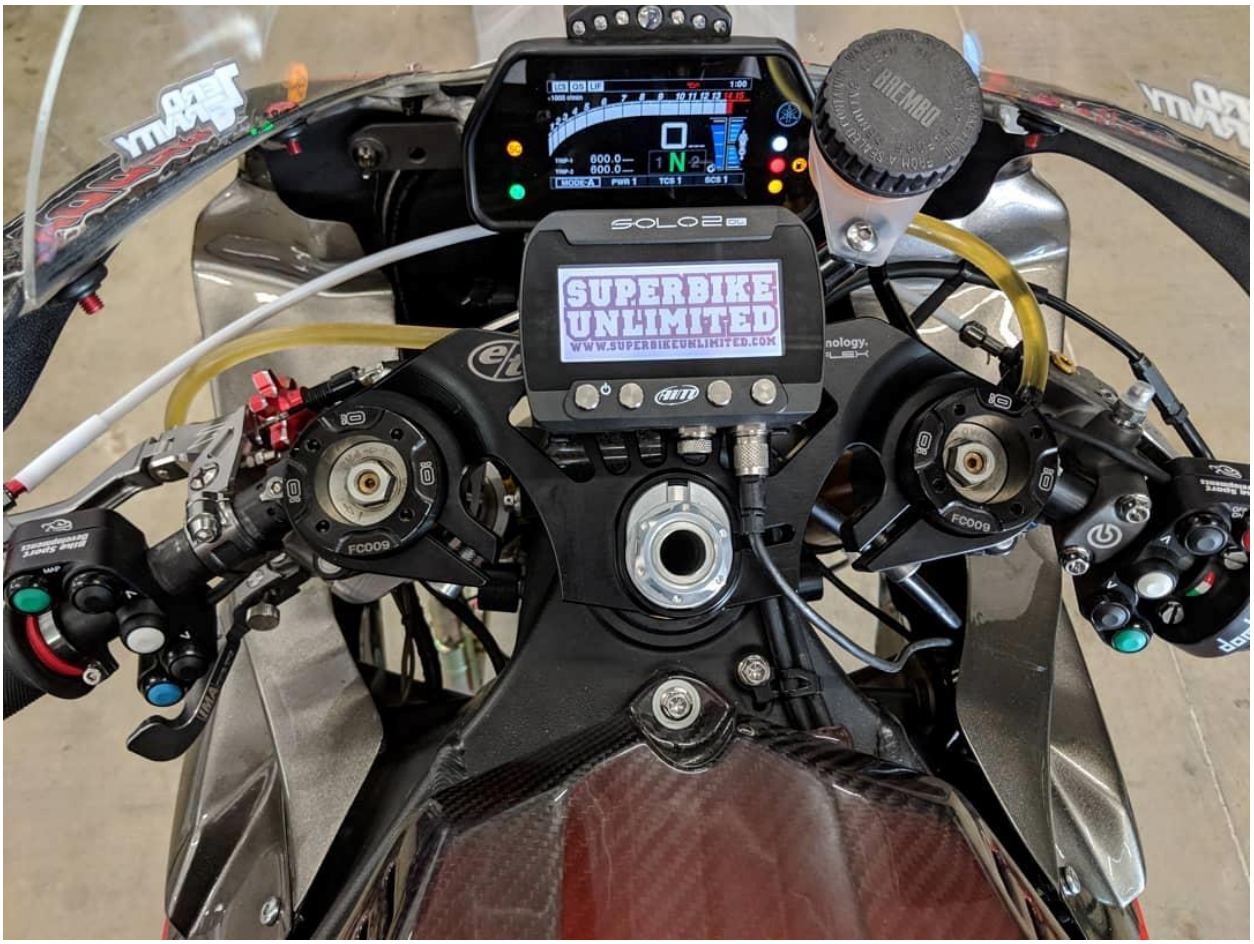
HEAT MAP – edgomez@itba.edu.ar

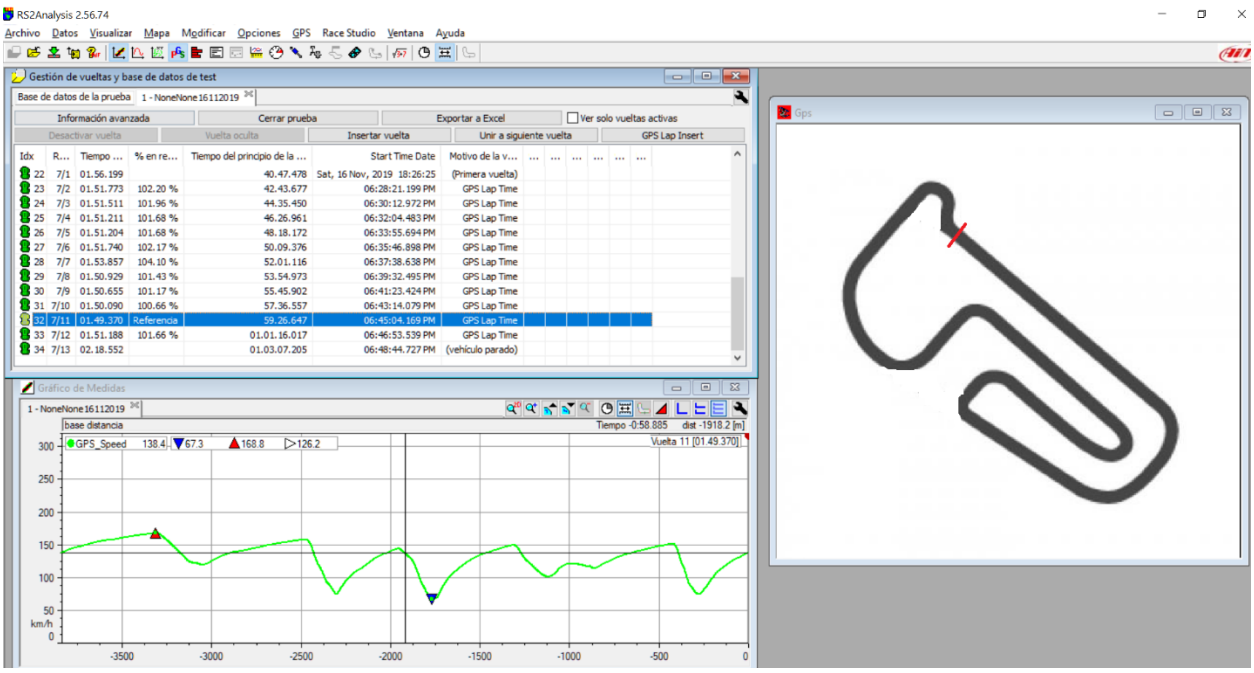
DELTA VELOCIDAD

COMPARACION ENTRE DOS VUELTAS

EN DEPORTES A MOTOR



