

```

1
2
3
4 # PUNTO A: load data
5 data = load('dataset/FlujoVehicular2019.dat');
6
7 # mapeo nombres - columnas del dataset
8
9 mes = 1;
10 diames = 2;
11 hora = 3;
12 diasemana = 4;
13 estacion = 5;
14 sentido = 6;
15 tipovehiculo = 7;
16 formapago = 8;
17 cantidadpasos = 9;
18
19
20 # PUNTO B: Balances
21
22 # balance de ingresos y egresos totales segun horario las 24hs del dia
23
24 # vector contador de ingresos x horario
25 vector_ingresos_hora = zeros (24,1);
26
27 # vector contador de egresos x horario
28 vector_egresos_hora = zeros (24,1);
29
30 cantidad_de_filas = rows(data);
31 disp(cantidad_de_filas);
32
33 # Opero sobre el data set para realizar el conteo x horarios
34 for fila = 1:cantidad_de_filas
35     # guardo la hora de la fila analizada
36     hora_actual = data(fila, hora) +1;
37     # analizo ingreso / egreso y sumo cantidades al contador correspondiente
38     if(data(fila, sentido) == 1)
39         vector_ingresos_hora(hora_actual) = vector_ingresos_hora(hora_actual) +
40         data(fila, cantidadpasos);
41     else
42         vector_egresos_hora(hora_actual) = vector_egresos_hora(hora_actual) + data(fila,
43         cantidadpasos);
44     endif
45 endfor
46
47 disp('vector de ingresos:');
48 disp(vector_ingresos_hora);
49 disp('vector de egresos:');
50 disp(vector_egresos_hora);
51
52 # Graficamos los ingresos por hora
53 plot(vector_ingresos_hora, 'color', 'g')
54 title ("CANTIDAD DE INGRESOS TOTALES POR HORA");
55 set(gca,'fontsize',12);
56 set(gca,'XTick',1:24);
57 xlabel ("HORA");
58 ylabel ("Cantidad de pasos");
59 print -djpg graficos/puntoB/Grafico_Ingresos_Totales_x_Hora.jpg
60
61 #Graficamos los egresos por hora
62 plot(vector_egresos_hora, 'color', 'r')
63 title ("CANTIDAD DE EGRESOS TOTALES POR HORA");
64 set(gca,'fontsize',12);

```

```

64 set(gca,'XTick',1:24);
65 xlabel ("HORA");
66 ylabel ("Cantidad de pasos");
67 print -djpg graficos/puntoB/Grafico_Egresos_Totales_x_Hora.jpg
68
69
70 #Punto C:
71 #Pasos totales por estacion.
72
73 vector_pasostot_por_estacion = zeros(8,1);
74
75 #lee por cada fila la estacion que se analiza y suma los pasos a un vector.
76 for fila = 1:cantidad_de_filas
77     estacion_analizada = data(fila, estacion);
78     vector_pasostot_por_estacion(estacion_analizada) =
79     vector_pasostot_por_estacion(estacion_analizada) + data(fila, cantidadpasos);
80 endfor
81
82 #analizo cual es el mayor valor y obtengo su posicion en el vector que coincide con
83 el numero de estacion.
84 maximo = -1;
85 repetidos = zeros (1,1);
86 for fila_de_vector = 1:8
87     if vector_pasostot_por_estacion(fila_de_vector, 1 ) > maximo
88         maximo = vector_pasostot_por_estacion(fila_de_vector, 1 );
89         estacion_max = fila_de_vector;
90     elseif vector_pasostot_por_estacion(fila_de_vector, 1 ) == maximo
91         estacion_max_rep = fila_de_vector;
92     endif
93 endfor
94
95 #analizo el balance de la estacion con maxima cantida de pasos.
96 vector_est_max_ingresos = zeros (24,1);
97 vector_est_max_egresos = zeros (24,1);
98
99 for fila = 1:cantidad_de_filas
100     # guardo la hora de la fila analizada
101     if (data(fila, estacion) == estacion_max)
102         hora_actual_est_max = data(fila, hora) +1;
103     # analizo ingreso / egreso y sumo cantidades al contador correspondiente
104     if(data(fila, sentido) == 1)
105         vector_est_max_ingresos(hora_actual_est_max) =
106         vector_est_max_ingresos(hora_actual_est_max) + data(fila, cantidadpasos);
107     else
108         vector_est_max_egresos(hora_actual_est_max) =
109         vector_est_max_egresos(hora_actual_est_max) + data(fila, cantidadpasos);
110     endif
111 endfor
112
113 # Graficamos los ingresos y egresos para la estacion con maxima cantidad de pasos
114 disp("La estacion con mayor cantidad de pasos totales es:");
115 disp(estacion_max);
116
117 matriz_balance = [vector_est_max_ingresos,vector_est_max_egresos];
118 bar(matriz_balance)
119 title ("PASOS POR HORA EN LA ESTACION CON MAXIMA CANTIDAD DE PASOS ");
120 legend("Ingresos","Egresos",'location','northwest')
121 set(gca,'fontsize',10);
122 set(gca,'XTick',1:24);
123 xticklabels(1:24)
124 xlabel ("HORAS");
125 ylabel ("Cantidad de pasos");
126 print -djpg graficos/puntoC/Pasos_x_hora_estacion_maxima.jpg

```

```

1
2 data = load('dataset/FlujoVehicular2019.dat');
3
4 # mapeo nombres - columnas del dataset
5
6 mes = 1;
7 diames = 2;
8 hora = 3;
9 diasemana = 4;
10 estacion = 5;
11 sentido = 6;
12 tipovehiculo = 7;
13 formapago = 8;
14 cantidadpasos = 9;
15 cantidad_de_filas = rows(data);
16
17 # PUNTO D
18 # IDENTIFICAR LAS DOS FRANJAS DE TRES HORAS CADA UNA, SIN SOLAPARSE QUE PRESENTAN
    MAYOR MOVILIDAD. CONTABILIZARLAS PARA TODAS LAS UNIDADES DE PEAJES TANTO INGRESOS
    COMO EGRESOS. GRAFICAR.
