

Video de evidencia

Enlace: <https://www.youtube.com/watch?v=hatLaefBdhM>

Proceso de podado(pruning)

Trama Nmea-GPS L70 Bee:

```
14:10:29.770 -> $GPVTG,0.00,T,,M,0.13,N,0.25,K,A*38
14:10:29.805 -> $GPGGA,191031.091,0614.4211,N,07535.4838,W,1,4,2.45,1539.7,M,1.2,M,,*46
14:10:29.873 -> $GPGSA,A,3,02,29,13,06,,,,,,,,,2.62,2.45,0.92*01
14:10:29.943 -> $GPGSV,2,1,06,13,69,098,48,51,52,260,,02,37,047,48,29,25,329,41*73
14:10:30.009 -> $GPGSV,2,2,06,06,15,087,49,25,,,30*4B
14:10:30.043 -> $GPGLL,0614.4211,N,07535.4838,W,191031.091,A,A*4E
14:10:31.598 -> $GPRMC,191033.000,A,0614.4223,N,07535.4834,W,0.15,0.00,070820,,,A*77
```

El proceso de podado se desarrolla cuando del conjunto de datos tomados por los sensores, en este caso el GPS, se escoge solo lo que se necesita, por ejemplo, de la trama Nmea para esta práctica se requiere solo obtener los datos de la longitud, latitud, hora y fecha de toda la trama, pero la trama, además de esos datos tiene otros datos como número de satélites, dilución horizontal de la posición, elevación, azimut, SNR, etc.

En primera instancia solo se necesitaría los datos mencionados en la práctica, pero hay datos que no sirven en segundo plano para verificar si los datos son precisos o no, es decir, que se requieren otros datos adicionales para saber si los datos obtenidos son los más precisos.

La oración GPGGA contiene:

```
14:10:29.805 -> $GPGGA,191031.091,0614.4211,N,07535.4838,W,1,4,2.45,1539.7,M,1.2,M,,*46
```

Hora GTM Coordenadas Calidad de fijación
Número de satélites

La oración GPRMC contiene:

```
14:10:31.598 -> $GPRMC,191033.000,A,0614.4223,N,07535.4834,W,0.15,0.00,070820,,,A*77
```

Hora GTM Fecha

Teorema central del límite:

El teorema nos dice que si una muestra es lo bastante grande (generalmente cuando el tamaño muestral (n) supera los 20), sea cual sea la distribución de la media muestral, seguirá aproximadamente una distribución normal. Es decir, dada cualquier variable aleatoria, si extraemos muestras de tamaño n ($n > 20$) y calculamos los promedios muestrales, dichos promedios seguirán una distribución normal. Además, la media será la misma que la de la variable de interés.

```

160 void Temperatura()
161 {
162     temperatura = "";
163     int i = 1;
164     do
165     {
166         suma = suma + sensor.readTemperature();
167         i++;
168     }
169     while(i<21);
170     promedio=suma/20;
171     temperatura.concat(String(promedio)), temperatura.concat("C");Serial.println("Temperatura = "+temperatura);
172     promedio = 0;
173     suma=0;
174     i=1;
175 }
176
177 void Humedad()
178 {
179     humedad="";
180     int i = 1;
181     do
182     {
183         suma = suma + sensor.readHumidity();
184         i++;
185     }
186     while(i<21);
187     promedio=suma/20;
188     humedad.concat(String(promedio));humedad.concat('%');
189     Serial.println("Humedad = "+humedad);
190     promedio=0;
191     suma=0;
192     i=1;
193 }

```

En estas partes del código las funciones humedad y temperatura capturan 20 muestras de las variables correspondientes y se procede a relaizar el respectivo promedio por el teorema central del límite para obtener los datos con el menor porcentaje de error, esto también hace parte del proceso de pruning.

Tiempo de duración con una batería de 2600mAh:

Dispositivos	End device activo	End device dormido
Sensor	1.3uA = 1.3×10^{-3} mA	100nA = 100×10^{-6} mA
GPS	18mA	18mA
NodeMCU	170mA	10uA = 10×10^{-3} mA

$$\text{Consumo}_{\text{activo}} = \text{tiempo}_{\text{activo}} * (1.3 * 10^{-3} + 18 + 170) [mAh]$$

$$\text{Consumo}_{\text{dormido}} = \text{tiempo}_{\text{dormido}} * (100 * 10^{-6} + 18 + 10 * 10^{-3}) [mAh]$$

$$\frac{10_{\text{min}} * 1_{\text{horas}}}{60_{\text{min}}} = \frac{1}{6}h = \text{tiempo}_{\text{activo}} + \text{tiempo}_{\text{dormido}} [horas]$$

```

15:45:03.024 -> Iniciando...
15:45:03.873 -> Datos más precisos posibles
15:45:04.043 -> Temperatura = 29.50°C
15:45:04.214 -> Humedad = 42.24%
15:45:04.214 -> Posición: 6°14'25.47"N,75°35'27.67"W
15:45:04.248 -> Fecha: 7/8/2020
15:45:04.282 -> Hora: 20:45:05GTM-5
15:45:04.315 -> Trama creada: 29.50°C;42.24%;6°14'25.47"N,75°35'27.67"W;7/8/2020;20:45:05GTM-5
15:45:04.385 -> Tiempo ecendido: 1263.00
15:45:04.419 -> Modo sleep
15:45:04.419 -> -----Zlſſſlſſl`8ſ
15:54:57.070 -> Iniciando...
15:54:57.852 -> Datos más precisos posibles
15:54:58.021 -> Temperatura = 29.64°C
15:54:58.191 -> Humedad = 41.63%
15:54:58.191 -> Posición: 6°14'25.61"N,75°35'27.31"W
15:54:58.226 -> Fecha: 7/8/2020
15:54:58.259 -> Hora: 20:54:59GTM-5
15:54:58.292 -> Trama creada: 29.64°C;41.63%;6°14'25.61"N,75°35'27.31"W;7/8/2020;20:54:59GTM-5
15:54:58.359 -> Tiempo ecendido: 1216.00
15:54:58.393 -> Modo sleep
15:54:58.393 -> -----

```

$$tiempo_{activo\ promedio} = \frac{1263 + 1216}{2} = 1240\ ms$$

$$= \frac{1240_{segundos} * 10^{-3} * 1_{minutos} * 1_{horas}}{60_{segundos} * 60_{minutos}} = \frac{31}{90000} h$$

$$tiempo_{dormido} = \frac{1}{6} - \frac{31}{90000} = 0.16632h$$

$$consumo_{activo} = 6.48 * 10^{-2}mAh$$

$$consumo_{dormido} = 3mAh$$

$$Consumo_{total} = 3.0648mAh$$

$$Duración_{bateria} = \frac{2600}{3.0648} = 8.4834 * 10^2h = 35.3475\ dias$$

Gráfica del consumo de corriente de 1 hora por el sensor HDC1080