Dispositivo AIM-Solo. Barre posición GPS y velocidad del dispositivo cada 0.1 segs.



Aplicación Race Analysis

Muy limitada y difícil de interpretar.

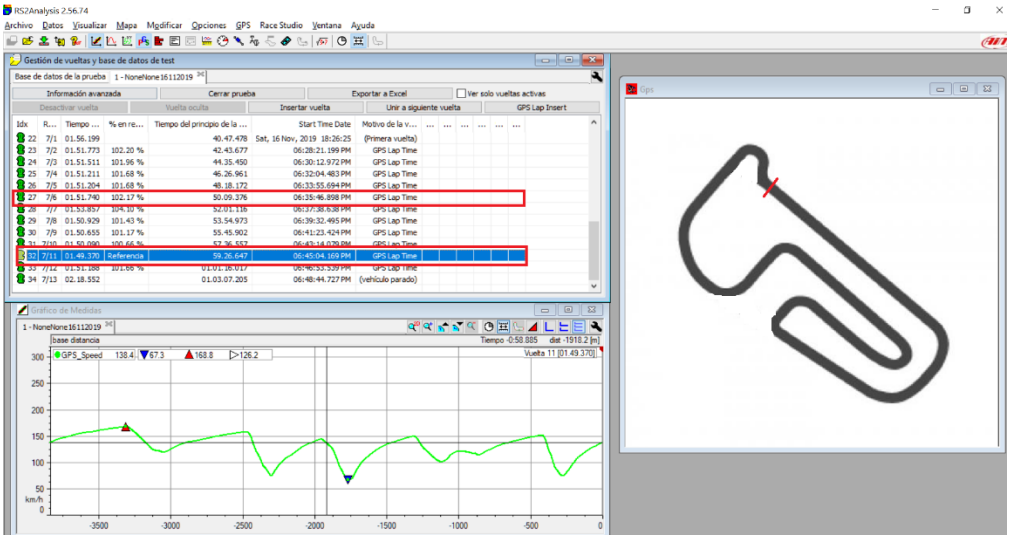
Especialmente para comparar 2 vueltas.