19
20 # defino un vector donde voy a ir guardando las franjas horarias.
21 vector_franjas_horarias = [-1 -1 -1 -1];
22
23 # calculo las franjas horarias posibles. para cada una de las horas, le sumo 2 para
    obtener las dos siguientes y guardo ese trio de horarios en el vector
24
25 for hora_numero = 0:21
26     vector_franjas_horarias = [vector_franjas_horarias; hora_numero hora_numero + 1
        hora_numero + 2 0];
27 endfor
28 disp('Franjas horarias:')
29 disp(vector_franjas_horarias(2:23,1:3));
30 # Recuento cantidad de ingresos + egresos para cada franja horaria, los voy a contar
    en la cuarta columna de la matriz que cree arriba vector_franjas_horarias
31
32 for fila_data = 1:cantidad_de_filas
33     # para cada una de las franjas horarias que tengo analizo mi data.
34     for franja = 1:rows(vector_franjas_horarias)
35         if(franja != 1)
36             if (data(fila_data, hora) == vector_franjas_horarias(franja, 1) ||
                data(fila_data, hora) == vector_franjas_horarias(franja, 2) ||
                data(fila_data, hora) == vector_franjas_horarias(franja, 3))
37                 vector_franjas_horarias(franja, 4) = vector_franjas_horarias(franja, 4) +
                    data(fila_data, cantidadpasos);
38             endif
39         endif
40     endfor
41 endfor
42
43 disp("cantidad de pasos por cada franja horaria de 3 horas:");
44 # obtengo ordenadas las franjas horarias y sus ingresos para saber cuales son las
    dos con mayor cantidad de pasos.
45 disp(sortrows(vector_franjas_horarias(2:23,1:4 ),4));
46
47 # Grafico la cantidad de pasos por cada una de las franjas horarias.
48 bar(vector_franjas_horarias(2:23,4));
49 title ('CANTIDAD DE PASOS POR FRANJA HORARIA');
50 set(gca, 'fontsize', 12);
51 set(gca, 'XTick', 1:23);
52 ylim([0 3e+07])
53 xlabel ("Franjas horarias");
54 ylabel ("Cantidad de pasos");
55 #print -djpg graficos/puntoD/cantidad_de_pasos_por_franja_horaria.jpg

```

```

1  # PUNTO A: load data
2  data = load('dataset/FlujoVehicular2019.dat');
3
4  # mapeo nombres - columnas del dataset
5
6  mes = 1;
7  diames = 2;
8  hora = 3;
9  diasemana = 4;
10  estacion = 5;
11  sentido = 6;
12  tipovehiculo = 7;
13  formapago = 8;
14  cantidadpasos = 9;
15
16  cantidad_de_filas = rows(data);
17  #punto E
18
19  #Se crean las matrices vacias que tendran por cada fila numero de estacion y
20
21  matriz_livianos_por_estacion = zeros (8,2);
22  matriz_pesados_por_estacion = zeros (8,2);
23
24  #lee por cada fila del archivo la estacion que se analiza, se fija si es liviano y
  suma los pasos a en la 2da columna los pasos.
25  #mientras que en la primera fija la posicion de la estacion.
26  for fila = 1:cantidad_de_filas
27      estacion_analizada_E = data(fila, estacion);
28      matriz_livianos_por_estacion(estacion_analizada_E,1) = estacion_analizada_E;
29      matriz_pesados_por_estacion (estacion_analizada_E,1) = estacion_analizada_E;
30      if (data(fila,tipovehiculo) == 1)
31          matriz_livianos_por_estacion (estacion_analizada_E,2) =
            matriz_livianos_por_estacion(estacion_analizada_E,2) + data(fila, cantidadpasos);
32      else
33          matriz_pesados_por_estacion (estacion_analizada_E,2) =
            matriz_pesados_por_estacion(estacion_analizada_E,2) + data(fila, cantidadpasos);
34      endif
35  endfor
36  #ordena las matrices respecto de la 2da columna, de esta manera nos queda en la
  segunda columna en forma ascendente
37  #la estacion con menor cantidad de pasos, y su respectiva cantidad de pasos.
