

Informe de Avances

Moises Stevend Meza Rodriguez

Día 28/09/19

- Se tuvo una conversación con el desarrollador de la parte de hardware del Health-Recover Joaquín Verastegui.
- Se analizó los esquemáticos del hardware.
- Se comprobó el funcionamiento del hardware.
- Se puso operativo las tarjetas Health-Recover.
- Se analizó la arquitectura de comunicación del proyecto.

Día 30/09/19

- Se monitoreo el avance según el cronograma establecido.
- Se coordino el uso de la plataforma para programar los algoritmos.

Día 1/10/19

- Se probó la kinect con su sdk y el toolkit developer 1.8.
- Se empezó a probar enlazarlo con algún lenguaje de programación para obtener los puntos, se empezo con pykinect pero no se obtuvo los resultados esperados.

Día 2/10/19

- Se probó diferentes lenguajes para obtener los puntos, pero no se logró enlazar a ningún lenguaje.

Día 3/10/19

- Se probó un enlace mqtt entre los 2 esp8266,verificando que trabajen conjuntamente.

Día 4/10/19

- Se probó el software ROS en ubuntu 18, pero no se obtuvo avances ya que el software no contaba con compatibilidad con la kinect v1

Día 5/10/19

- Se tuvo una reunión para ver el seguimiento de los proyectos según el cronograma.
- Se instaló ubuntu12.04 para probar la versión libre de la kinect, no se obtuvo avances ya que no existe soporte para esa versión de ubuntu.

Día lunes 07/10/19

- Se instaló diferentes drivers para el reconocimiento del sdk-kinect modo developer, con el fin de poder programarla, se uso windows como sistema operativo.

- Se probó diferentes lenguajes de programación para obtener los puntos del esqueleto de una persona.
- Se probó C# por ser el lenguaje que mantiene compatibilidad con la Kinect v1. Este lenguaje nos podría servir si queremos algún punto en particular.
- Se instaló C# en Visual Studio y se procedió a programar un script de reconocimiento de trayectorias.

