INCISO 2
Configuración DH

```
91  void Robot::configurarTH(){
92      int es = 6;
93      /// ConfiguraciÃ³n Home del robot
94      x1= 0;       z1 = 102*es ;    theta1=radian(0) ;       alpha1=radian(-90);
95      x2= 50*es;   z2 = 0      ;    theta2=radian(0) ;       alpha2=radian(0)  ;
96      x3= 0;       z3 = 0      ;    theta3=radian(0) ;       alpha3=radian(-90);
97      x4= 0;       z4 = 48.5*es;    theta4=radian(0) ;       alpha4=radian(-90);
98      x5= 0;       z5 = 0      ;    theta5=radian(90);       alpha5=radian(-90);
99      x6= 0;       z6 = 18*es  ;    theta6=radian(0) ;       alpha6=radian(0)  ;
100     x7= 0;       z7 = 0      ;    theta7=radian(0) ;       alpha7=radian(0)  ;
101     /// Posiciona con las transformaciones al robot en HOME
102     // Base
103     DefinirTHz(0,vector3d(0,0,0));
104     THList.push_back(THz);
105     DefinirTHx(0,vector3d(0,0,0));
106     THList.push_back(THx);
107
108     //E11
109     DefinirTHz(theta1,vector3d(0,0,z1));
110     THList.push_back(THz);
111     DefinirTHx(alpha1,vector3d(x1,0,0));
112     THList.push_back(THx);
113
114     //E21
115     DefinirTHz(theta2,vector3d(0,0,z2));
116     THList.push_back(THz);
117     DefinirTHx(alpha2,vector3d(x2,0,0));
118     THList.push_back(THx);
119
120     //E31
121     DefinirTHz(theta3,vector3d(0,0,z3));
122     THList.push_back(THz);
123     DefinirTHx(alpha3,vector3d(x3,0,0));
124     THList.push_back(THx);
125
126
127     DefinirTHz(theta4,vector3d(0,0,z4));
128     THList.push_back(THz);
129     DefinirTHx(alpha4,vector3d(x4,0,0));
130     THList.push_back(THx);
131
132     DefinirTHz(theta5,vector3d(0,0,z5));
133     THList.push_back(THz);
134     DefinirTHx(alpha5,vector3d(x5,0,0));
135     THList.push_back(THx);
136
137     DefinirTHz(theta6,vector3d(0,0,z6));
138     THList.push_back(THz);
139     DefinirTHx(alpha6,vector3d(x6,0,0));
140     THList.push_back(THx);
141
142
143     /// Muestra la multiplicaciÃ³n de las transformaciones 3 eslabones
144     Matrix T(4,4);
145     T.identity(4);
146
147     cout << "============================" << endl;
148     cout << "||        Matriz T01      ||" << endl;
149     cout << "============================" << endl;
150     T = THList[2]*THList[3];
151     T.mostrar();
152
153     T.identity(4);
154     cout << "============================" << endl;
155     cout << "||        Matriz T12      ||" << endl;
156     cout << "============================" << endl;
157     T = THList[4]*THList[5];
158     T.mostrar();
159
```

```
160       T.identity(4);
161       cout << "============================" << endl;
162       cout << "||         Matriz T23      ||" << endl;
163       cout << "============================" << endl;
164       T = THList[6]*THList[7];
165       T.mostrar();
166
167       T.identity(4);
168       cout << "============================" << endl;
169       cout << "||         Matriz T34      ||" << endl;
170       cout << "============================" << endl;
171       T = THList[8]*THList[9];
172       T.mostrar();
173
174
175       T.identity(4);
176       cout << "============================" << endl;
177       cout << "||         Matriz T45      ||" << endl;
178       cout << "============================" << endl;
179       T = THList[10]*THList[11];
180       T.mostrar();
181
182
183       T.identity(4);
184       cout << "============================" << endl;
185       cout << "||         Matriz T56      ||" << endl;
186       cout << "============================" << endl;
187       T = THList[12]*THList[13];
188       T.mostrar();
189
190       T.identity(4);
191       for( int m = 0; m < 7; m++ ) {
192           T = T* THList[2*m+0]*THList[2*m+1];
193       }
194       cout << "============================" << endl;
195       cout << "||         Matriz T06      ||" << endl;
196       cout << "============================" << endl;
197       T.mostrar();
198
199       InicializarCinematicaInversa();
200   }
```

```
============================
||        Matriz T01      ||
============================


1  0  0  0

0  -4.37114e-008  1  0

0  -1  -4.37114e-008  612

0  0  0  1

============================
||        Matriz T12      ||
============================


1  0  0  300

0  1  0  0

0  0  1  0

0  0  0  1

============================
||        Matriz T23      ||
============================


1  0  0  0

0  -4.37114e-008  1  0

0  -1  -4.37114e-008  0

0  0  0  1
```

```
============================
||        Matriz T34      ||
============================


1  0  0  0

0  -4.37114e-008  1  0

0  -1  -4.37114e-008  291

0  0  0  1

============================
||        Matriz T45      ||
============================


-4.37114e-008  4.37114e-008  -1  0

1  1.91069e-015  -4.37114e-008  0

0  -1  -4.37114e-008  0

0  0  0  1

============================
||        Matriz T56      ||
============================


1  0  0  0

0  1  0  0

0  0  1  108

0  0  0  1
```

```
============================
||        Matriz T06      ||
============================


-4.37114e-008  4.37114e-008  -1  192

1.31134e-007  1  4.37114e-008  -2.07192e-005

1  -1.31134e-007  -4.37114e-008  321

0  0  0  1

============================
||       Posicion rO6     ||
============================


192

-2.07192e-005

321

1

============================
||   Angulos sin C.I.     ||
============================
theta1 : 0
theta2 : 0
theta3 : 0
theta4 : 0
theta5 : 1.5708
theta6 : 0
```