38  matriz_livianos_por_estacion_ordenada = sortrows(matriz_livianos_por_estacion, -2);
39  matriz_pesados_por_estacion_ordenada = sortrows(matriz_pesados_por_estacion, -2);
40
41  disp('  Estacion  Pasos de Vehiculos Livianos')
42  disp(matriz_livianos_por_estacion_ordenada);
43  disp(' ');
44  disp('  Estacion  Pasos de Vehiculos Pesados')
45  disp(matriz_pesados_por_estacion_ordenada);
46  disp(' ');
47
48  bar(matriz_livianos_por_estacion_ordenada(:,1),
  matriz_livianos_por_estacion_ordenada(:,2), 0.4)
49  title ("CANTIDAD DE PASOS DE VEHICULOS LIVIANOS POR ESTACION");
50  set(gca,'fontsize',12);
51  set(gca,'XTick',1:8);
52  xlabel ("Estaciones");
53  ylabel ("Cantidad de pasos");
54  print -djpg graficos/puntoE/grafico_pasos_livianos_x_estacion.jpg
55
56  bar(matriz_pesados_por_estacion(:,2),0.4)
57  title ("CANTIDAD DE PASOS DE VEHICULOS PESADOS POR ESTACION");
58  set(gca,'fontsize',12);
59  set(gca,'XTick',1:8);
60  xlabel ("Estaciones");

```

```
61  ylabel ("Cantidad de pasos");  
62  print -djpg graficos/puntoE/grafico_pasos_pesados_x_estacion.jpg  
63
```

```

1  # TP NUMERO 1 -- PEAJES -- PUNTO F
2  # PUNTO A: load data
3  data = load('dataset/FlujoVehicular2019.dat');
4
5  function [vector_ingresos_prom_x_dia,vector_egresos_prom_x_dia] =
pasos_prom_estacion (data,estacion_obs)
6      # mapeo nombres - columnas del dataset
7      mes = 1;
8      diames = 2;
9      hora = 3;
10     diasemana = 4;
11     estacion = 5;
12     sentido = 6;
13     tipovehiculo = 7;
14     formapago = 8;
15     cantidadpasos = 9;
16
17     cantidad_de_filas = rows(data);
18     #creo los vectores totales vacios
19     vector_ingresos_totales_x_dia = zeros (7,1);
20     vector_egresos_totales_x_dia = zeros (7,1);
21     for fila = 1:cantidad_de_filas
22         dia_sem = data(fila, diasemana);
23         if ((data(fila,estacion)== estacion_obs) && (data(fila,sentido) == 1))
24             vector_ingresos_totales_x_dia (dia_sem) = vector_ingresos_totales_x_dia
(dia_sem) + data(fila,cantidadpasos);
25         elseif ((data(fila,estacion)== estacion_obs) && (data(fila,sentido) == 2))
26             vector_egresos_totales_x_dia(dia_sem) = vector_egresos_totales_x_dia (dia_sem)
+ data(fila,cantidadpasos);
27         endif
28     endfor
29     #Chequeo que no haya ceros ingresos o egresos en algun peaje y divido por 52 cada
dato ya que es la cantidad
30     #promedio de dias que hay por dia por año, asi creo los vectores promedios
31     for j = 1:7
32         vector_ingresos_prom_x_dia(j) = vector_ingresos_totales_x_dia(j) ./ 52;
33         vector_egresos_prom_x_dia(j) = vector_egresos_totales_x_dia(j) ./ 52;
34     endfor
35 endfunction
36 #Llamados a la funcion, resultados y grafico cada estacion
37 #estacion 1
38 [vector_ingresos_prom_x_dia_estacion_1,vector_egresos_prom_x_dia_estacion_1] =
pasos_prom_estacion (data,1)
39 matriz_1 =
[vector_ingresos_prom_x_dia_estacion_1;vector_egresos_prom_x_dia_estacion_1];
40 bar(matriz_1')
41 title ("CANTIDAD DE PASOS PROMEDIO POR DIA EN LA ESTACION 1");
42 legend("Ingresos","Egresos")
43 set(gca,'fontsize',12);
44 set(gca,'XTick',1:7);
45 xticklabels({'Domingo','Lunes','Martes','Miercoles','Jueves','Viernes','Sabado'})
46 xlabel ("Días");
47 ylabel ("Cantidad de pasos promedio");
48 print -djpg graficos/puntoF/Grafico_pasos_prom_x_dia_01.jpg
49
50 #Resultados y grafico para la estacion 2
51 [vector_ingresos_prom_x_dia_estacion_2,vector_egresos_prom_x_dia_estacion_2] =
pasos_prom_estacion (data,2)
52 matriz_2 =
[vector_ingresos_prom_x_dia_estacion_2;vector_egresos_prom_x_dia_estacion_2];
53 bar(matriz_2')
