



**URJC**



Escuela de Ingeniería  
de Fuenlabrada



**RoboticsLabURJC**  
Programming Robot Intelligence

# **Modelado y Simulación de Robots**

## **URDF: Unified Robot Description Format**

---

Grado en Ingeniería de Robótica Software

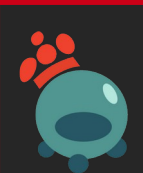
Teoría de la Señal y las Comunicaciones y  
Sistemas Telemáticos y Computación

Roberto Calvo Palomino  
[roberto.calvo@urjc.es](mailto:roberto.calvo@urjc.es)

# URDF

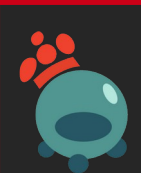
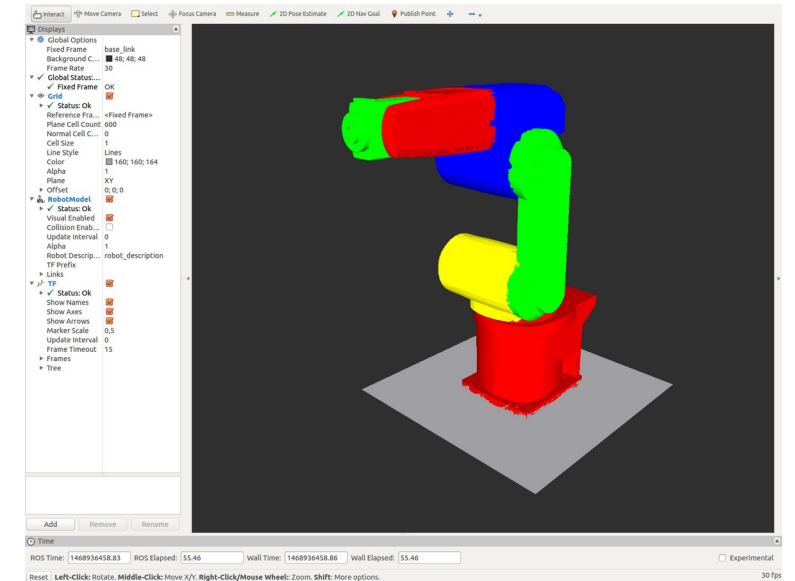
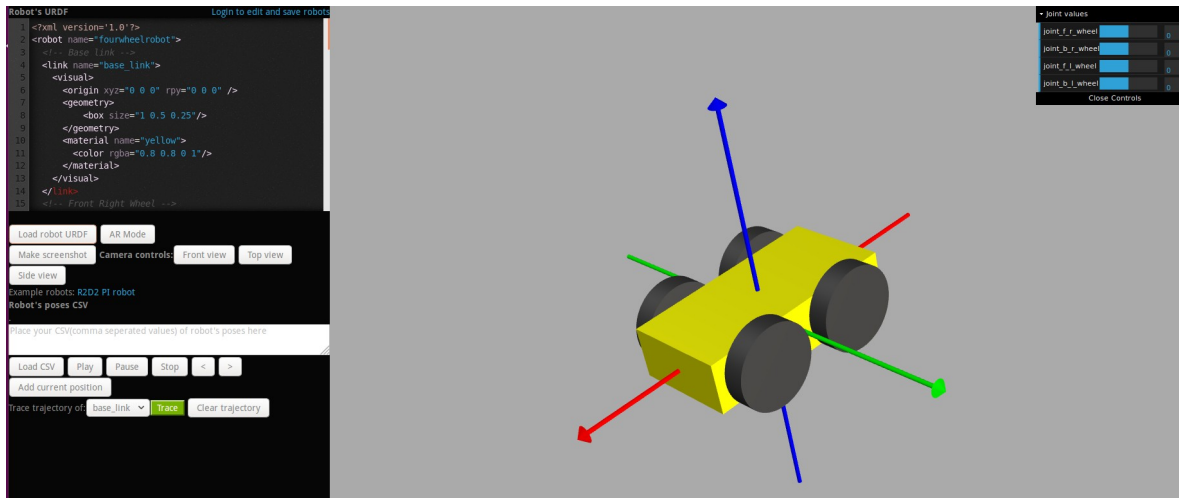
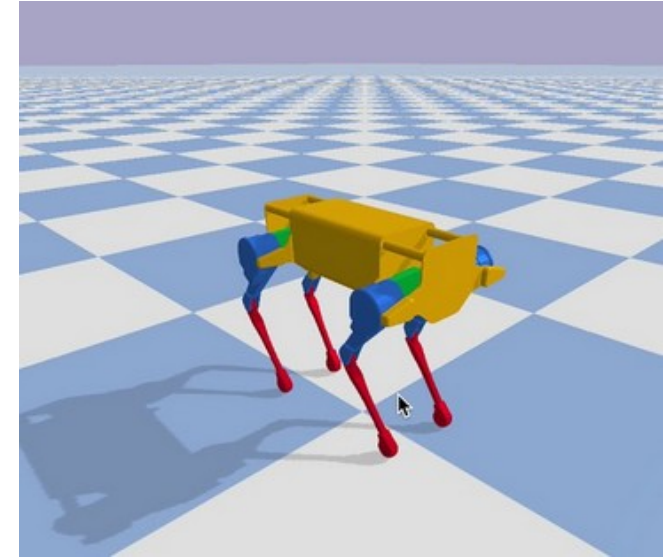
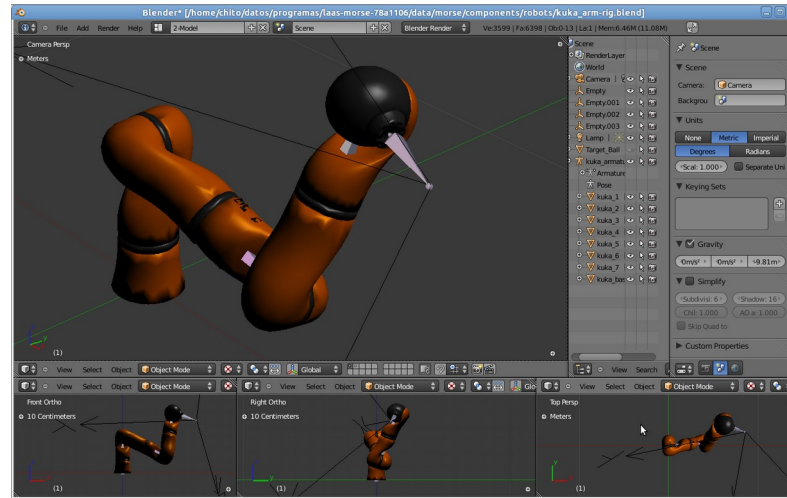
- **URDF** es una especificación XML para describir un modelo de un robot. Podemos representar:
  - Descripción cinemática
  - Descripción física
  - Representación visual
  - Modelo de colisión
- Formato mayormente usado para la creación de modelos de robots (de hecho un estándar en ROS).

