

**Instituto Tecnológico y de Estudios Superiores de  
Monterrey**



Modelación de sistemas mínimos y  
arquitecturas computacionales (Gpo 570)

**Avance 3 – Reto (Entrega Final)**

**Marlon Yahir Martínez Chacón - A01424875**

01/02/2023

La forma en como la humanidad se transporta de un lado a otro ha cambiado con el paso del tiempo, durante un largo periodo de tiempo el medio de transporte por excelencia para recorrer tanto largas como cortas distancias eran los seres vivos, en este caso el caballo que dependiendo del lugar donde vivías podía ser un camello o un burro; después este medio se fue desarrollando hasta que los animales dejaron de ser necesarios y se introdujeron los primeros autos que eran impulsados con un motor. Los automóviles fueron mejorando con el paso de los años aumentando su potencia y las funciones que existen para mejorar la experiencia del conductor, como el uso de las pantallas para usar todas las funciones del auto para controlar el aire acondicionado, reclinar los asientos, entre otras cosas, y proteger su integridad física como los sensores para notificar al conductor y los pasajeros que utilicen el cinturón de seguridad, y también las bolsas de aire en caso de accidente. Todavía siguen existiendo problemas dentro de la industria automotriz que no vemos a simple vista pero que sucesos que aparecen en el mundo nos hacen reflexionar acerca de encontrar una solución para resolver el problema, uno de esos sucesos es el siguiente.

Durante finales de 2019 y los años posteriores experimentamos algo que pensamos que no volvería a suceder, un virus que fue llamado covid-19 se expandió por todo el mundo y una pandemia inició, esta enfermedad era muy contagiosa por lo que se implementaron muchas medidas para poder prevenirla y también detectarla a tiempo para que las personas sean conscientes de su condición de salud y en caso de estar infectado sea atendido a la brevedad para no afectar a las demás personas de su alrededor, estas medidas que se implementaron para la detección eran tomar la temperatura de las personas que acudían a lugares cerrados donde el virus se propagaba de manera más rápida, estos lugares eran los supermercados, oficinas, bancos, centro comerciales, escuelas, etc., en la actualidad esta medida de tomar la temperatura sigue presente en algunos establecimientos alrededor del mundo, esto es porque a pesar de que una gran parte de la población ya cuenta con alguna vacuna para protegerse del virus siguen surgiendo más variantes que ocasionan más casos. Durante la pandemia pudimos observar cómo ha cambiado la forma en como nos tomamos la temperatura, ya que en los establecimientos para no tener contacto con los clientes se usaban termómetros infrarrojos para conocer la temperatura y mediante estas medidas nos recordaron que es muy importante estar al tanto de ese dato porque puede ayudarnos a detectar si es que estamos enfermos.



Uno de los problemas que encontré en la industria automotriz es que un automóvil carece de sensores que sirvan como un medio para poder detectar los datos que estén relacionados con la salud del conductor, en este caso no me refiero al cinturón de

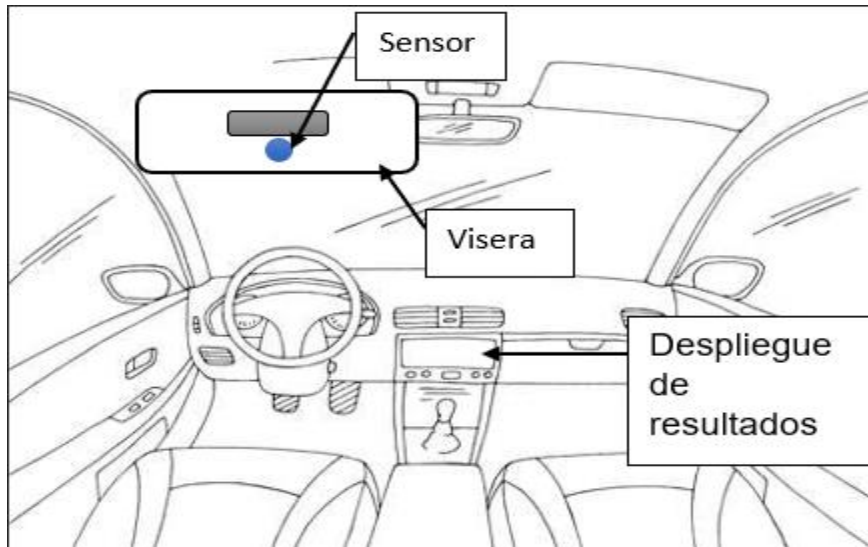
seguridad o airbags que están para proteger al conductor y sus pasajeros, sino a datos cómo el ritmo cardíaco, la sudoración, la temperatura corporal, entre otros; para dar solución al problema me enfocaré en la temperatura corporal, debido a que como mencioné anteriormente es un dato que es de mucha ayuda para tener un panorama de la salud de una persona en este caso del conductor, por lo que es importante implementar una solución a esta carencia existente para que podamos cuidar mejor nuestra salud, y seamos conscientes de nuestra condición para que en dado caso de ser necesario acudir a un médico para ser tratados y también prevenir contagiar a las personas que nos rodean para que no se extienda lo que hemos estado viviendo durante los último años.