54 title ("CANTIDAD DE PASOS PROMEDIO POR DIA EN LA ESTACION 2");
55 legend("Ingresos","Egresos")
56 set(gca,'fontsize',12);
57 set(gca,'XTick',1:7);

```

```

58 xticklabels({'Domingo','Lunes','Martes','Miercoles','Jueves','Viernes','Sabado'})
59 xlabel ("Días");
60 ylabel ("Cantidad de pasos promedio");
61 print -djpg graficos/puntoF/Grafico_pasos_prom_x_dia_02.jpg
62
63 #Resultados y grafico para la estacion 3
64 [vector_ingresos_prom_x_dia_estacion_3,vector_egresos_prom_x_dia_estacion_3] =
pasos_prom_estacion (data,3)
65 matriz_3 =
[vector_ingresos_prom_x_dia_estacion_3;vector_egresos_prom_x_dia_estacion_3];
66 bar(matriz_3')
67 title ("CANTIDAD DE PASOS PROMEDIO POR DIA EN LA ESTACION 3");
68 legend("Ingresos","Egresos")
69 set(gca,'fontsize',12);
70 set(gca,'XTick',1:7);
71 xticklabels({'Domingo','Lunes','Martes','Miercoles','Jueves','Viernes','Sabado'})
72 xlabel ("Días");
73 ylabel ("Cantidad de pasos promedio");
74 print -djpg graficos/puntoF/Grafico_pasos_prom_x_dia_03.jpg
75
76 #Resultados y grafico para la estacion 4
77 [vector_ingresos_prom_x_dia_estacion_4,vector_egresos_prom_x_dia_estacion_4] =
pasos_prom_estacion (data,4)
78 matriz_4 =
[vector_ingresos_prom_x_dia_estacion_4;vector_egresos_prom_x_dia_estacion_4];
79 bar(matriz_4')
80 title ("CANTIDAD DE PASOS PROMEDIO POR DIA EN LA ESTACION 4");
81 legend("Ingresos","Egresos")
82 set(gca,'fontsize',12);
83 set(gca,'XTick',1:7);
84 xticklabels({'Domingo','Lunes','Martes','Miercoles','Jueves','Viernes','Sabado'})
85 xlabel ("Días");
86 ylabel ("Cantidad de pasos promedio");
87 print -djpg graficos/puntoF/Grafico_pasos_prom_x_dia_04.jpg
88
89 #Resultados y grafico para la estacion 5
90 [vector_ingresos_prom_x_dia_estacion_5,vector_egresos_prom_x_dia_estacion_5] =
pasos_prom_estacion (data,5)
91 matriz_5 =
[vector_ingresos_prom_x_dia_estacion_5;vector_egresos_prom_x_dia_estacion_5];
92 bar(matriz_5')
93 title ("CANTIDAD DE PASOS PROMEDIO POR DIA EN LA ESTACION 5");
94 legend("Ingresos","Egresos")
95 set(gca,'fontsize',12);
96 set(gca,'XTick',1:7);
97 xticklabels({'Domingo','Lunes','Martes','Miercoles','Jueves','Viernes','Sabado'})
98 xlabel ("Días");
99 ylabel ("Cantidad de pasos promedio");
100 print -djpg graficos/puntoF/Grafico_pasos_prom_x_dia_05.jpg
101
102 #Resultados y grafico para la estacion 6
103 [vector_ingresos_prom_x_dia_estacion_6,vector_egresos_prom_x_dia_estacion_6] =
pasos_prom_estacion (data,6)
104 matriz_6 =
[vector_ingresos_prom_x_dia_estacion_6;vector_egresos_prom_x_dia_estacion_6];
105 bar(matriz_6')
106 title ("CANTIDAD DE PASOS PROMEDIO POR DIA EN LA ESTACION 6");
107 legend("Ingresos","Egresos")
108 set(gca,'fontsize',12);
109 set(gca,'XTick',1:7);
110 xticklabels({'Domingo','Lunes','Martes','Miercoles','Jueves','Viernes','Sabado'})
111 xlabel ("Días");
112 ylabel ("Cantidad de pasos promedio");
113 print -djpg graficos/puntoF/Grafico_pasos_prom_x_dia_06.jpg
114

```

```

115 #Resultados y grafico para la estacion 7
116 [vector_ingresos_prom_x_dia_estacion_7,vector_egresos_prom_x_dia_estacion_7] =
pasos_prom_estacion (data,7)
117 matriz_7 =
[vector_ingresos_prom_x_dia_estacion_7;vector_egresos_prom_x_dia_estacion_7];
118 bar(matriz_7')
119 title ("CANTIDAD DE PASOS PROMEDIO POR DIA EN LA ESTACION 7");
120 legend("Ingresos","Egresos")
121 set(gca,'fontsize',12);
122 set(gca,'XTick',1:7);
123 xticklabels({'Domingo','Lunes','Martes','Miercoles','Jueves','Viernes','Sabado'})
124 xlabel ("Días");
125 ylabel ("Cantidad de pasos promedio");
126 print -djpg graficos/puntoF/Grafico_pasos_prom_x_dia_07.jpg
127
128 #Resultados y grafico para la estacion 8