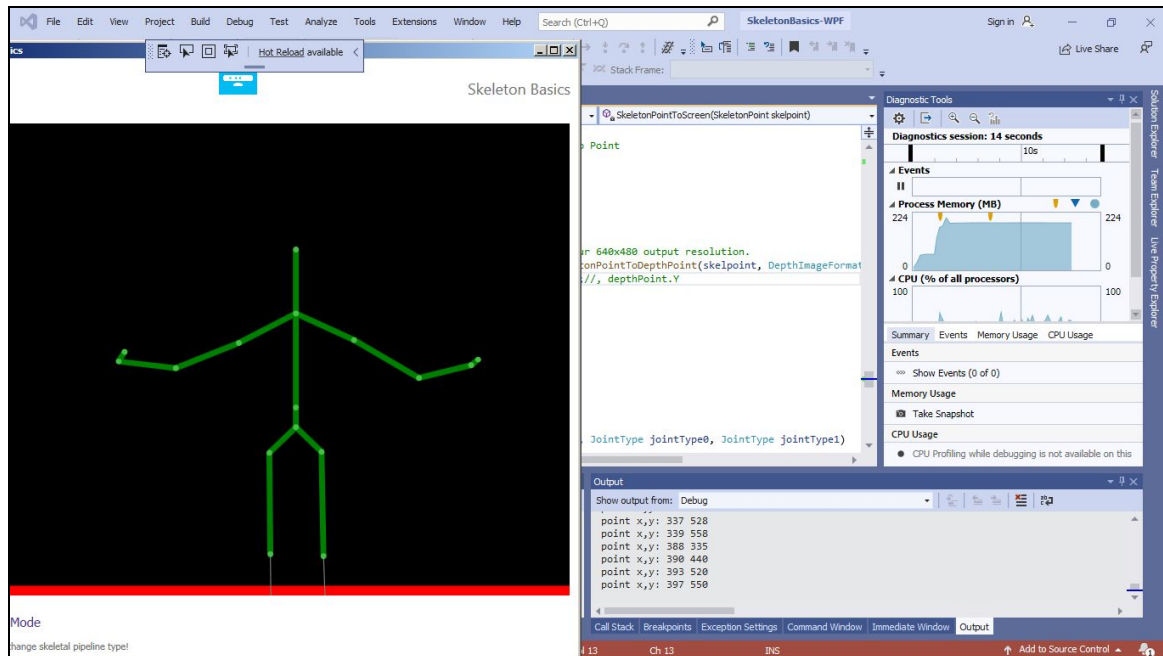


FIG.1- Esqueleto mediante líneas y puntos mediante C#

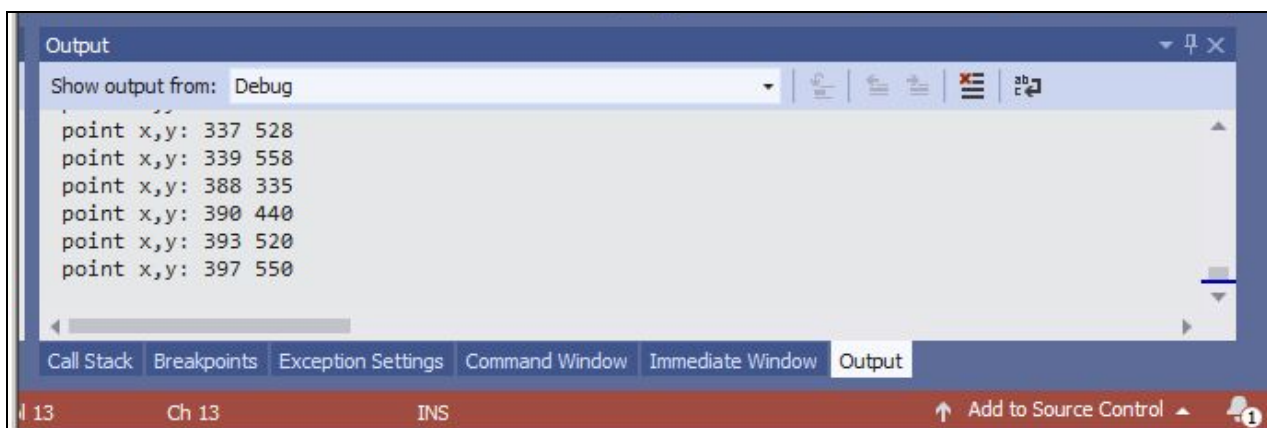


FIG.2- Puntos del esqueleto en C#

- Se instaló el software BREKEL como alternativa para obtener los puntos de manera rápida, se logró tener una visualización de los objetos y exportar los puntos de la parte

superior del cuerpo, la desventaja es que solo se puede hacer la toma de puntos del cuerpo por 4 segundos.