CONTROL MANUAL

>>> Main.cpp

```
case '1':
    Miclase->SSRMS.theta1=Miclase->SSRMS.theta1+dtheta;
    Miclase->SSRMS.DefinirTHz(Miclase->SSRMS.theta1, {0,0, Miclase->SSRMS.z1});        //Eslabon 1
    Miclase->SSRMS.THList[2]=Miclase->SSRMS.THz;
    Miclase->SSRMS.InicializarCinematicaInversa();
    Miclase->SSRMS.t = 0;
    break;

case '2':
    Miclase->SSRMS.theta1=Miclase->SSRMS.theta1-dtheta;
    Miclase->SSRMS.DefinirTHz( Miclase->SSRMS.theta1, {0,0,Miclase->SSRMS.z1});        //Eslabon 1
    Miclase->SSRMS.THList[2]=Miclase->SSRMS.THz;
    Miclase->SSRMS.InicializarCinematicaInversa();
    Miclase->SSRMS.t = 0;
    break;
```

Para la implementación del control manual, se la asigna una letra a cada rotación, positiva o negativa; con la cual efectuará la actualización de $Theta_i$.

La variable $Theta_i$, para i = 1, 2,3,…, 6. Se le agrega un diferencial de theta (equivalente a una rotación del eslabón en esa articulación). Posteriormente se actualiza el ángulo THz correspondiente y se vuelve a apilar el modelo en la pila de matrices THz:

```
Miclase->SSRMS.THList[N]=Miclase->SSRMS.THz;
```

Siendo N = 2*i
Lo que Vuelve a cargar la posición.

Y por último se llama a inicializar cinemática que resetea los puntos a graficar y calcula de nuevo la posición rO6.
De igual manera se resetea t=0; para poder empezar la paramétrica desde un valor de t neutro.

Después se renderiza el grafico.