129 [vector_ingresos_prom_x_dia_estacion_8,vector_egresos_prom_x_dia_estacion_8] =
pasos_prom_estacion (data,8)
130 matriz_8 =
[vector_ingresos_prom_x_dia_estacion_8;vector_egresos_prom_x_dia_estacion_8];
131 bar(matriz_8')
132 title ("CANTIDAD DE PASOS PROMEDIO POR DIA EN LA ESTACION 8");
133 legend("Ingresos","Egresos")
134 set(gca,'fontsize',12);
135 set(gca,'XTick',1:7);
136 xticklabels({'Domingo','Lunes','Martes','Miercoles','Jueves','Viernes','Sabado'})
137 xlabel ("Días");
138 ylabel ("Cantidad de pasos promedio");
139 print -djpg graficos/puntoF/Grafico_pasos_prom_x_dia_08.jpg
140

```



```

1  # TP NUMERO 1 -- PEAJES -- Punto G
2  #load data
3  data = load('dataset/FlujoVehicular2019.dat');
4
5  function [vector_ingresos_totales_x_mes,vector_egresos_totales_x_mes] =
pasos_prom_estacion (data,estacion_obs)
6      # mapeo nombres - columnas del dataset
7      mes = 1;
8      diames = 2;
9      hora = 3;
10     diasemana = 4;
11     estacion = 5;
12     sentido = 6;
13     tipovehiculo = 7;
14     formapago = 8;
15     cantidadpasos = 9;
16
17     cantidad_de_filas = rows(data);
18     #creo los vectores totales vacios
19     vector_ingresos_totales_x_mes = zeros (12,1);
20     vector_egresos_totales_x_mes = zeros (12,1);
21     for fila = 1:cantidad_de_filas
22         mes_obs = data(fila, mes);
23         if ((data(fila,estacion)== estacion_obs) && (data(fila,sentido) == 1))
24             vector_ingresos_totales_x_mes (mes_obs) = vector_ingresos_totales_x_mes
(mes_obs) + data(fila,cantidadpasos);
25         elseif ((data(fila,estacion)== estacion_obs) && (data(fila,sentido) == 2))
26             vector_egresos_totales_x_mes(mes_obs) = vector_egresos_totales_x_mes (mes_obs)
+ data(fila,cantidadpasos);
27         endif
28     endfor
29 endfunction
30 #Llamados a la funcion, resultados y grafico cada estacion
31 #estacion 1
32 [vector_ingresos_prom_x_mes_estacion_1,vector_egresos_prom_x_mes_estacion_1] =
pasos_prom_estacion (data,1)
33 matriz_1 = [vector_ingresos_prom_x_mes_estacion_1,vector_egresos_prom_x_mes_estacion_1]
34 bar(matriz_1)
35 title ("CANTIDAD DE PASOS PROMEDIO POR MES EN LA ESTACION 1");
36 legend("Ingresos","Egresos",'location','northwest')
37 set(gca,'fontsize',10);
38 set(gca,'XTick',1:12);
39 xticklabels(1:12)
40 xlabel ("MESES");
41 ylabel ("Cantidad de pasos promedio");
42 print -djpg graficos/puntoG/Grafico_pasos_prom_x_mes_01.jpg
43
44 #estacion 2
45 [vector_ingresos_prom_x_mes_estacion_2,vector_egresos_prom_x_mes_estacion_2] =
pasos_prom_estacion (data,2)
46 matriz_2 = [vector_ingresos_prom_x_mes_estacion_2,vector_egresos_prom_x_mes_estacion_2]
47 bar(matriz_2)
48 title ("CANTIDAD DE PASOS PROMEDIO POR MES EN LA ESTACION 2");
49 legend("Ingresos","Egresos",'location','northwest')
50 set(gca,'fontsize',10);
51 set(gca,'XTick',1:12);
52 xticklabels(1:12)
53 xlabel ("MESES");
54 ylabel ("Cantidad de pasos promedio");
55 print -djpg graficos/puntoG/Grafico_pasos_prom_x_mes_02.jpg
56
57 #estacion 3
58 [vector_ingresos_prom_x_mes_estacion_3,vector_egresos_prom_x_mes_estacion_3] =
pasos_prom_estacion (data,3)
59 matriz_3 = [vector_ingresos_prom_x_mes_estacion_3,vector_egresos_prom_x_mes_estacion_3]

```

```

60 bar(matriz_3)
61 title ("CANTIDAD DE PASOS PROMEDIO POR MES EN LA ESTACION 3");
62 legend("Ingresos","Egresos",'location','northwest')
63 set(gca,'fontsize',10);
64 set(gca,'XTick',1:12);
65 xticklabels(1:12)
66 xlabel ("MESES");
67 ylabel ("Cantidad de pasos promedio");
68 print -djpg graficos/puntoG/Grafico_pasos_prom_x_mes_03.jpg
69
70 #estacion 4