<https://mymodelrobot.appspot.com/>



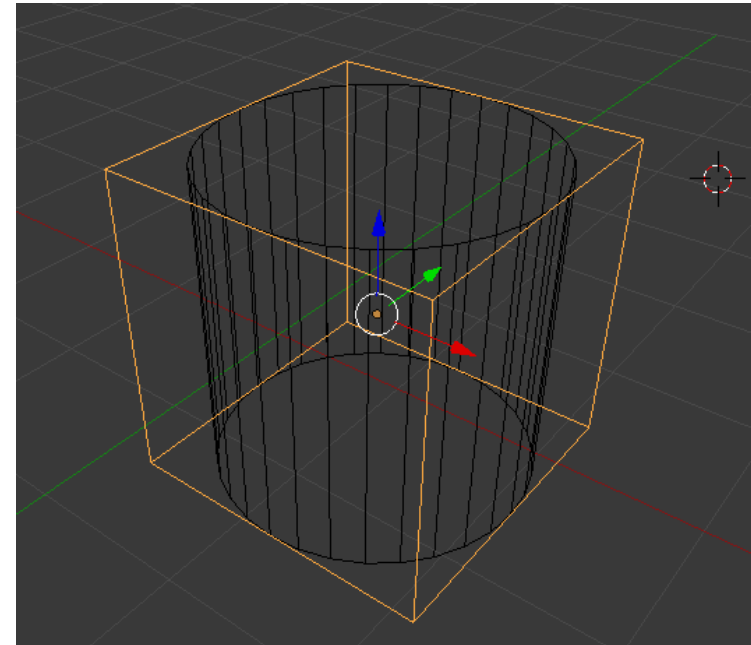
# URDF

- Visores
  - Rviz
  - Blender
  - Pybullet
  - mymodelrobot

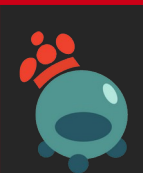


# URDF - XML

- Dentro del modelo visual y de colisión podemos definir una geometría específica.
  - Caja (Box)
  - Cilindro (cylinder)
  - Esfera (sphere)
  - Malla (mesh)

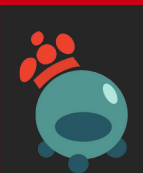
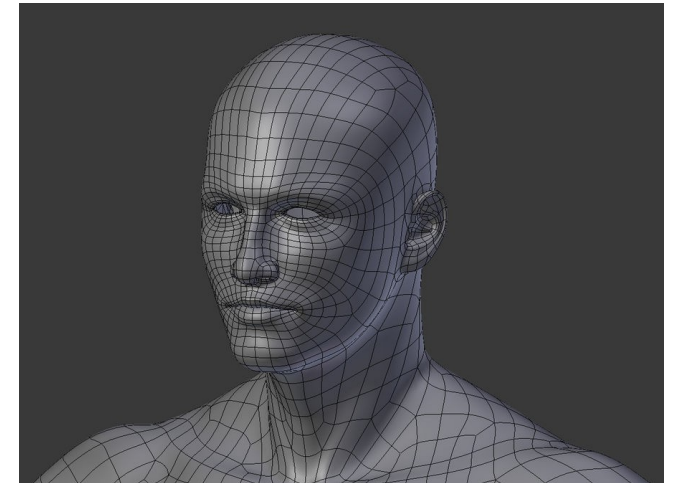
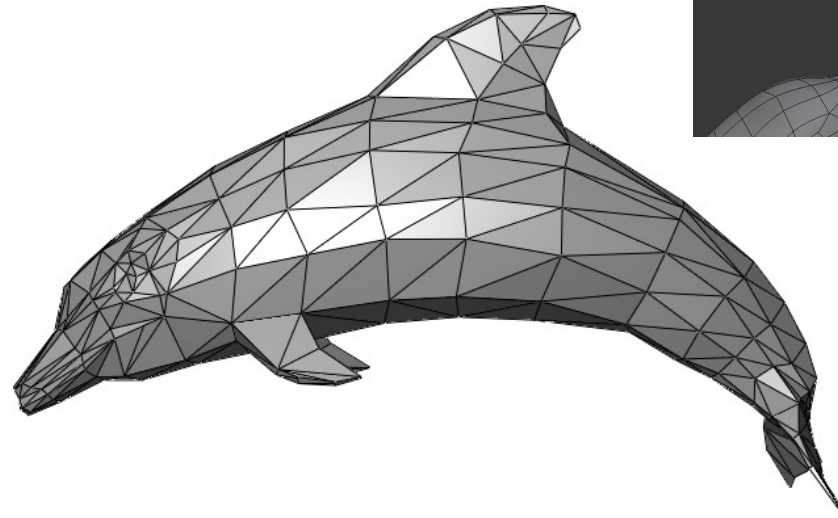
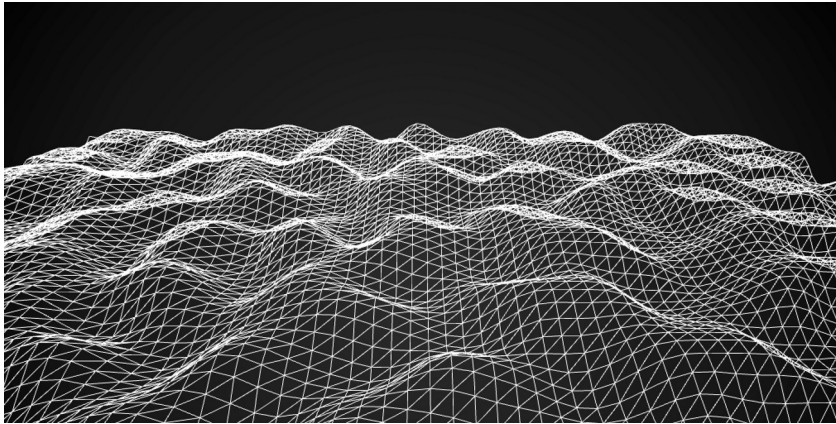