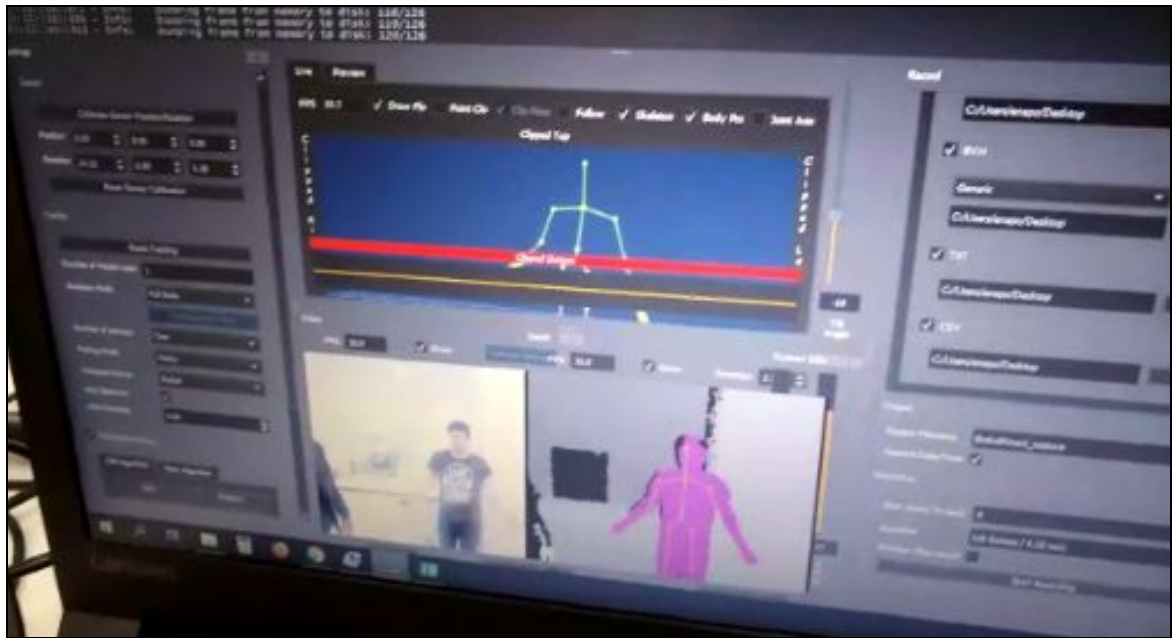


FIG.3- Entorno de trabajo de BREKEL

BrekelKinect_capture07_10_2019_15_28_25 - Excel (Producto sin licencia)																
frame																
frame	time	waist_conf	waist_tx	waist_ty	waist_tz	waist_rx	waist_ry	waist_rz	spine_conf	spine_rx	spine_ry	spine_rz	neck_conf	neck_rx	neck_ry	neck_rz
0	0	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	84.621513	-3.876397	13.5
1	0.033333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	84.621513	-3.876397	13.5
2	0.066667	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	84.621513	-3.876397	13.5
3	0.1	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	84.621513	-3.876397	13.5
4	0.133333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	84.621513	-3.876397	13.5
5	0.166667	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	84.326378	-3.366278	12.5
6	0.2	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	84.326378	-3.366278	12.5
7	0.233333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	84.018715	-2.88464	12.4
8	0.266667	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	83.637886	-2.396238	11.5
9	0.3	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	83.090569	-1.856264	10.5
10	0.333333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	82.313416	-1.203928	9.4
11	0.366667	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	81.705162	-0.793394	8.5
12	0.4	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	80.767357	-0.23107	7.2
13	0.433333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	79.933479	0.086915	5.4
14	0.466667	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	79.224602	0.196676	4.4
15	0.5	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	78.316612	0.356631	3.2
16	0.533333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	77.753174	0.131185	2.0
17	0.566667	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	77.065926	0.006049	0.4
18	0.6	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	76.388039	-0.045741	-0.2
19	0.633333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	75.814148	-0.180051	-1.2
20	0.666667	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	75.155602	-0.278246	-2.0
21	0.7	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	74.553093	-0.404904	-3.0
22	0.733333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	74.086411	-0.587667	-3.5
23	0.766667	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	73.576462	-0.715103	-4.5
24	0.8	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	72.655655	-1.022734	-6.0
25	0.833333	0.5	-49.051884	94.489738	-291.946899	155.679993	-0.000005	174.699997	0.5	128.606949	-8.079885	176.39798	1	72.655655	-1.022734	-6.0

FIG.4- Formato de puntos obtenido de una grabación de secuencial en BREKEL

Día miercoles 09/10/19

- Se probó el metasisistema operativo ROS(kinectic-ubuntu16 xenial) con la kinect v1, obteniendo los puntos a nivel de nodos de ROS (lo que nos permitiría enlazarlo con otros nodos-arduino, otra gran ventaja es poder obtener los puntos de manera ilimitada).