71 [vector_ingresos_prom_x_mes_estacion_4,vector_egresos_prom_x_mes_estacion_4] =
pasos_prom_estacion (data,4)
72 matriz_4 = [vector_ingresos_prom_x_mes_estacion_4,vector_egresos_prom_x_mes_estacion_4]
73 bar(matriz_4)
74 title ("CANTIDAD DE PASOS PROMEDIO POR MES EN LA ESTACION 4");
75 legend("Ingresos","Egresos",'location','northwest')
76 set(gca,'fontsize',10);
77 set(gca,'XTick',1:12);
78 xticklabels(1:12)
79 xlabel ("MESES");
80 ylabel ("Cantidad de pasos promedio");
81 print -djpg graficos/puntoG/Grafico_pasos_prom_x_mes_04.jpg
82
83 #estacion 5
84 [vector_ingresos_prom_x_mes_estacion_5,vector_egresos_prom_x_mes_estacion_5] =
pasos_prom_estacion (data,5)
85 matriz_5 = [vector_ingresos_prom_x_mes_estacion_2,vector_egresos_prom_x_mes_estacion_5]
86 bar(matriz_5)
87 title ("CANTIDAD DE PASOS PROMEDIO POR MES EN LA ESTACION 5");
88 legend("Ingresos","Egresos",'location','northwest')
89 set(gca,'fontsize',10);
90 set(gca,'XTick',1:12);
91 xticklabels(1:12)
92 xlabel ("MESES");
93 ylabel ("Cantidad de pasos promedio");
94 print -djpg graficos/puntoG/Grafico_pasos_prom_x_mes_05.jpg
95
96 #estacion 6
97 [vector_ingresos_prom_x_mes_estacion_6,vector_egresos_prom_x_mes_estacion_6] =
pasos_prom_estacion (data,6)
98 matriz_6 = [vector_ingresos_prom_x_mes_estacion_6,vector_egresos_prom_x_mes_estacion_6]
99 bar(matriz_6)
100 title ("CANTIDAD DE PASOS PROMEDIO POR MES EN LA ESTACION 6");
101 legend("Ingresos","Egresos",'location','northwest')
102 set(gca,'fontsize',10);
103 set(gca,'XTick',1:12);
104 xticklabels(1:12)
105 xlabel ("MESES");
106 ylabel ("Cantidad de pasos promedio");
107 print -djpg graficos/puntoG/Grafico_pasos_prom_x_mes_06.jpg
108
109 #estacion 7
110 [vector_ingresos_prom_x_mes_estacion_7,vector_egresos_prom_x_mes_estacion_7] =
pasos_prom_estacion (data,7)
111 matriz_7 = [vector_ingresos_prom_x_mes_estacion_7,vector_egresos_prom_x_mes_estacion_7]
112 bar(matriz_7)
113 title ("CANTIDAD DE PASOS PROMEDIO POR MES EN LA ESTACION 7");
114 legend("Ingresos","Egresos",'location','northwest')
115 set(gca,'fontsize',10);
116 set(gca,'XTick',1:12);
117 xticklabels(1:12)
118 xlabel ("MESES");
119 ylabel ("Cantidad de pasos promedio");
120 print -djpg graficos/puntoG/Grafico_pasos_prom_x_mes_07.jpg

```

```
121
122 #estacion 8
123 [vector_ingresos_prom_x_mes_estacion_8,vector_egresos_prom_x_mes_estacion_8] =
    pasos_prom_estacion (data,8)
124 matriz_8 = [vector_ingresos_prom_x_mes_estacion_8,vector_egresos_prom_x_mes_estacion_8]
125 bar(matriz_8)
126 title ("CANTIDAD DE PASOS PROMEDIO POR MES EN LA ESTACION 8");
127 legend("Ingresos","Egresos",'location','northwest')
128 set(gca,'fontsize',10);
129 set(gca,'XTick',1:12);
130 xticklabels(1:12)
131 xlabel ("MESES");
132 ylabel ("Cantidad de pasos promedio");
133 print -djpg graficos/puntoG/Grafico_pasos_prom_x_mes_08.jpg
```

```

1  # TP NUMERO 1 -- PEAJES -- punto H
2
3
4  # PUNTO A: load data
5  data = load('dataset/FlujoVehicular2019.dat');
6
7  # mapeo nombres - columnas del dataset
8
9  mes = 1;
10 diames = 2;
11 hora = 3;
12 diasemana = 4;
13 estacion = 5;
14 sentido = 6;
15 tipovehiculo = 7;
16 formapago = 8;
17 cantidadpasos = 9;
18
19 #Punto C:
20 #Pasos totales por estacion.
21
22 vector_pasostot_por_estacion = zeros(8,1);
23 cantidad_de_filas = rows(data);
24
25
26 # PUNTO H
27
28 # formas de pago
29 efectivo = 101;
30 exento = 102;
31 infraccion = 103;
32 no_cobrado = 104;
33 discapacidad = 105;
34 telepase = 106;
35
36 # me interesa saber la cantidad de NO COBRADOS por cada una de las horas del dia y
    ademas la cantidad de TIPOS de no cobrados totales para la hora con mayor cantidad
    de no cobrados.