- Los 3 primeros son más eficientes computacionalmente y las mallas son más precisas para el modelo.



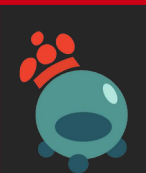
# URDF - XML

- Malla o *mesh* es una superficie creada mediante un método tridimensional generado por sistemas de vértices posicionados en un espacio virtual



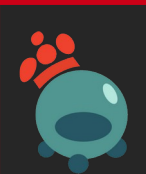
# URDF - XML

- El elemento **<robot>**
- Obligatorio en el fichero (nodo padre)
- Descripción completa del robot u objeto
- Elementos:
  - **<link>**
    - Descripción de un cuerpo rígido (enlaceS)
  - **<joint>**
    - Articulaciones entre links
  - **<gazebo>**
    - Añade extensiones de simulación para Gazebo



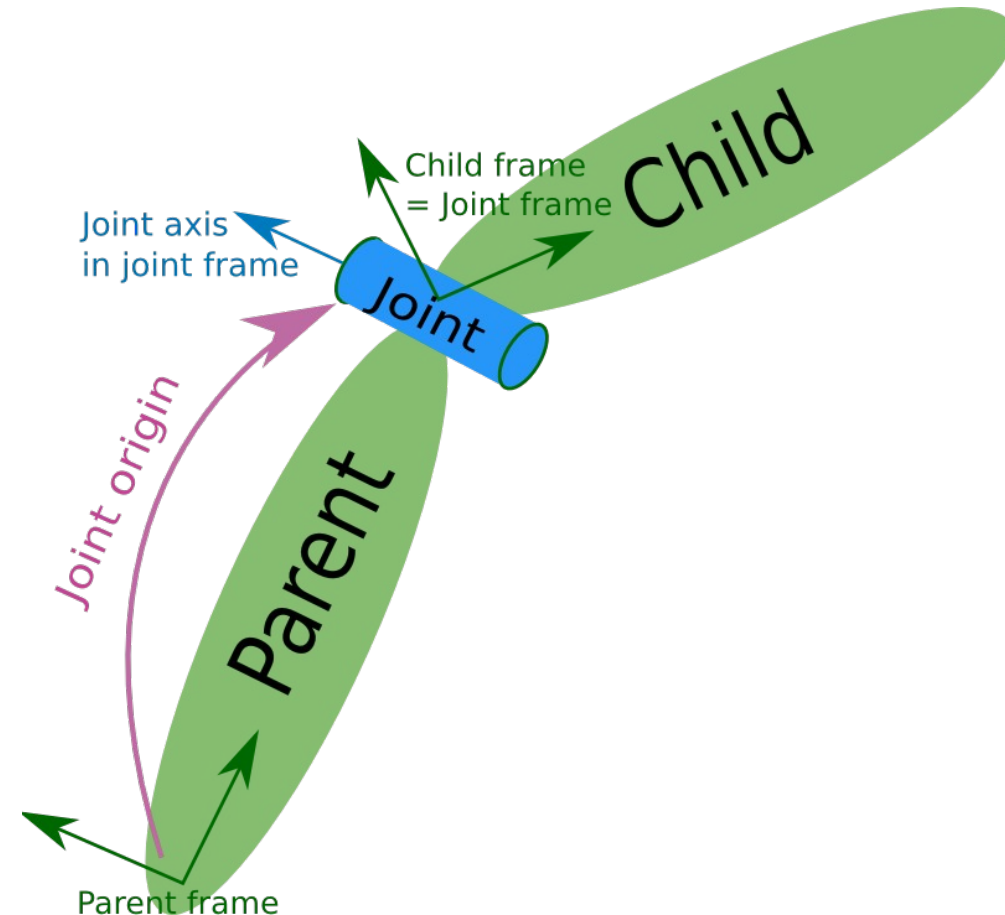
# URDF - XML

- El elemento **<link>** describe un cuerpo rígido con inercia, propiedades visuales y modelo de colisión.
- 3 Elementos podremos definir dentro de <link>
  - <inertial>
  - <visual>
  - <collision>
- Más opciones: <http://wiki.ros.org/urdf/XML/link>



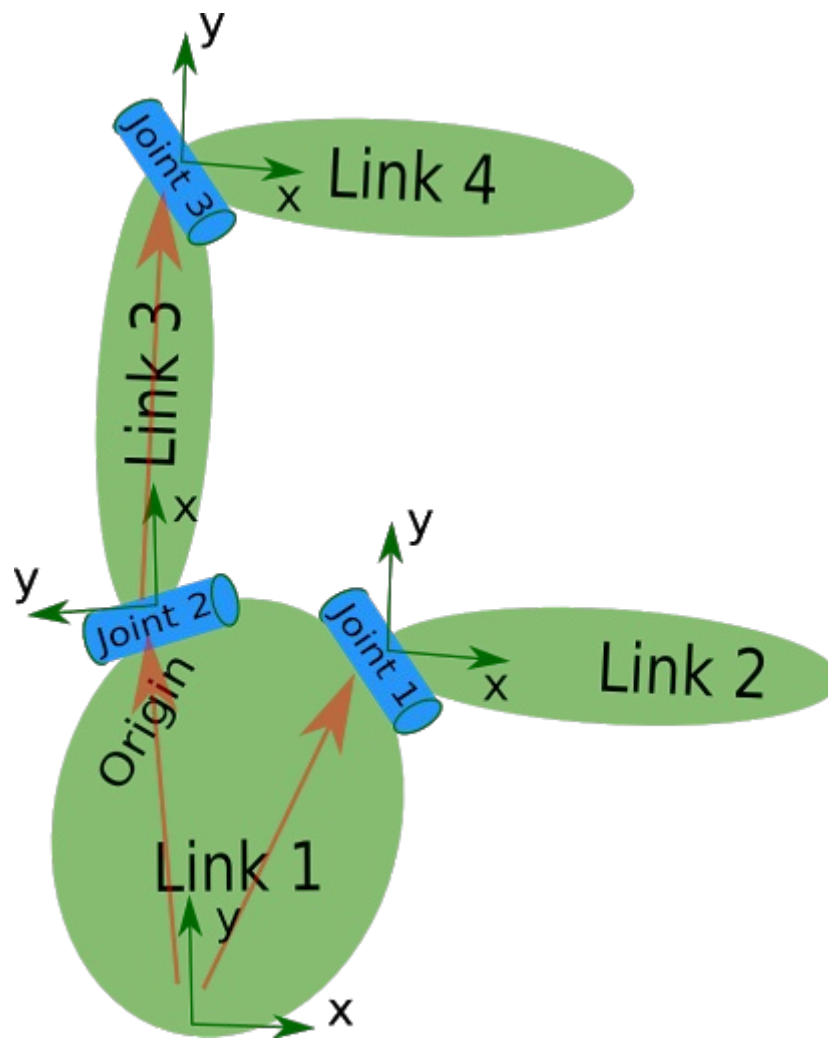
# URDF - XML

- El elemento *joint* describe la cinemática y dinámica de la articulación e incluso se pueden definir los límites de los movimientos.
- Un *joint* está formado por 2 links:
  - Parent
  - Child
- Podrá contener restricciones de movimientos
- <http://wiki.ros.org/urdf/XML/joint>