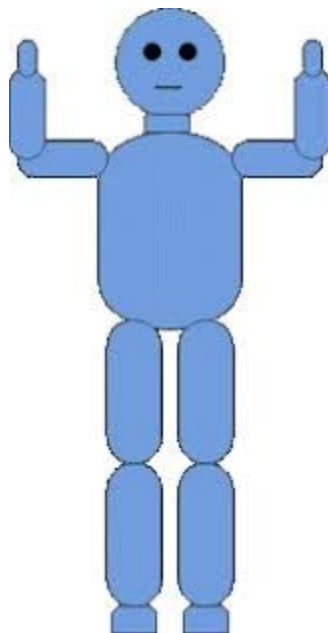
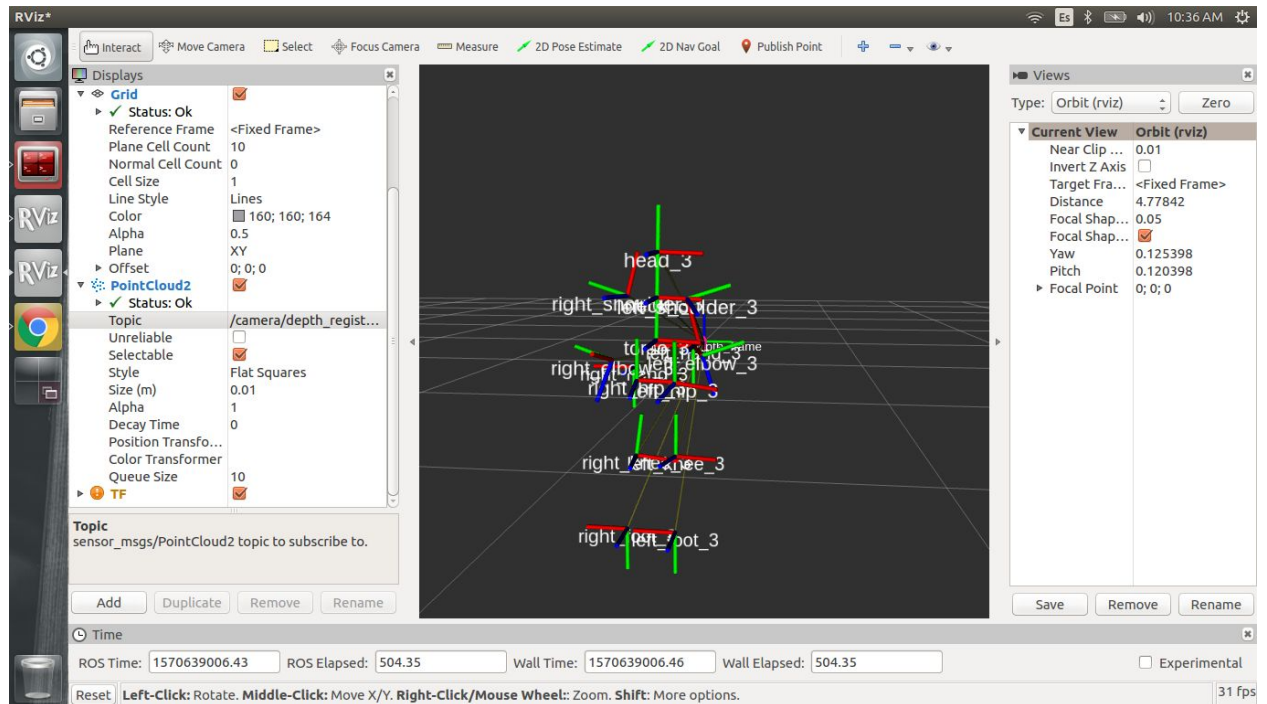