La información que se busca recolectar con esta solución es la temperatura corporal del conductor, dependiendo del país en el que haya comprado el automóvil, la unidad de medida de la temperatura pueden ser los grados Celsius ( $^{\circ}\text{C}$ ) o los grados Fahrenheit ( $^{\circ}\text{F}$ ), la temperatura corporal tiene como características que es volátil y temporal, por lo que dependiendo del estado emocional o físico en que nuestro cuerpo se encuentre y de factores externos como la temperatura del ambiente esta puede cambiar conforme pasa el tiempo; el rango de temperatura corporal normal que tiene un cuerpo humano se establece entre  $36.1^{\circ}\text{C}$  a  $37.2^{\circ}\text{C}$  ( $97^{\circ}\text{F}$  a  $99^{\circ}\text{F}$ ), por lo que si superamos el límite superior ya podría llegar a ser considerado como fiebre y puede ser necesario tomar ciertas medidas para que no incremente y empeore.

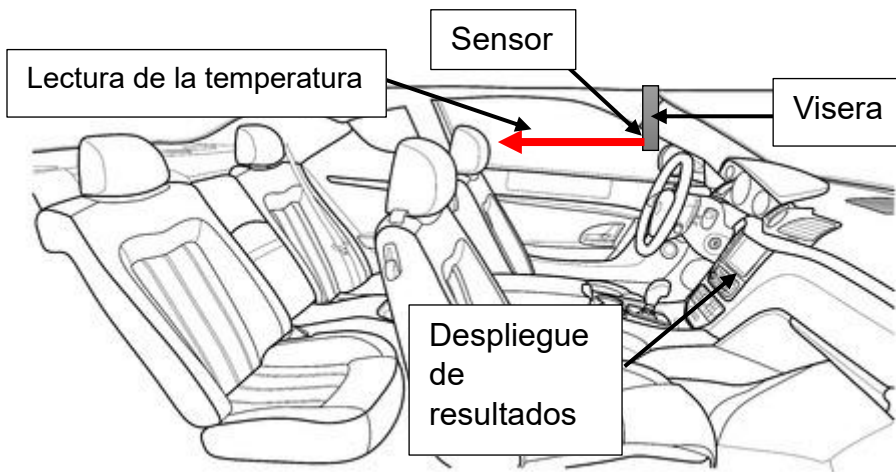
En la actualidad, existen muchos sensores para poder medir las variables que se encuentran a nuestro alrededor y con las que interactuamos día con día, el que propongo utilizar para poder medir la temperatura corporal del conductor sería necesario implementar en el automóvil un sensor de temperatura corporal infrarrojo, esta clase de sensores fueron muy utilizados en los termómetros infrarrojos que se utilizaron durante la pandemia ya que no se necesitaba tener contacto con la otra persona para conocer su temperatura, por lo que queda perfecto para obtener la variable que queremos medir para que el usuario no se sienta incómodo con la medición; lo único que se necesita para que el sensor funcione es apuntarlo a una parte del cuerpo en este caso la frente ya que es lo más recomendable de medir si es que buscamos obtener la mejor lectura de la temperatura corporal y cuando la temperatura sea tomada el sensor mandará la información mediante señales eléctricas a una computadora.



Para poder una medida de la temperatura de la frente del conductor es necesario poner el sensor a una buena altura para que la persona no tenga problemas a la hora de realizar la toma de temperatura, el mejor lugar sería en la visera del lado del conductor ya que al bajarla para usar el espejo incorporado queda justo a la altura de la frente, además de que al estar integrada en un elemento que siempre ha estado en los automóviles no llega a estorbar para el conductor y una vez que se tome la temperatura el resultado de esa lectura será mostrado en la pantalla principal del automóvil para que el conductor lo vea.



Sensor de temperatura infrarrojo



Resumiendo, mi propuesta para medir la temperatura corporal sería integrar el sensor de temperatura corporal infrarroja en la visera del conductor para que en cuanto encienda el auto se le notifique que debe realizarse la lectura de la temperatura y el automóvil baje de manera automática la visera o el conductor lo haga de manera manual, una vez que el sensor capture la temperatura se mandará esa información mediante señales eléctricas a la computadora del automóvil para que sea interpretada, y al final el resultado sea mandado a la pantalla del automóvil para que el conductor lo conozca en  $^{\circ}\text{C}$  o  $^{\circ}\text{F}$ , y se le notifique si es que su temperatura está dentro del promedio o es mayor, para que de esta forma el conductor este consciente de su condición actual y tome las medidas necesarias.