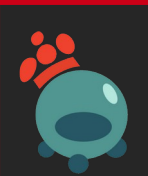


# URDF - XML



# URDF - Joints

- **Revolute:** Rota sobre uno de sus ejes y se puede especificar sus limites superiores e inferiores
- **Continuous:** Similar al revolute pero sin límites.
- **Prismatic:** Se desliza a lo largo del eje.,
- **Fixed:** Bloquea movimiento en todos los grados.
- **Floating:** Permite movimiento en 6-DOF
- **Planar:** Permite movimiento en un plano perpendicular al eje



# URDF - XML

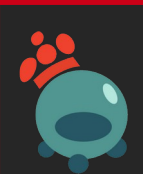
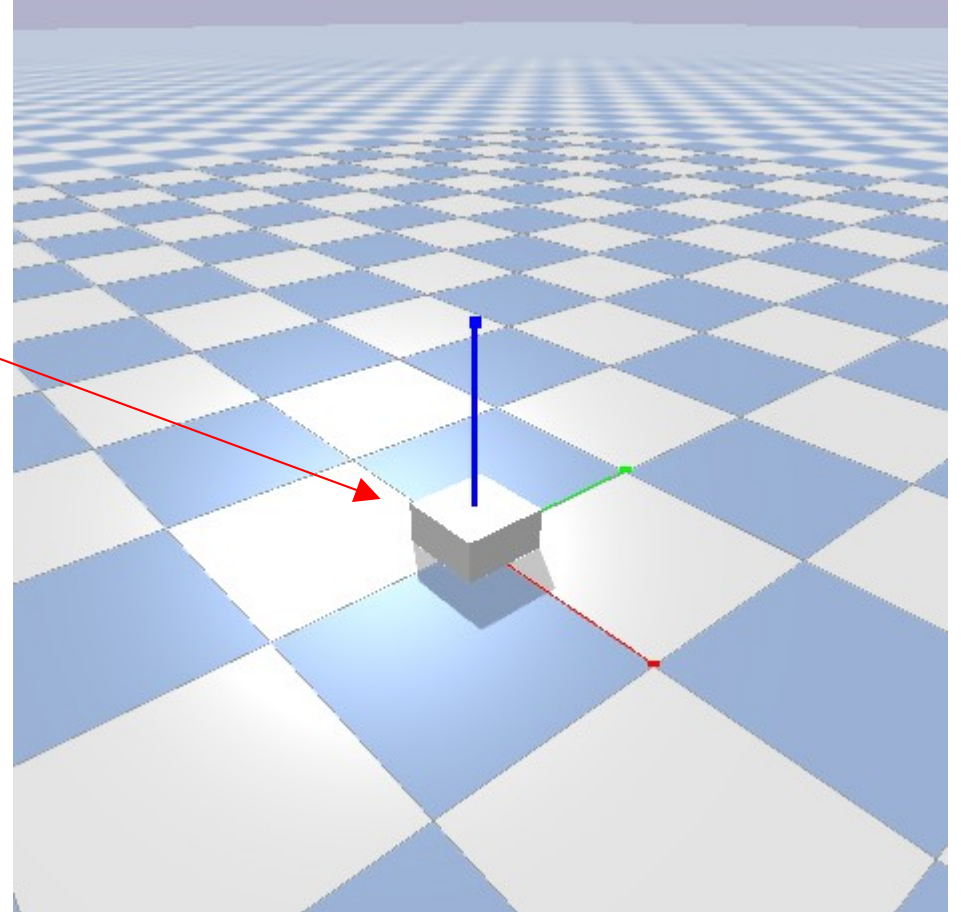
```
<?xml version="1.0"?>
<robot name="myfirstrobot">

  <link name="base_link">

    <visual>
      <origin xyz="0 0 0.1"/>
      <geometry>
        <box size="0.4 0.4 0.2"/>
      </geometry>
    </visual>

  </link>

</robot>
```