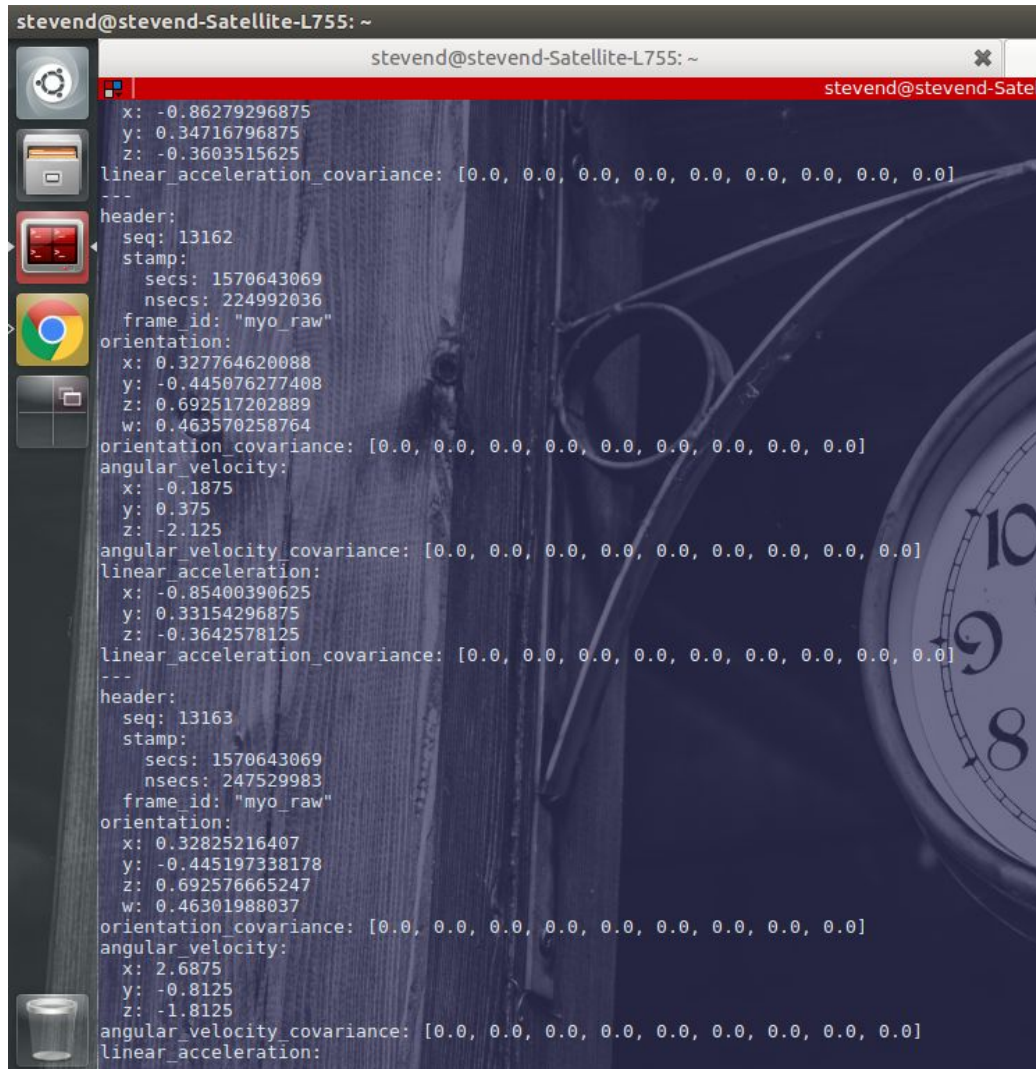