En la actualidad, una gran parte de las computadoras tienen la arquitectura de Von-Neumann y muchas cosas que tenemos a nuestro alrededor tienen una computadora integrada que recibe entradas para que sean analizadas y podamos visualizar sus resultados, con el paso del tiempo las computadoras empezaron a tener más hardware que pudiera recibir entradas de los usuarios como teclados, controles, mouse, etc, por lo que la arquitectura fue muy utilizada y lo sigue siendo para poder procesar toda la información que recibe para que el usuario la pueda visualizar, sin esta arquitectura no tendríamos tantos avances en el sector salud como por ejemplo el poder visualizar el ritmo cardíaco de un paciente recibiendo su pulso como una entrada para que sea analizado y podamos ver una gráfica en una pantalla, también en la industria del entretenimiento para disfrutar de los videojuegos donde las consolas reciben las entradas de los jugadores para mover al personaje y hacer al juego interactivo; pero en cuanto al sensor que voy a ocupar se puede observar la arquitectura cuando el sensor recibe la temperatura para que sea enviada a la unidad de control para que la unidad de aritmética y lógica pueda hacer su trabajo interpretando la información que recibe y de esta forma, en la pantalla que se encuentra en el automóvil el conductor y los pasajeros presentes puedan visualizar la temperatura y conocer un poco acerca de su estado de salud.

En cuanto a la implementación del sensor de temperatura corporal, el siguiente paso sería establecer una comunicación con el microcontrolador, en este caso se busca que el microcontrolador reciba información de la temperatura corporal del usuario mediante el sensor. Pero para poder lograr esta comunicación se necesita realizar un programa que le dé las instrucciones necesarias al microcontrolador para que sepa cómo es que debe de recibir los datos, guardarlos y procesarlos; este programa como se verá más adelante se realizará mediante MARIE.js usando el lenguaje ensamblador.

Pero como se menciona, el ambiente en que se desarrollará y probará el programa para solucionar el reto es uno diferente al que se ocupa en la vida real, es decir, el programa lo vamos a desarrollar en un simulador que nos permite programar en lenguaje ensamblador, por lo que no lo estaremos programando directamente en el microcontrolador que usa un auto por lo que existirán algunas diferencias que podrían afectar el rendimiento de nuestro programa y de cómo son procesadas las lecturas del sensor. Por lo que es importante conocer las principales diferencias en las características que tiene un microcontrolador de un auto y nuestro simulador de desarrollo que será MARIE.

Para realizar la comparación nos enfocaremos en las características que tiene el Microcontrolador V850E1 el cual es usado en los autos de la marca Toyota, entre otras marcas de autos y lo compararemos con el microcontrolador PIC18F45K50, el cual tiene características muy similares al simulador MARIE.js, además que es un objeto físico, por lo que se obtiene una mejor comparación entre estos dos microcontroladores.

Estas son las principales diferencias entre ambos microcontroladores.

Características	V850E1 (Vida real)	PIC18F45K50 (Reto)
Tamaño de la memoria de programa	512, 1024, 1280, 1536 kb	32 kb
Tamaño Memoria RAM	60,76,92 kb	2 kb
Canales conversores analógico a digital (ADC)	10-bit x 16 canales	10-bit x 25 canales
Canales conversores de digital a analógico (DAC)	8-bit x 2 canales	5-bit x 1 canal
Temporizadores	16-bit x 13 canales	8-bit x 2 canales – 16-bit x 2 canales
Rango de temperatura	-40 – 85°C	-40 – 125°C
Puertos I/O	128,156	15,25,36
Voltaje de alimentación	2.85 – 3.6 V	1.8 – 5.5 V

## PARTE 1: BASES DEL CÓDIGO

Para el funcionamiento de la propuesta, el sensor que vamos a implementar en el auto nos mandará lecturas a través de un bus paralelo a razón de 12 bits/ms por lo que se realizó un programa que reciba esta lectura y las almacene para después los podamos manipular, el programa se realizó usando el simulador de MARIE.js programando en ensamblador. El programa consiste en un bucle que recibe las primeras 20 mediciones que nos va a mandar el sensor para que las guardemos en la memoria y después podamos realizar cálculos con estos valores.