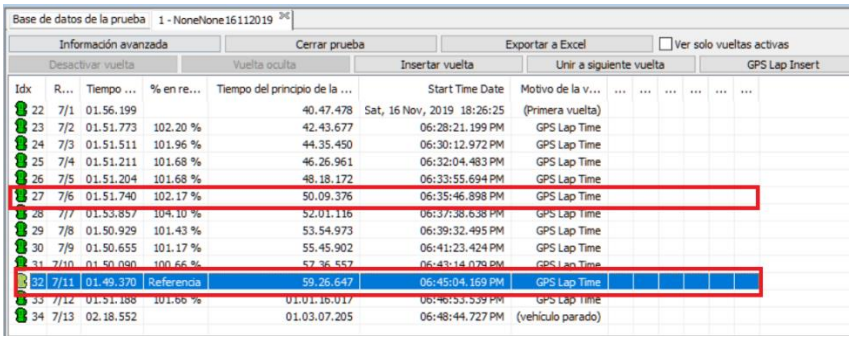
Pésima UX.

Pensada en 1990 y nunca actualizada.

PASO 1 – EXPORTACION DE DATOS

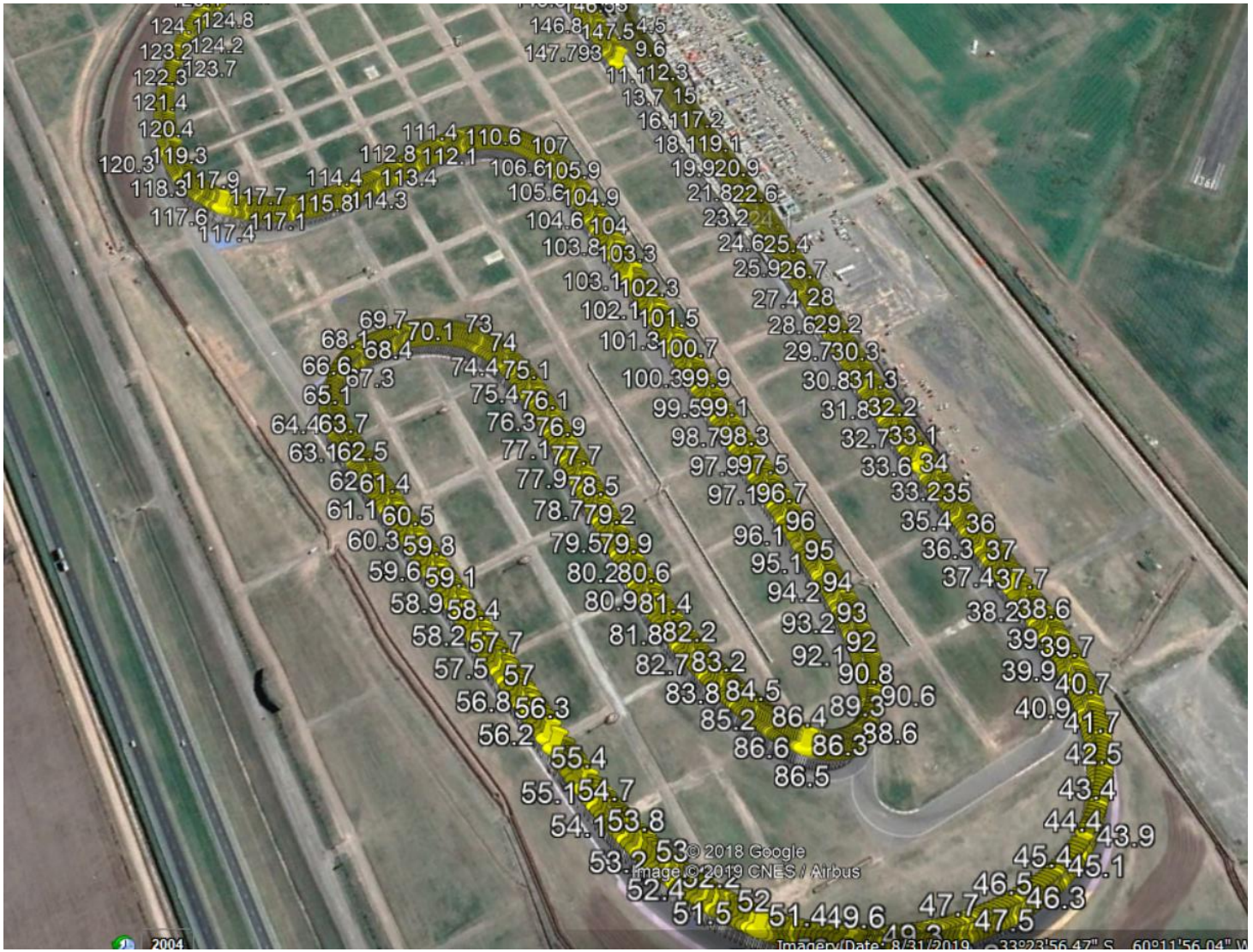
Circuito de San Nicolas, Provincia de Bs. As.