INCISO 4
CINEMATICA Inversa 2 Método analítico

```
292  bool Robot::CinematicaInversa2(){
293      ///Calcular O4 es necesario
294      Matrix rO4(4,1);
295      rO4=rO6-z6*k6;
296
297      xO4=rO4.entry(0,0);
298      yO4=rO4.entry(1,0);
299      zO4=rO4.entry(2,0);
300
301      bool state = CinematicaInversa1();
302
303      if(state==false) return false;
304
305      Parametrica2();
306      curva.push_back(vector3d(xO6,yO6,zO6));
307
308      Matrix T03(4,4);
309      T03.resetIdentity();
310
311      modelo3D *model;
312
313      for( int m = 0; m < 4; m++ ) {
314          model = modelos[m];
315          T03 = T03* THList[2*m+0]*THList[2*m+1];
316      }
317
318      Matrix A(T03.inversa()*T06);
319      float t41, t42, t43, q51, q52, q53, t61, t62, t63;
320      float axuTheta4 = theta4;
321      axuTheta4 = theta4;
322
323
324
325      t41=atan2(A.entry(1,2),A.entry(0,2));
326      t42=atan2(A.entry(1,2),A.entry(0,2))+PI;
327      t43=atan2(A.entry(1,2),A.entry(0,2))+2*PI;
328      theta4 = t41; //menor(t41,t42-PI,t43,theta4);
329
330
331
332      t61=atan2(A.entry(2,1),-1*A.entry(2,0));
333      t62=atan2(A.entry(2,1),-1*A.entry(2,0))+PI;
334      t63=atan2(A.entry(2,1),-1*A.entry(2,0))+2*PI;
335      theta6 = menor(t61,t62,t63,theta6);
```

```
336
337       q5=theta5-radian(90);
338
339       q51=atan2( -1*sin(theta4)*A.entry(2,2),A.entry(1,2));
340       q52=atan2( -1*sin(theta4)*A.entry(2,2),A.entry(1,2))+PI;
341       q53=atan2( -1*sin(theta4)*A.entry(2,2),A.entry(1,2))+2*PI;
342       q5 = menor(q51,q52,q53,q5);
343
344       /*
345           theta4 = atan2(A.entry(1,3),A.entry(0,3);
346           theta4 = atan2(A.entry(1,2),A.entry(0,2));
347           theta6 = atan2(A.entry(2,1),-1*A.entry(2,0));
348           q5 = atan2( -1*sin(theta4)*A.entry(2,2),A.entry(1,2));/// Checar con mauri->>>
349
350       */
351
352       theta5 = q5+radian(90);
353       cout << "theta4 : " << theta4 << endl;
354       cout << "theta5 : " << theta5 << endl;
355       cout << "theta6 : " << theta6 << endl;
356       cout << "-------------------------------------------------------------------------" << endl;
357       DefinirTHz(theta4,{0,0,z4});
358       THList[8]=THz;
359       DefinirTHz(theta5,{0,0,z5});
360       THList[10]=THz;
361       DefinirTHz(theta6,{0,0,z6});
362       THList[12]=THz;
363
364       return true;
365   }
```

$$
{}^{3}_{4}\mathrm{T} = \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & z_4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & x_4 \\ 0 & \cos(\alpha_4) & -\sin(\alpha_4) & 0 \\ 0 & \sin(\alpha_1) & \cos(\alpha_4) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta_4 & 0 & -\sin\theta_4 & 0 \\ \sin\theta_4 & 0 & \cos\theta_4 & 0 \\ 0 & -1 & 0 & z_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$${}^{4}_{5}T=\begin{bmatrix}\cos(\theta_5+q_5) & -\sin(\theta_5+q_5) & 0 & 0\\ \sin(\theta_5+q_5) & \cos(\theta_5+q_5) & 0 & 0\\ 0 & 0 & 1 & z_5\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & x_5\\ 0 & \cos(\alpha_5) & -\sin(\alpha_5) & 0\\ 0 & \sin(\alpha_5) & \cos(\alpha_5) & 0\\ 0 & 0 & 0 & 1\end{bmatrix}:\left|\begin{bmatrix}-\sin q_5 & 0 & -\cos q_5 & 0\\ \cos q_5 & 0 & -\sin q_5 & 0\\ 0 & -1 & 0 & z_5\\ 0 & 0 & 0 & 1\end{bmatrix}\right.$$