### Código completo

```

1  ORG 100
2  //Marlon Yahir Martínez Chacón
3  //Inicio del loop (1 al 20) para introducir los 20 datos
4  Start,Input
5  StoreI direccion //se piden valores y se almacenan en la direccion
6  Load direccion //se carga al AC el valor de la direccion
7  Add unoHex // se le suma uno a la dirección en donde se guarda para moverla un lugar
8  Store direccion //Se guarda el nuevo valor de la direccion
9
10 //i++
11 Load i //se carga el valor que sirve como contador
12 Add uno //Se le suma uno para que no sea infinito
13 Store i //se guarda el nuevo valor del contador para avanzar en los valores
14
15 //Condicion i<=20
16 subtr veinte //se le resta 20 al nuevo valor del contador
17 Skipcond 800//se comprueba si la resta es mayor que 0 para conocer si i>20
18 Jump Start //Si i es menor o igual que 20 entonces regresa al inicio del loop
19
20
21 //Termina la ejecución
22 Final,Halt
23
24 i, dec 1 //Se inicializa i (contador) en 1
25 uno, dec 1 //Se inicializa el valor que se va a sumar al final del loop (constante)
26 veinte, dec 20 //Se inicializa el valor que es el número máximo de entradas
27 direccion, hex 120 //Se inicializa la direccion a donde se va a mandar el primer dato
28 unoHex, hex 001 //Se inicializa el valor que modificara la direccion
29 //para que al final del loop se mueva un lugar

```

Después de ensamblar el código, se almacenarán en la memoria los valores de las instrucciones junto a los valores que existen en cada una de las direcciones esto conformará cada uno de los espacios de 16 bits que tenemos.

## Memoria al ensamblar

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
100	5000	E110	1110	3111	2110	110D	310E	210D	410F	8800	9100	910C	7000	0001	0001	0014
110	0120	0001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
120	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Estas son las etiquetas que vamos a usar en nuestro programa que nos ayudaran a dirigirnos a las direcciones correctas.

## Tabla de etiquetas al ensamblar

Output log	RTL log	Watch list	Input list
Label		Address	Value
Start		100	5000
Final		10C	7000
i		10D	0001
uno		10E	0001
veinte		10F	0014
direccion		110	0120
unoHex		111	0001

Y para probar que los datos que vamos a recibir de nuestro sensor se van a almacenar de manera correcta en nuestro programa los vamos a probar con las primeras 20 mediciones las cuáles serán las siguientes:

## Input list

Output log

RTL log

Watch list

Input list

INPUT MODE:

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

Output log

RTL log

Watch list

Input list

INPUT MODE:

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

Como se puede observar serán los números del 1 al 20 para que al momento de que queden guardados en nuestra memoria lo podamos de reconocer de manera más sencilla y podamos comprobar que funciona correctamente. Estos datos serán guardados mediante lo equivalente a un bucle for que fue implementado cambiando los valores del contador y también de la dirección que usábamos como un apuntador a la dirección donde buscamos almacenar el valor.



Momentos después de correr el programa introduciendo las primeras 20 mediciones que vamos a recibir del sensor, la memoria quedo con los siguientes valores.

## Memoria después de correr el programa

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0F0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
100	5000	E110	1110	3111	2110	110D	310E	210D	410F	8800	9100	910C	7000	0015	0001	0014
110	0134	0001	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
120	0001	0002	0003	0004	0005	0006	0007	0008	0009	000A	000B	000C	000D	000E	000F	0010
130	0011	0012	0013	0014	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

Como se puede observar las instrucciones se encuentran del 100 al 111 y como se puede observar de la dirección 120 a 133 se agregaron 20 valores más, los cuales corresponden a los 20 valores que recibimos de parte de nuestro sensor.

Y así es como queda nuestra tabla de etiquetas al final.

## Tabla de etiquetas después de correr el programa

Label	Address	Value
Start	100	5000
Final	10C	7000
i	10D	0015
uno	10E	0001
veinte	10F	0014
direccion	110	0134
unoHex	111	0001

La memoria con la que cuenta el microcontrolador que recibirá las lecturas es limitada, por lo que es necesario tener en cuenta el tamaño de nuestro programa para que no ocupe mucho espacio y podamos recibir más lecturas para guardarlas. El programa que realizamos en MARIE.js cuenta con 18 líneas de instrucciones por lo que son 18 espacios en la memoria de 16 bits cada uno de ellos por lo que serían 288 bits de espacio en la memoria, pero aparte de las instrucciones también tenemos que tener en cuenta las 20 mediciones que guardamos con nuestro programa que serían en total 320 bits, sumando todos los bits que se ocupan en nuestro programa por los 38 espacios de memoria de 16 bits nos darían un total de **608** bits de memoria, para los microcontroladores que tenemos actualmente el espacio que vamos a ocupar no representa una gran parte por lo que no tendremos problema para usarlo para obtener las lecturas de nuestro sensor que medirá la temperatura del usuario.

