i + 1] += dif;
93         }
94     }
95     return M;
96 }
97
98 int[] VecHaVecMin(String[] vlle){
99     int[] minLleg = new int[vlle.length];
100     int i=0;
101     for (String hm : vlle) {
102         minLleg[i++] = TotalMinutos(SepararHora(hm), SepararMinutos(hm));
103     }
104     return minLleg;
105 }
106
107 //Calcula el promedio de un vector de enteros
108 double Promedio (int[] V){
109     double Pro=0;
110     for (int i : V) {
111         Pro+=i;
112     }
113     Pro/=10;
114     return Pro;
115 }
116
117 public void Calcular() {
118     // TODO add your handling code here:
119     String[] vlle = SepararString(entredal);
120     String[] vini = SepararString(entrada2);
121     String[] vfin = SepararString(entrada3);
122     //Transformar todo a minutos
123     int[] VMLle = VecHaVecMin(vlle);
124     int[] VMIni = VecHaVecMin(vini);
125     int[] VMFin = VecHaVecMin(vfin);

```

```

125
126 //Vectores string para numeros aleatorios
127 String[] TL = new String[VMLle.length - 1];
128 String[] TI = new String[VMini.length];
129 String[] TF = new String[VFin.length];
130 //Llenar los vectores para números aleatorios
131 for (int i = 0; i < TL.length; i++) {
132     TL[i] = (VMLle[i + 1] - VMLle[i]) + ""; //Tiempos de llegada
133 }
134 for (int i = 0; i < TI.length; i++) {
135     TI[i] = (VMini[i] - VMLle[i] + ""); //Tiempos de inicio
136 }
137 for (int i = 0; i < TF.length; i++) {
138     TF[i] = (VFin[i] - VMini[i] + ""); //Tiempos de fin
139 }
140 //Vectores de números aleatorios
141 int VALlega[] = NumerosAleatorios(TL);
142 int VAINi[] = NumerosAleatorios(TI);
143 int VAFin[] = NumerosAleatorios(TF);
144 //Mostrar los números aleatorios en las areas de texto
145 //     JTextArea4.setText("");
146 //     JTextArea5.setText("");
147 //     JTextArea6.setText("");
148 //     for (int i = 0; i < 10; i++) {
149 //         JTextArea4.setText(JTextArea4.getText() + VALlega[i] + "\n");
150 //         JTextArea5.setText(JTextArea5.getText() + VAFin[i] + "\n");
151 //         JTextArea6.setText(JTextArea6.getText() + VAINi[i] + "\n");
152 //     }
153 int[][] M = UnirMatriz(VALlega, VAINi, VAFin);
154 int TotTiempoLle = 8 * 60;
155 //Ajusta a minutos con horas
156 for (int j = 0; j < 10; j++) {
157     M[0][j] = TotTiempoLle += M[0][j];
158     M[1][j] += M[0][j];
159     if (j > 0 && M[1][j] < M[2][j - 1]) {
160         M[1][j] = M[2][j - 1];
161     }
162     M[2][j] += M[1][j];
163 }
164 //Visualiza la matriz en la tabla en formato h:m
165 //     for (int i = 0; i < 3; i++) {
166 //         for (int j = 0; j < 10; j++) {
167 //             jTable1.setValueAt(MinaHor(M[i][j]), j, i);
168 //         }
169 //     }
170 // }
171 //Calcular promedios
172 ProLle = Promedio(VALlega);
173 ProIni = Promedio(VAINi);
174 ProFin = Promedio(VAFin);
175
176 int TieLib = 0;
177 for (int j = 1; j < 10; j++) {
178     TieLib += M[1][j] - M[2][j - 1];
179 }
180

```

d) Código del simulador (promedio)