$${}^{5}_{6}T=\begin{bmatrix}\cos(\theta_6) & -\sin(\theta_6) & 0 & 0\\ \sin(\theta_6) & \cos(\theta_6) & 0 & 0\\ 0 & 0 & 1 & z_6\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 0 & x_6\\ 0 & \cos(\alpha_6) & -\sin(\alpha_6) & 0\\ 0 & \sin(\alpha_6) & \cos(\alpha_6) & 0\\ 0 & 0 & 0 & 1\end{bmatrix}=\begin{bmatrix}\cos\theta_6 & -\sin\theta_6 & 0 & 0\\ \sin\theta_6 & \cos\theta_6 & 0 & 0\\ 0 & 0 & 1 & z_6\\ 0 & 0 & 0 & 1\end{bmatrix}$$

$${}^{3}_{6}T=\begin{bmatrix}\cos\theta_4 & 0 & -\sin\theta_4 & 0\\ \sin\theta_4 & 0 & \cos\theta_4 & 0\\ 0 & -1 & 0 & z_4\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}-\sin q_5 & 0 & -\cos q_5 & 0\\ \cos q_5 & 0 & -\sin q_5 & 0\\ 0 & -1 & 0 & z_5\\ 0 & 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos\theta_6 & -\sin\theta_6 & 0 & 0\\ \sin\theta_6 & \cos\theta_6 & 0 & 0\\ 0 & 0 & 1 & z_6\\ 0 & 0 & 0 & 1\end{bmatrix}:$$

$$\begin{bmatrix}\sin\theta_4\sin\theta_6-\cos\theta_4\cos\theta_6\sin q_5 & \cos\theta_6\sin\theta_4+\cos\theta_4\sin\theta_6\sin q_5 & -\cos\theta_4\cos q_5 & -z_5\sin\theta_4-z_6\cos\theta_4\cos q_5\\ -\cos\theta_4\sin\theta_6-\cos\theta_6\sin\theta_4\sin q_5 & \sin\theta_4\sin\theta_6\sin q_5-\cos\theta_4\cos\theta_6 & -\sin\theta_4\cos q_5 & z_5\cos\theta_4-z_6\sin\theta_4\cos q_5\\ -\cos\theta_6\cos q_5 & \sin\theta_6\cos q_5 & \sin q_5 & z_4+z_6\sin q_5\\ 0 & 0 & 0 & 1\end{bmatrix}$$

Para resolver analiticamente se considera

si ${}^{3}_{6}T=({}^{0}_{3}T)\ {}^{1}\ {}^{0}_{6}T$

$$A=\begin{bmatrix}a00 & a01 & a02 & a03\\ a10 & a11 & a21 & a11\\ a20 & a21 & a22 & a11\\ a30 & a31 & a32 & a33\end{bmatrix}=({}^{0}_{3}T)\ {}^{1}\ {}^{0}_{6}T$$

$$\begin{bmatrix} \sin\theta_4 \sin\theta_6 - \cos\theta_4 \cos\theta_6 \sin q_5 & \cos\theta_6 \sin\theta_4 + \cos\theta_4 \sin\theta_6 \sin q_5 & -\cos\theta_4 \cos q_5 & -z_5 \sin\theta_4 - z_6 \cos\theta_4 \cos q_5 \\ -\cos\theta_4 \sin\theta_6 - \cos\theta_6 \sin\theta_4 \sin q_5 & \sin\theta_4 \sin\theta_6 \sin q_5 - \cos\theta_4 \cos\theta_6 & -\sin\theta_4 \cos q_5 & z_5 \cos\theta_4 - z_6 \sin\theta_4 \cos q_5 \\ -\cos\theta_6 \cos q_5 & \sin\theta_6 \cos q_5 & \sin q_5 & z_4 + z_6 \sin q_5 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} a00 & a01 & a02 & a03 \\ a10 & a11 & a21 & a11 \\ a20 & a21 & a22 & a11 \\ a30 & a31 & a32 & a33 \end{bmatrix}$$