## PARTE 2: IMPLEMENTACIÓN DE FUNCIONES AL CÓDIGO

Una vez que ya se realizó la simulación de la comunicación que existe entre el sensor de temperatura que implementamos y nuestro programa desarrollado en Marie, el siguiente paso es procesar esos datos para poder manipularlos y mostrarle la información que es esencial que el usuario vea para que nuestro sensor cumpla con su objetivo.

Las líneas de código que se agregaron al programa que ya tenía tienen como función que una vez hayamos guardado las 20 mediciones del sensor, sumar todos esos valores y una vez que ya tengamos la suma total dividirlo entre el número total de las mediciones, es decir, entre 20 para poder obtener el promedio o la media estadística que será desplegada para que el usuario pueda conocer la temperatura corporal que tiene en ese momento, pero además de desplegar la temperatura también se agregaron otros segmentos de código para que dependiendo de si el promedio de la temperatura es mayor a 37 o menor a 36 se le desplegará un símbolo que representa si es que la temperatura es normal para un ser humano o no (ya sea alta o baja), la investigación que se realizó fue usada para establecer lo que se indica como una temperatura normal ya que como se menciona la temperatura normal de un humano se establece entre 36.1°C a 37.2°C (97°F a 99°F); los símbolos se agregaron para que el usuario pueda saber de una manera más gráfica mediante una palomita verde y una tache rojo, los cuales son símbolos que todas las personas asocian con algo correcto o incorrecto, si es que su temperatura es buena o está enfermo.

Uno de los cambios principales que se implementó fue que a diferencia del código anterior donde las 20 mediciones se guardaban del 120 en adelante, en el código con el promedio se cambio a que fuera del 150 en adelante, esto es debido a que con las líneas de código que se agregaron el tamaño del código aumentó por lo que lo mejor fue

cambiarlo a una dirección más lejana para que no tuviera problema sobrescribiendo instrucciones.

## Código completo

```
1  ORG 100
2  //Marlon Yahir Martínez Chacón
3  //Inicio del loop (1 al 20) para introducir los 20 datos
4  Start,Input
5  StoreI direccion //se piden valores y se almacenan en la direccion
6
7  //Sumar los valores que se ingresan
8  Load sumaValores //Se carga la sumaValores en el AC
9  AddI direccion //Se le suma el valor de la direccion a la que apunta la etiqueta direccion
10 Store sumaValores //Se guarda la suma en sumaValores
11
12 Load direccion //se carga al AC el valor de la direccion
13 Add unoHex // se le suma uno a la dirección en donde se guarda para moverla un lugar
14 Store direccion //Se guarda el nuevo valor de la direccion
15
16 //i++
17 Load i //se carga el valor que sirve como contador
18 Add uno //Se le suma uno para que no sea infinito
19
20
21 Store i //se guarda el nuevo valor del contador para avanzar en los valores
22
23 //Condicion i<=20
24 subtl veinte //se le resta 20 al nuevo valor del contador
25 Skipcond 800//se comprueba si la resta es mayor que 0 para conocer si i>20
26 Jump Start //Si i es menor o igual que 20 entonces regresa al inicio del loop
27 Jump DivisionPromedio //Si i es mayor que 20 terminar el programa
28
29 //Empieza el loop para la division
30 DivisionPromedio,Load sumaValores //Carga el valor que almacenara la suma de los valores
31 Subtl veinte//Se le resta 20 a la sumaValores para saber si cabe 20 en la suma
32 Store sumaValores //Guarda el resultado de la resta en la misma sumaValores
33 Skipcond 000 //Checa si el número 20 (siendo el resultado menor que 0)
34 //no cabe en la sumaValores
35 Jump SumarUno //Si es mayor o igual que 0 entonces salta para sumarle uno al cociente
36 Jump Resultado //Si es menor que 0 entonces se llegó al resultado de la división
```

```
35
36 //Se obtiene el cociente (media estadística) sumándole uno a su valor
37 SumarUno, Load mediaEstadistica //Carga el valor de la media estadística en el AC
38 Add uno //Se le suma el valor de uno a la media estadística
39 Store mediaEstadistica //Se guarda el nuevo valor de la media estadística
40 Jump DivisionPromedio //Salta de vuelta al loop de la division
41
42 //Una vez que se completa la división despliega el resultado
43 Resultado, Load mediaEstadistica //Carga la mediaEstadística en el AC y lo despliega
44 Output
45 //Obtener si la temperatura es mayor que 37 o menor
46 Subt treintasiete //le resta 37 a la mediaEstadistica
47 Skipcond 800 //Comprueba si el valor de la resta es mayor que 0 (mayor que 37)
48 Jump mayorQue35//Si es menor que 0 entonces salta hacia desplegar la palomita verde
49 Jump mostrarTache//Si es mayor que 0 entonces salta para desplegar una tache
50
51 mayorQue35, Load mediaEstadistica //Carga la mediaEstadística en el AC y lo despliega
52
53
52 //Obtener si la temperatura es mayor que 36 o menor
53 Subt treintacinco //le resta 35 a la mediaEstadistica
54 Skipcond 800 //Comprueba si el valor de la resta es mayor que 0 (mayor que 35)
55 Jump mostrarTache //Si es menor o igual que 0 entonces salta para desplegar una tache
56 Jump mostrarPaloma //Si es mayor que 0 entonces salta para desplegar una palomita verde
57
58
59 //Despliega un simbolo con respecto a la temperatura del usuario
60 //El simbolo representa que la temperatura del usuario es normal (menor a 38)
61 mostrarPaloma, Load normal // Carga el simbolo de la palomita y lo despliega
62 Output
63 Jump AsignarValor//Salta hacia donde se les asigna el valor de 0 a los valores utilizados
64
65 //El simbolo representa que la temperatura del usuario es alta (mayor a 37)
66 mostrarTache, Load alta // Carga el simbolo de la tache y lo despliega
67 Output
68 Jump AsignarValor//Salta hacia donde se les asigna el valor de 0 a los valores utilizados
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84 //VARIABLES
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