Fig .- Posición de inicio: “Pose PSI”

Se deja algunos enlaces usados:

- https://www.reddit.com/r/ROS/comments/6qeiy0/openni_kinect_installation_on_kinetic_i_ndigo/
 - https://answers.ros.org/question/10325/how-do-i-run-the-openni_tracker/
 - <http://dougspots.blogspot.com/2012/02/ros-kinect-skeleton-tracking.html>
 - <https://github.com/arnaud-ramey/NITE-Bin-Dev-Linux-v1.5.2.23>
 - <http://mitchtech.net/ubuntu-kinect-openni-primense/>
- Se probó el MYo-armband y se obtuvo los valores del IMU en ROS kinetic.



```
stevend@stevend-Satellite-L755: ~
x: -0.86279296875
y: 0.34716796875
z: -0.3603515625
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
---
header:
  seq: 13162
  stamp:
    secs: 1570643069
    nsecs: 224992036
  frame_id: "myo_raw"
orientation:
  x: 0.327764620088
  y: -0.445076277408
  z: 0.692517202889
  w: 0.463570258764
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: -0.1875
  y: 0.375
  z: -2.125
angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
  x: -0.85400390625
  y: 0.33154296875
  z: -0.3642578125
linear_acceleration_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
---
header:
  seq: 13163
  stamp:
    secs: 1570643069
    nsecs: 247529983
  frame_id: "myo_raw"
orientation:
  x: 0.32825216407
  y: -0.445197338178
  z: 0.692576665247
  w: 0.46301988037
orientation_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
angular_velocity:
  x: 2.6875
  y: -0.8125
  z: -1.8125
angular_velocity_covariance: [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
linear_acceleration:
```

Día viernes 11/10/19

- Taller de sistemas operativos y linux.
- Instalación de ubuntu en PCs

Día martes 15/10/19

- Adquisición de datos del mpu mediante mqtt con la laptop.
- Se tuvo una reunión sobre las modificaciones que tendrá las futuras versiones.

Día jueves 17/10/19

- Se instaló raspbian en la raspberry pi3
- Se hizo un mini taller sobre raspberry pi
- Se creó un script para adquirir datos del mpu6050

Datos raspberry pi3:

- ❖ **Red wifi:** LAB.ING.BIOMEDICA
- ❖ **Acceso via ssh:** ssh -X pi@healthrecoverpi
- ❖ **Username:** pi
- ❖ **Password:** MicroRobotica19

Día jueves 18/10/19

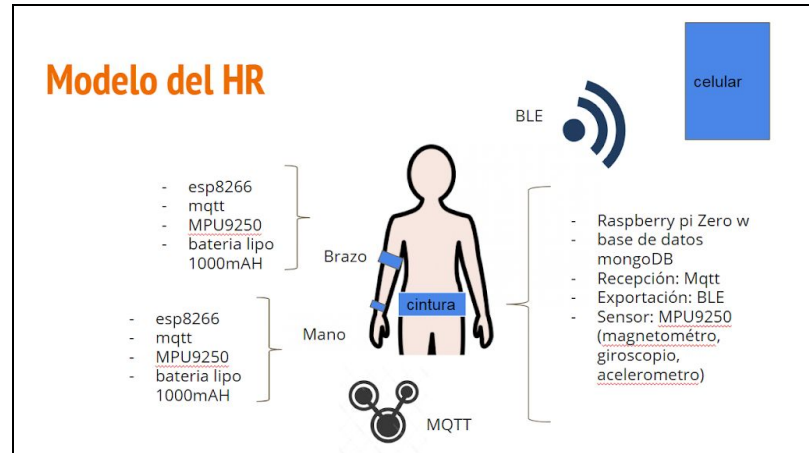
- Se hizo una pcb para la conexión entre mpu6050 y rpi3

Día jueves 21/10/19

- Se optó por rediseñar el dispositivo para un mejor control de trayectorias, basado en un esp32.
- Se diseñó el circuito electrónico del esp32 con el IMU-9 grados de libertad.

Día jueves 22/10/19

- Se diseñó la nueva arquitectura de comunicación de los dispositivos hardware y el smartphone.



Día jueves 25/10/19

- Pruebas de enlace entre el raspberry pi y los nodos del health recover.
- Se probó la instalación del ROS en el raspberry pi.

Día jueves 26/10/19

- Reunión semanal de avances del proyecto.

Día jueves 28/10/19

- Se analizó la identificación de patrones en WEKA
- Se analizó la Visualización de Datos en Python
- Se analizó la Base de Datos en MongoDB

Día jueves 31/10/19

- Se realizaron las compras para el nuevo dispositivo.

*En los días no mencionados se leyó artículos y tesis relacionados al proyecto.