37
38 # vector contador de no cobrados x hora del dia
39 vector_no_cobrados_hora = zeros (24,1);
40 # creo una matriz que va a contar la cantidad de cada tipo de no cobrados (columnas)
    por cada una de las horas del dia (filas). la ultima columna de la matrix va a
    contener los totales de no cobrados para ese horario
41 matriz_contador_tipos_hora = [102, 103, 104, 105, -1; zeros(24,5)];
42
43 # Opero sobre el data set para realizar el conteo x horarios
44 for fila = 1:cantidad_de_filas
45     # guardo la hora de la fila analizada
46     hora_actual = data(fila, hora) +1;
47     # analizo si el tipo de pago fue no cobrado:
48     if(data(fila, formapago) != 101 && data(fila, formapago) != 106)
49         # sumo totales en la ultima columna para la hora analizada
50         matriz_contador_tipos_hora(hora_actual+1, 5) =
            matriz_contador_tipos_hora(hora_actual+1,5) + data(fila, cantidadpasos);
51         # elijo la fila que voy a escribir segun el tipo de defecto
52         if (data(fila, formapago) == 102)
53             fila_a_escribir = 1;
54         elseif (data(fila, formapago) == 103)
55             fila_a_escribir = 2;
56         elseif (data(fila, formapago) == 104)
57             fila_a_escribir = 3;
58         elseif (data(fila, formapago) == 105)
59             fila_a_escribir = 4;
60         endif

```

```

61     # sumo el tipo de forma de pago para la hora analizada
62     matriz_contador_tipos_hora(hora_actual +1, fila_a_escribir) =
        matriz_contador_tipos_hora(hora_actual +1, fila_a_escribir) + data(fila,
        cantidadpasos);
63     endif
64 endfor
65
66 disp('---Cantidad de no cobrados x hora del dia:');
67 disp(matriz_contador_tipos_hora);
68
69 # analizo cual es la hora del dia con mayor cantidad de casos no pagos.
70 maximo = 0;
71 hora_del_maximo = -1;
72 for x = 1:rows(matriz_contador_tipos_hora)
73     if(matriz_contador_tipos_hora(x,5) > maximo)
74         maximo = matriz_contador_tipos_hora(x,5);
75         hora_del_maximo = x;
76     endif
77 endfor
78
79 # para todas las horas calculo el maximo de cada uno de los tipos
80 vector_totales_por_tipo = sum(matriz_contador_tipos_hora(2:24,1:4));
81 # analizo de los totales por tipo cual es el que mas incide
82 maximo_total = 0;
83 tipo_del_maximo = -1;
84 for col = 1:4
85     if(vector_totales_por_tipo(1,col) > maximo_total)
86         maximo_total = vector_totales_por_tipo(1,col);
87         tipo_del_maximo = col;
88     endif
89 endfor
90 disp('---totales por tipo:');
91 disp(vector_totales_por_tipo);
92 disp('---Analizando la totalidad de datos, el tipo de no pago que mas incide es el
de la columna: (el tipo con mas casos en total)');
93 disp(tipo_del_maximo);
94 disp('con un total de:');
95 disp(maximo_total);
96
97
98 disp('La hora con mayor cantidad de casos no pagos tiene un total de:');
99 disp(maximo);
100 disp('Y es la hora:');
101 disp(hora_del_maximo);
102
103 # grafico la cantidad de no pagados x hora
104 bar(matriz_contador_tipos_hora(2:24,5));
105 title ('CANTIDAD DE NO PAGADOS POR HORA');
106 set(gca,'fontsize',10);
107 set(gca,'xTick',0:23);
108 ylim ([0 800000]);
109 xlabel ("Horas");
110 ylabel ("Cantidad de no pagos");
111 print -djpg graficos/puntoH/Grafico_cantidad_no_pagados_por_hora.jpg
112 # grafico los distintos tipos de codigos de no pagado para la hora con mayor
cantidad de casos no pagos. (luego de saber cual es la hora del maximo)
113 bar(matriz_contador_tipos_hora(hora_del_maximo,1:4));
114 title ('CANTIDAD POR CADA TIPO PARA LA HORA MAXIMA');
115 set(gca,'fontsize',10);
116 set(gca,'xTick',1:4);
117 xticklabels({'Exento','Infraccion','No Cobrado','Discapacidad'})
118 xlabel ("Tipo de Incobrabilidad");
119 ylabel ("Cantidad de no pagos");
120 print -djpg graficos/puntoH/Grafico_distribucion_de_tipos_nopago_en_hora_del_maximo.jpg
121 # grafico los totales por tipo en la totalidad del tiempo

```

```
122 bar(vector_totales_por_tipo);
123 title ('CANTIDADES TOTALES POR CADA TIPO HISTORICO');
124 set(gca,'fontsize',10);
125 set(gca,'xTick',1:4);
126 xticklabels({'Exento','Infraccion','No Cobrado','Discapacidad'})
127 xlabel ("Tipo de Incobrabilidad");
128 ylabel ("Cantidad de no pagos");
129 print -djpg graficos/puntoH/Grafico_distribucion_de_tipos_nopago_historico.jpg
130
```

```

1  # TP NUMERO 1 -- PEAJES punto I
2
3
4  # PUNTO A: load data
5  data = load('dataset/FlujoVehicular2019.dat');
6
7  # mapeo nombres - columnas del dataset
8
9  mes = 1;
10 diames = 2;
11 hora = 3;
12 diasemana = 4;
13 estacion = 5;
14 sentido = 6;
15 tipovehiculo = 7;
16 formapago = 8;
17 cantidadpasos = 9;
18 cantidad_de_filas = rows(data);
19
20 # PUNTO I
21 # Realizar una serie diaria anual de pasos pagados segun las modalidades efectivo y
    telepase.