```

85 i, dec 1 //Se inicializa i (contador) en 1
86 direccion, hex 150 //Se inicializa la direccion a donde se va a mandar el primer dato
87 sumaValores, dec 0 // Se inicializa el espacio que almacenara la suma de los valores en 0
88 mediaEstadistica, dec 0 //Se inicializa la media estadistica en 0
89
90 //CONSTANTES
91 unoHex, hex 001 //Se inicializa el valor que modificara la direccion
92 //para que al final del loop se mueva un lugar
93 uno, dec 1 //Se inicializa el valor que se va a sumar al final del loop (constante)
94 veinte, dec 20 //Se inicializa el valor que es el número máximo de entradas
95 normal, hex 2705 //Se inicializa el valor de normal con el simbolo de palomita verde
96 alta, hex 274C //Se inicializa el valor de alta con el simbolo de una tache roja
97 treintasiete, dec 37 //Se inicializa el valor de treintasiete con 37
98 treintacinco, dec 35 //Se inicializa el valor de treintasiete con 37
99 cero, dec 0 //Se inicializa el valor de cero con 0
100 cientoCincuenta, hex 150 //Se inicializa el valor cientoCincuenta con 150 hex

```

## Memoria al ensamblar

Así es como queda nuestra memoria al ensamblar nuestro programa.

	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+A	+B	+C	+D	+E	+F
0E0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
0F0	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
100	5000	E133	1134	B133	2134	1133	3136	2133	1132	3137	2132	4138	8800	9100	910F	1134
110	4138	2134	8000	9115	9119	1135	3137	2135	910F	1135	6000	413B	8800	911F	9127	1135
120	413C	8800	9127	9124	1139	6000	912A	113A	6000	912A	1137	2132	113D	2134	2135	113E
130	2133	9100	0001	0150	0000	0000	0001	0001	0014	2705	274C	0025	0023	0000	0150	0000
140	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000
150	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000	0000

## Tabla de etiquetas al ensamblar

Y estas son las etiquetas que vamos a ocupar para nuestro programa, que nos ayudan a dirigirnos a las direcciones correctas.

Output log	RTL log	Watch list	Input list
Label	Address	Value	
Start	100	5000	
DivisionPromedio	10F	1134	
SumarUno	115	1135	
Resultado	119	1135	
mayorQue35	11F	1135	
mostrarPaloma	124	1139	
mostrarTache	127	113A	
AsignarValor	12A	1137	
i	132	0001	
direccion	133	0150	
sumaValores	134	0000	
mediaEstadistica	135	0000	
unoHex	136	0001	
uno	137	0001	
veinte	138	0014	
normal	139	2705	
alta	13A	274C	
treintasiete	13B	0025	
treintacinco	13C	0023	
cero	13D	0000	
cientoCincuenta	13E	0150	

## Input list

Como las mediciones que vamos a recibir de parte de nuestro sensor son de la temperatura corporal de un cuerpo humano, los valores que recibamos pueden estar entre 34°C y 40°C, por lo que para probar nuestro programa ingresaremos valores entre estos números para simular que nuestro sensor ya esta implementado y esta tomando la temperatura a una persona.

[Output log](#) [RTL log](#) [Watch list](#) [Input list](#)

INPUT MODE:

36

38

36

38

39

40

39

34

37

36

35

37

36

39

35

35

39

39

37

35

## Memoria después de correr el programa

[illegible]