```

31
32 public void simular(){
33     for (int i = 0; i < 10000; i++) {
34         cModelo OTM=ejecutaModelo();
35         listaM.add(OTM);
36         pSL+= OTM.ProLle;
37         pSI+= OTM.ProIni;
38         pSF+= OTM.ProFin;
39     }
40
41     pSL /= 10000;
42     pSI /= 10000;
43     pSF /= 10000;
44

```

e) Ejecución del modelo con datos

Simulador

# HORAS

LLEGADA

8:08  
8:17  
8:27  
8:38  
8:40  
8:52  
8:55  
9:01  
9:12  
9:18

INICIO ATENCIÓN

8:08  
8:17  
8:27  
8:38  
8:42  
8:52  
8:56  
9:02  
9:12  
9:18

FIN ATENCIÓN

8:12  
8:22  
8:33  
8:42  
8:46  
8:56  
9:02  
9:04  
9:18  
9:24

CALCULAR

Promedio tiempo de llegada 7,77    Promedio tiempo de inicio 0,40    Promedio tiempo de fin 4,69

Posición modelo 1...1000    

Mostrar resultado modelo

Tiempo llegada 8,80    Tiempo inicio 0,70    Tiempo fin 5,20

## Rúbrica

Ejecución	Completo	Mayor a 50%	Inferior a 50%	Puntaje máximo
1.a	1	0.5	0.25	1
1.b	1	0.5	0.25	1
1.c	1	0.5	0.25	1
1.d	1	0.5	0.25	1
2.a	1	0.5	0.25	1
2.b	1	0.5	0.25	1
2.c	1	0.5	0.25	1
2.d	2	1.0	0.50	2
2.e	1	0.5	0.25	1
Total				10

## Anexo

Tablas de tiempos por grupos

Grupo A				Grupo B				Grupo C				Grupo D			
1	8:05	8:05	8:09	1	8:06	8:06	8:13	1	8:09	8:09	8:13	1	8:10	8:10	8:16
2	8:13	8:13	8:21	2	8:15	8:15	8:21	2	8:10	8:13	8:17	2	8:22	8:22	8:27
3	8:18	8:21	8:25	3	8:20	8:21	8:28	3	8:17	8:17	8:23	3	8:27	8:27	8:34
4	8:25	8:25	8:33	4	8:29	8:29	8:38	4	8:22	8:23	8:26	4	8:36	8:36	8:38
5	8:33	8:33	8:42	5	8:36	8:38	8:43	5	8:30	8:30	8:32	5	8:44	8:44	8:49
6	8:37	8:42	8:44	6	8:43	8:43	8:47	6	8:34	8:34	8:37	6	8:46	8:49	8:51
7	8:46	8:46	8:51	7	8:49	8:49	8:54	7	8:43	8:43	8:50	7	8:52	8:52	9:01
8	8:50	8:51	8:54	8	8:52	8:54	8:58	8	8:45	8:50	8:56	8	8:59	9:01	9:10
9	8:59	8:59	9:05	9	8:57	8:58	9:02	9	8:46	8:56	8:58	9	9:09	9:10	9:18
10	9:08	9:08	9:16	10	9:00	9:02	9:07	10	8:48	8:58	9:07	10	9:18	9:18	9:20

Grupo E				Grupo F				Grupo G				Grupo H			
1	8:05	8:05	8:13	1	8:08	8:08	8:12	1	8:05	8:05	8:08	1	8:07	8:07	8:17
2	8:10	8:13	8:15	2	8:17	8:17	8:22	2	8:12	8:12	8:15	2	8:13	8:17	8:25
3	8:17	8:17	8:21	3	8:27	8:27	8:33	3	8:21	8:21	8:24	3	8:17	8:25	8:28
4	8:25	8:25	8:33	4	8:38	8:38	8:42	4	8:29	8:29	8:38	4	8:28	8:28	8:37
5	8:37	8:37	8:40	5	8:40	8:42	8:46	5	8:31	8:38	8:42	5	8:38	8:38	8:43
6	8:49	8:49	8:52	6	8:52	8:52	8:56	6	8:42	8:42	8:44	6	8:50	8:50	8:57
7	8:52	8:52	9:00	7	8:55	8:56	9:02	7	8:49	8:49	8:52	7	9:00	9:00	9:04
8	8:54	9:00	9:04	8	9:01	9:02	9:04	8	8:55	8:55	9:00	8	9:01	9:04	9:09
9	8:55	9:04	9:08	9	9:12	9:12	9:18	9	8:56	9:00	9:03	9	9:02	9:09	9:11
10	8:58	9:08	9:12	10	9:18	9:18	9:24	10	9:02	9:03	9:08	10	9:10	9:11	9:17



## Asignación de datos por estudiantes

#	Estudiante	Grupo
1	ALAVA RODRIGUEZ <b>JUAN</b> PABLO	H
2	ALAY CRUZ EDITH <b>MARIBEL</b>	A
3	BASURTO MUÑOZ <b>FERNANDA</b> ELIZABETH	D
4	CARRANZA MOREIRA <b>TIFANI</b> BELEN	G
5	CAZA ROMERO <b>DIANA</b> PAOLA	G
6	CEDEÑO BRAVO <b>WILLIAN</b> JAVIER	E
7	CEDEÑO LOPEZ <b>ROMMEL</b> ANTONIO	E
8	CHICA GOMEZ <b>RUTH</b> ESTEFANIA	F
9	DAZA SANTOS <b>SILVANA</b> YANIRA	F
10	FUERES MADERA <b>ARIEL</b> ALEXANDER	D
11	LOOR MERA <b>NAYELI</b> MARIA	A
12	MEDRANDA ESTEVEZ ERICK <b>ALEJANDRO</b>	G
13	MOREIRA HUERTA <b>ANGIE</b> ELIZABETH	B
14	MUÑOZ CEDEÑO <b>WELLINGTON</b> BIENVENIDO	F
15	VELAZQUEZ VERGARA <b>EVELYN</b> JAZMIN	C
16	VELEZ SOLORZANO <b>DANIEL</b> ALEXANDER	H
17	VELEZ VILLAVICENCIO <b>ANTHONY</b> BLADIMIR	F
18	VERA LOOR <b>MILLER</b> EFRAIN	G
19	VIVAS MACIAS <b>ADRIAN</b> ALEXANDER	C
20	ZAMBRANO BARRE <b>JENIFFER</b> MARIA	D
21	ZAMBRANO VERA <b>NEICER</b> DANIEL	B