22 # por cada dia en un año queiro tener un vector que tenga la cantidad de efectivos,
    telepases, y uno que sea la suma de ambos
23 # Guardamos los datos de efectivo, telepase y totales en tres matrices de 31x12 que
    tendran como columnas meses y como filas los dias del año
24
25 matriz_efectivo = [1,2,3,4,5,6,7,8,9,10,11,12;zeros(31,12)];
26 matriz_telepase = [1,2,3,4,5,6,7,8,9,10,11,12;zeros(31,12)];
27 matriz_totales = [1,2,3,4,5,6,7,8,9,10,11,12;zeros(31,12)];
28
29 for fila = 1:cantidad_de_filas
30     if(data(fila, formapago)==101)
31         # si pago en efectivo
32         mes_actual = data(fila,mes);
33         dia_actual = data(fila,diames);
34         if(mes_actual != 0 && dia_actual != 0)
35             matriz_efectivo(dia_actual+1, mes_actual) = matriz_efectivo(dia_actual+1,
                mes_actual) + data(fila, cantidadpasos);
36             matriz_totales(dia_actual+1, mes_actual) = matriz_efectivo(dia_actual+1,
                mes_actual) + matriz_telepase(dia_actual+1, mes_actual);
37         endif
38     elseif(data(fila, formapago)==106)
39         # si pago con telepase
40         mes_actual = data(fila,mes);
41         dia_actual = data(fila,diames);
42         if(mes_actual != 0 && dia_actual != 0)
43             matriz_telepase(dia_actual+1, mes_actual) = matriz_telepase(dia_actual+1,
                mes_actual) + data(fila, cantidadpasos);
44             matriz_totales(dia_actual+1, mes_actual) = matriz_efectivo(dia_actual+1,
                mes_actual) + matriz_telepase(dia_actual+1, mes_actual);
45         endif
46     endif
47 endfor
48
49 disp('serie anual- pagos en efectivo');
50 disp(matriz_efectivo);
51
52 disp('serie anual- pagos con telepase');
53 disp(matriz_telepase);
54
55 disp('serie anual- pagos con telepase + pagos con efectivo');
56 disp(matriz_totales);
57
58 # creo las series temporales sumando cada una de las columnas de la matriz menos la

```

```

59 primera fila.
60 serie_temporal_efectivo = [];
61 serie_temporal_telepase = [];
62 serie_temporal_totales = [];
63
64 for col = 1:12
65     serie_temporal_efectivo = [serie_temporal_efectivo;matriz_efectivo(2:32,col)];
66     serie_temporal_telepase = [serie_temporal_telepase;matriz_telepase(2:32,col)];
67     serie_temporal_totales = [serie_temporal_totales;matriz_totales(2:32,col)];
68 endfor
69
70 disp('Serie temporal efectivo');
71 disp(serie_temporal_efectivo);
72 disp('Serie temporal telepase');
73 disp(serie_temporal_telepase);
74 disp('Serie temporal totales');
75 disp(serie_temporal_totales);
76 disp(rows(serie_temporal_efectivo));
77
78 plot(nonzeros(serie_temporal_efectivo));
79 title ('SERIE TEMPORAL EFECTIVO');
80 set(gca,'fontsize',10);
81 set(gca,'xTick',1:30:365);
82 xlabel ("DIAS");
83 ylabel ("Cantidad de pagos");
84 print -djpg graficos/puntoI/Grafico_serie_temporal_efectivo.jpg
85
86 plot(nonzeros(serie_temporal_telepase));
87 title ('SERIE TEMPORAL TELEPASE');
88 set(gca,'fontsize',10);
89 set(gca,'xTick',1:30:365);
90 xlabel ("DIAS");
91 ylabel ("Cantidad de pagos");
92 print -djpg graficos/puntoI/Grafico_serie_temporal_telepase.jpg
93
94 plot(nonzeros(serie_temporal_totales));
95 title ('SERIE TEMPORAL EFECTIVO + TELEPASE');
96 set(gca,'fontsize',10);
97 set(gca,'xTick',1:30:365);
98 xlabel ("DIAS");
99 ylabel ("Cantidad de pagos");
100 print -djpg graficos/puntoI/Grafico_serie_temporal_totales.jpg
101

```