Como se puede observar las instrucciones se encuentran del 100 al 13E y como se puede observar de la dirección 150 a 163 se agregaron 20 valores más, los cuales corresponden a los 20 valores que recibimos de parte de nuestro sensor. Y dentro de los espacios que están para las instrucciones se pueden observar los valores que almacenan la suma de valores, promedio y la dirección que aumenta; pero aquí se observan en 0 debido a que se les asignó para que volviera a empezar y tomara otras 20 mediciones.

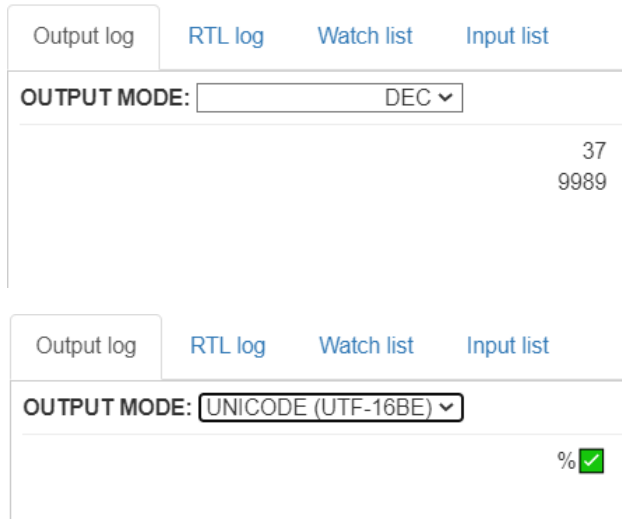
## Tabla de etiquetas después de correr el programa

Output log	RTL log	Watch list	Input list
Label	Address	Value	
Start	100	5000	
DivisionPromedio	10F	1134	
SumarUno	115	1135	
Resultado	119	1135	
mayorQue35	11F	1135	
mostrarPaloma	124	1139	
mostrarTache	127	113A	
AsignarValor	12A	1137	
i	132	0001	
direccion	133	0150	
sumaValores	134	0000	
mediaEstadistica	135	0000	
unoHex	136	0001	
uno	137	0001	
veinte	138	0014	
.	139	2705	
normal	139	2705	
alta	13A	274C	
treintasiete	13B	0025	
treintacinco	13C	0023	
cero	13D	0000	
cientoCincuenta	13E	0150	

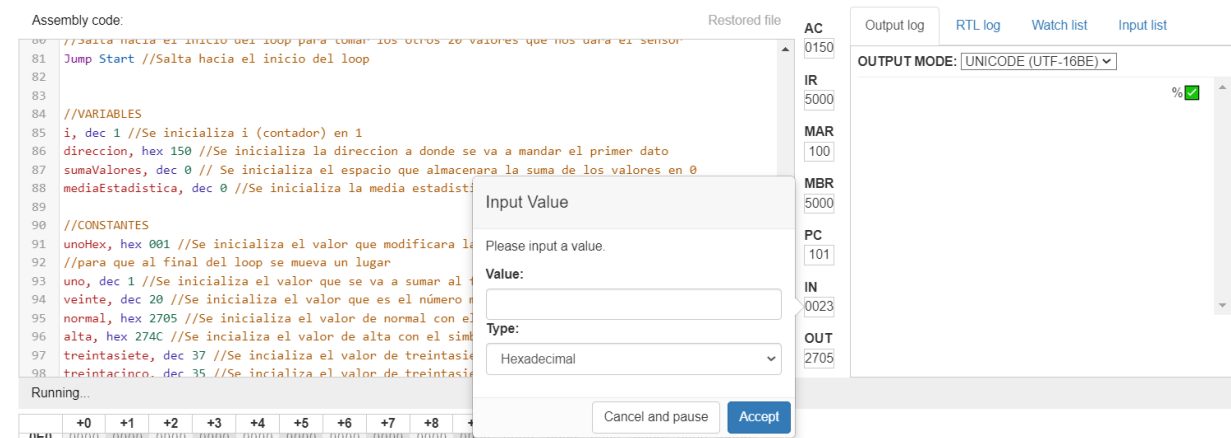


## OUTPUT

Y esto es lo que despliega en pantalla al usuario el valor del promedio de la temperatura y también el símbolo que representa si es que el promedio es de una temperatura normal o este enfermo.



Y por último se puede observar cómo es que el bucle vuelve a empezar, ya que una vez que te despliega el promedio de la temperatura y el símbolo, el programa te pide que ingrese otros 20 valores para volver a obtener el promedio de las nuevas 20 mediciones, y lo pide de manera infinita, ya que un sensor no va a parar de recibir mediciones de la temperatura corporal a menos que se le indique, por lo que de esta forma simulamos más el comportamiento que el sensor tiene.



Como mencioné anteriormente, la memoria con la que cuenta el microcontrolador que recibirá las lecturas es limitada, por lo que es necesario tener en cuenta el tamaño de nuestro programa realizado MARIE.js; a diferencia del programa anterior el nuevo cuenta con 63 líneas de instrucciones por lo que son 63 espacios en la memoria que vamos a ocupar, cada uno de 16 bits, por lo que en total tendríamos 1008 bits de espacio en la

memoria de instrucciones, además de esto debemos de tener en cuenta las 20 mediciones que vamos a recibir por lo que serían 320 bits, en total sumando todos los espacios de memoria que vamos a ocupar nos daría **1328 bits**, esto sigue sin representar mucho problema por la capacidad que tenemos actualmente en nuestros microcontroladores.

Como varias industrias que han surgido con el paso de los años en la historia de la humanidad, la industria automotriz se convirtió en la industria que cambió la forma en la que nos transportamos utilizando mucha tecnología de vanguardia que la ha mantenido como una de las industrias más importante para los humanos ya que de ella depende nuestra movilidad. Y como todas las industrias también se tuvo que ir innovando año tras año para poder cumplir con las nuevas tendencias tecnológicas que surgen día con día para poder explotarlas al máximo