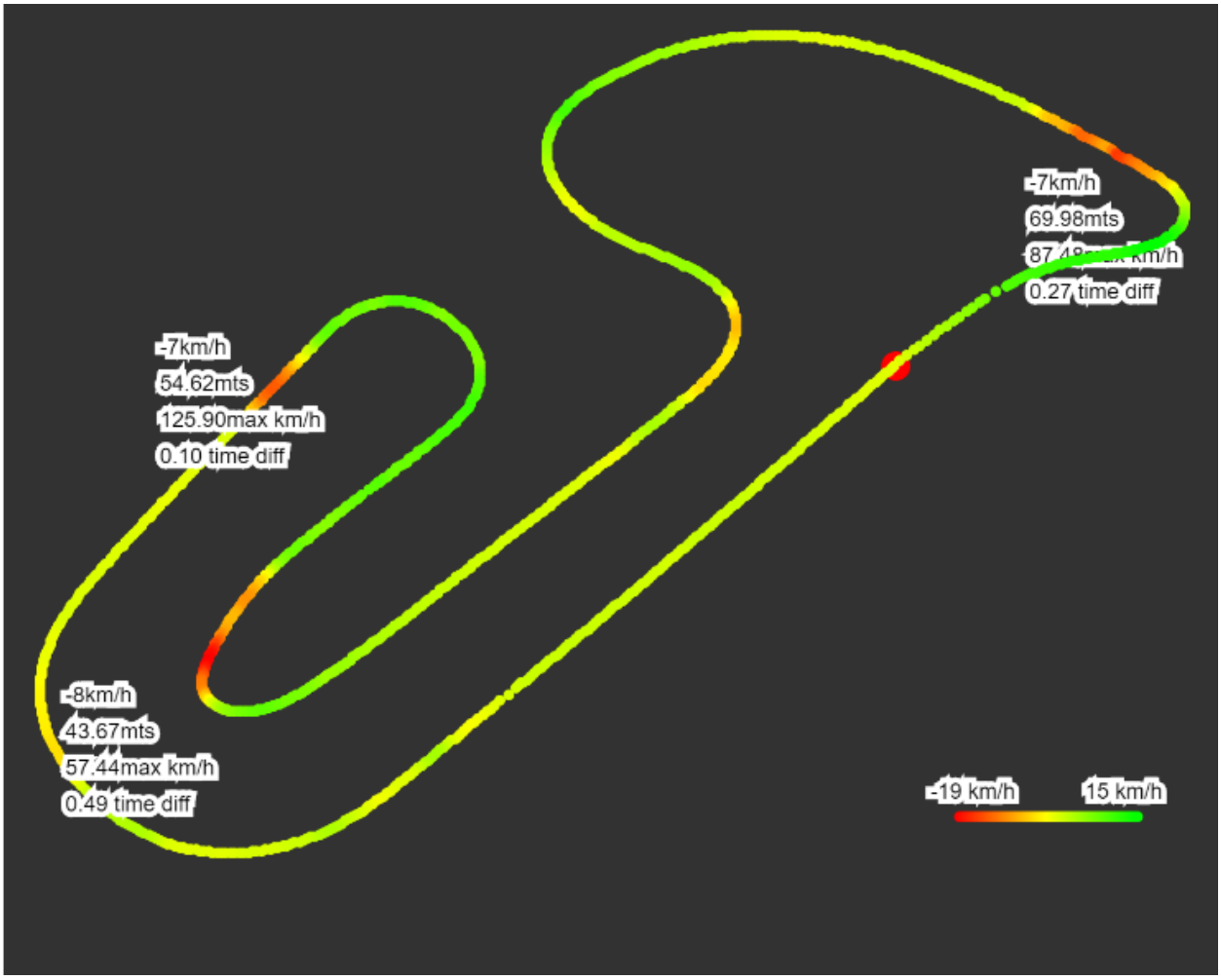


1	Format	AIM CSV File			
2	Venue	None			
3	Vehicle	None			
4	User	None			
5	Data Source	AIM Data Logger			
6	Comment				
7	Date	11/16/2019			
8	Time	18:26:25			
9	Sample Rate	10			
10	Duration	367.818			
11	Segment	Session			
12	Beacon Marke	147.793, 258.448, 367.818			
13	Segment Time	02:27.8	01:50.7	01:49.4	
14					
15	Time	Distance	GPS_Speed	GPS_Latitude	GPS_Longitude
16	Time	Distance	GPS_Speed	GPS_Latitude	GPS_Longitude
17	sec	km	km/h	°	°
18			1	2	3
19					
20	0	0	2.6	-33.395597	-60.201699
21	0.1	0	2.6255	-33.395598	-60.201699
22	0.2	0	2.651	-33.395599	-60.2017
23	0.3	0	2.6765	-33.3956	-60.201701
24	0.4	0	2.732	-33.3956	-60.201702
25	0.5	0	3.14	-33.395601	-60.201702
26	0.6	0	3.548	-33.395602	-60.201703

PASO 2 – VERIFICACION DE DATOS


© 2018 Google
Image © 2019 CNES / Airbus
2004
Imagery Date: 8/31/2019

PASO 3 – PROCESAMIENTO DE DATOS Y CREACION DEL HEAT MAP



CODIGO FUENTE

https://editor.p5js.org/enicolasgomez/sketches/M2YGUyIiy

EXPLICACION CODIGO FUENTE

Carga de dos vueltas (rápida, lenta)

```
function preload() {
  fast_lap = loadTable('16112019_2.csv', 'csv', 'header');
  lap = loadTable('16112019_1.csv', 'csv', 'header');
}
```

Creación de nuevo data set con diferencia de velocidad para cada coordenada

```
function create_dif_lap() {
  dif_lap = lap;
  for (let i = 1; i < 20; i++)
    dif_lap.removeRow(dif_lap.getRowCount() - i);
  for (let r = 0; r < lap.getRowCount(); r++) {