De donde se despejan

$$\theta_4 = \arctan\left(\frac{a12}{a02}\right)$$

$$\theta_6 = \arctan\left(\frac{a21}{-1*a20}\right)$$

$$q_5 = \arctan\left(\frac{-1*\sin(\theta_4)*a22}{a12}\right)$$

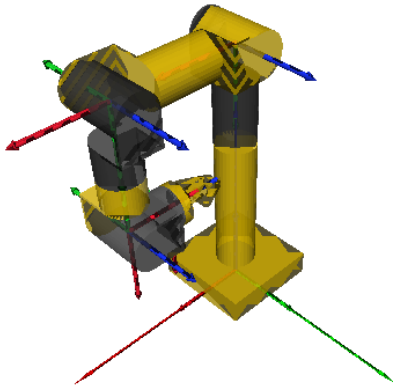
CINEMATICA Inversa 1 Método Número

```
628  Matrix Robot::Jacobiano(float x, float y, float z){
629      Matrix J(3,3);
630      J.entry(0,0)=(z4*cos(y)*sin(x)*sin(z))-1*(x2*cos(y)*sin(x))+(z4*cos(z)*sin(x)*sin(y));
631      J.entry(1,0)=(x2*cos(x)*cos(y))-1*(z4*cos(x)*cos(y)*sin(z))-1*(z4*cos(x)*cos(z)*sin(y));
632      J.entry(2,0)=0;
633
634      J.entry(0,1)=(z4*cos(x)*sin(y)*sin(z))-1*(z4*cos(x)*cos(y)*cos(z))-1*(x2*cos(x)*sin(y));
635      J.entry(1,1)=(z4*sin(x)*sin(y)*sin(z))-1*(z4*cos(y)*cos(z)*sin(x))-1*(x2*sin(x)*sin(y));
636      J.entry(2,1)=(z4*cos(y)*sin(z))-1*(x2*cos(y))+(z4*cos(z)*sin(y));
637
638      J.entry(0,2)=(z4*cos(x)*sin(y)*sin(z))-1*(z4*cos(x)*cos(y)*cos(z));
639      J.entry(1,2)=(z4*sin(x)*sin(y)*sin(z))-1*(z4*cos(y)*cos(z)*sin(x));
640      J.entry(2,2)=(z4*cos(y)*sin(z))+(z4*cos(z)*sin(y));
641      return J;
642  }
643  Matrix Robot::F(float x, float y, float z){
644      Matrix Fs(3,1);
645      float c=xO4, d=yO4, e=zO4;
646      Fs.entry(0,0) =x2*cos(x)*cos(y)-1*z4*cos(x)*cos(y)*sin(z)-1*z4*cos(x)*cos(z)*sin(y)-c;
647      Fs.entry(1,0) =x2*cos(y)*sin(x)-1*z4*cos(y)*sin(x)*sin(z)-1*z4*cos(z)*sin(x)*sin(y)-d;
```

```
    Fs.entry(2,0) =z1-1*x2*sin(y)-1*z4*cos(y)*cos(z)+z4*sin(y)*sin(z)-e;

    return Fs;
}
bool Robot::NewtonRapshon(float &x, float &y, float  &z){

    int i = 0;
    Matrix qs(3,1),dqs(3,1);
    qs.entry(0,0)=x;
    qs.entry(1,0)=y;
    qs.entry(2,0)=z;

    while( fabs(F(x,y,z).magnitud()) > 0.0001 ){
        dqs = (-1*Jacobiano(x,y,z).inversa())*F(x,y,z);
        qs = qs+dqs;
        x = qs.entry(0,0); y = qs.entry(1,0); z = qs.entry(2,0);
        if(i==1000){return false;}
        i++;
    }
    return true;
}
```

INCISO 5

```
321

1

============================
||       Posicion rO6     ||
============================


192

-1.38533e-005

321

1

============================
||   Angulos sin C.I.     ||
============================
theta1 : 0
theta2 : 0
theta3 : 0
theta4 : -3.14159
theta5 : 4.71239
theta6 : 4.37114e-008
------------------------------------------------
```



```
==============================
||        Posicion rO6      ||
==============================


349.5

157.5

478.5

1

==============================
||    Angulos sin C.I.      ||
==============================
theta1 : 0.331554
theta2 : -0.277547
theta3 : -0.458341
theta4 : -2.70679
theta5 : 5.39999
theta6 : 0.286659
---------------------------------------
```