y seguir teniendo clientes que buscan la vanguardia, en este caso lo que le paso a la industria automotriz fue una transformación digital ya que pasaron de que sus sistemas funcionaran con el uso de gases que enviaban señales o potencia para poder mover varios mecanismos, a que los sistemas funcionaran con señales eléctricas de las que se obtienen datos que recibe una computadora para poder manipularlos y de esta forma tener una medida precisa de que es lo que se requiere hacer; pero no solo se queda ahí a diferencia del sistema mecánico-neumático, el sistema electrónico-digital nos permite integrar sensores y usar sus datos para tener varias mediciones que podemos utilizar para varias tareas como por ejemplo desplegar la temperatura, velocidad, kilómetros de manera más precisa, gracias al procesamiento que realiza la computadora central de nuestro auto.

Este nuevo sistema es lo que nos permitió realizar nuestro reto, sin esta transformación digital no habría manera de poder establecer una conexión entre el sensor que implementamos con la computadora del carro para recibir sus datos y manipularlos, además de que gracias a la transformación digital podemos manipular la computadora del carro para poder implementar nuestros propios programas para hacer conexión con varios sensores que existen actualmente y poder obtener datos que nos pueden ayudar a solucionar un problema que exista durante el manejo, como la solución que implemente para la situación problema que involucra a una pandemia que ocurrió y que nos afecto de varias formas. La transformación digital nos ha abierto muchas puertas para poder obtener datos de todo lo que nos rodea para resolver problemas que van desde la seguridad hasta la comodidad mientras conducimos.

## Referencias

- Temperatura corporal normal. (s. f.). <https://medlineplus.gov/spanish/ency/article/001982.htm>
- Mr Khan's Classes. (2017, 11 noviembre). Architecture of Computer | What is Von Neumann Architecture [Video]. YouTube. <https://www.youtube.com/watch?v=-SADbPS8UgA>
- Fundamentos y función de la medición IR de la temperatura. (s. f.). Optris. <https://www.optris.es/concepto-basico-de-infrarrojo>
- Saldias, A. (2022, 19 septiembre). Tipos de Sensores de Temperatura. SRC Sistemas de Regulación y Control. <https://srcsl.com/tipos-sensores-temperatura/>
- Corvo, H. S. (2019, 23 octubre). Arquitectura von Neumann: origen, modelo, cómo funciona. Lifeder. <https://www.lifeder.com/arquitectura-von-neumann/>
- Renesas. (s. f.). V850 Family MCUs <https://www.renesas.com/us/en/products/microcontrollers-microprocessors/other-mcus-mpus/v850-family-mcus?field-cpu=V850E1>
- Microchip Technology. (s. f.). PIC18F45K50. <https://www.microchip.com/>
- Newark. (s. f.-b). PIC18F45K50-I/P <https://www.newark.com/microchip/pic18f45k50-i-p/microcontroller-mcu-8-bit-pic18/dp/06W2982>
- posted by: admin. (2022, 13 junio). La transformación digital en la industria automotriz – CENTRO DE INNOVACIÓN INDUSTRIAL PARA EL SECTOR AUTOMOTRIZ TLALNEPANTLA. <https://www.ciia.mx/la-transformacion-digital-en-la-industria-automotriz/>