    let dif = fast_lap.getNum(r, "GPS_Speed") - lap.getNum(r, "GPS_Speed");
    dif_lap.set(r, 'GPS_Speed', dif);
  }
  print(dif_lap.getColumn('GPS_Speed'));
}
```

Calculo del color para un punto dado

```
function perc2color(perc, min, max) {
```

Creación de la barra de referencia

```
function reference_graph(max_speed, min_speed, x_loc, y_loc) {
  for (let x = 1; x < 100; x++) {
    let hex = perc2color(x, 1, 100);
    stroke(hex);
    point(x_loc + x, y_loc);
  }
  stroke("#FFFFFF")
  text(parseInt(min_speed) + " km/h", x_loc - 15, y_loc - 10);
  text(parseInt(max_speed) + " km/h", x_loc + 70, y_loc - 10);
}
```

Ploteo del grafico

```
function speed_graph(table, max_speed, min_speed, start_y) {
  let max_x = max(table.getColumn('GPS_Latitude'));
  let min_x = min(table.getColumn('GPS_Latitude'));

  let max_y = max(table.getColumn('GPS_Longitude'));
  let min_y = min(table.getColumn('GPS_Longitude'));

  for (let r = 0; r < table.getRowCount(); r++) {
    let xpos = table.getNum(r, "GPS_Latitude");
    let ypos = table.getNum(r, "GPS_Longitude");
    xpos = normalize(xpos, width * 0.9, min_x, max_x);
    ypos = normalize(ypos, graph_height * 0.9, min_y, max_y);
    let speed = table.getNum(r, "GPS_Speed");
    let speed_percent = ((speed - min_speed) / (max_speed - min_speed)) * 100;
    let hex = perc2color(speed_percent, 1, 97);
    stroke(hex);
    point(xpos + 20, start_y+ypos + 20);
  }
}
```