# URDF - XML

- Añadimos colisión e inercia
- Ahora nuestro objeto tiene dinámica de movimientos
- Cada modelo podría tener un tamaño diferente
- Lo ideal es que el modelo visual y el modelo de colisión tenga la misma geometría y tamaño.

```
<?xml version="1.0"?>
<robot name="myfirstrobot">

  <link name="base_link">

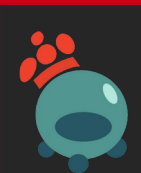
    <inertial>
      <origin xyz="0 0 0.1" />
      <mass value="10.0" />
      <inertia ixx="1.0" ixy="0.0" ixz="0.0"
        iyy="1.0" iyz="0.0" izz="1.0" />
    </inertial>

    <visual>
      <origin xyz="0 0 0.1"/>
      <geometry>
        <box size="0.4 0.4 0.2"/>
      </geometry>
    </visual>

    <collision>
      <origin xyz="0 0 0.1"/>
      <geometry>
        <box size="0.4 0.4 0.2"/>
      </geometry>
    </collision>

  </link>

</robot>
```



# URDF - XML

- Creamos otro link en nuestro robot llamado body\_link
- Podemos asignar colores a las distintas partes de nuestro robot con <material>
- Nuestro xml contiene ahora 2 links
  - base\_link
  - body\_link
- Necesitamos definir la articulación entre los 2 links.

```
<material name="blue">
  <color rgba="0 0 0.8 1"/>
</material>

<link name="body_link">

  <inertial>
    <origin rpy="0 0 0" xyz="0 0 0.5" />
    <mass value="1.0" />
    <inertia ixx="1.0" ixy="0.0" ixz="0.0"
      iyy="0.0" iyz="0.0" izz="1.0" />
  </inertial>

  <visual>
    <origin rpy="0 0 0" xyz="0 0 0.5"/>
    <geometry>
      <box size="0.1 0.1 0.8"/>
    </geometry>
    <material name="blue"/>
  </visual>

  <collision>
    <origin rpy="0 0 0" xyz="0 0 0.5"/>
    <geometry>
      <box size="0.1 0.1 0.8"/>
    </geometry>
  </collision>

</link>
```



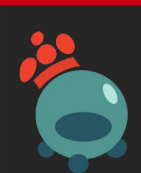
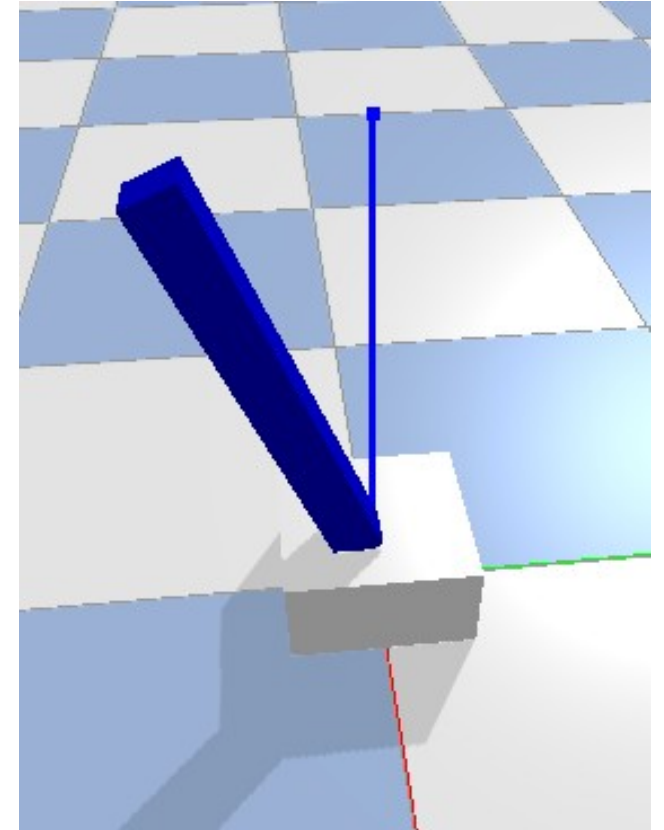
# URDF - XML

- Generamos un joint con 2 links (base y body)
- Solo permitimos rotación sobre su eje X

```
<joint name="base_to_body" type="revolute">
  <parent link="base_link"/>
  <child link="body_link"/>
  <dynamics damping="0.0" friction="0.0001"/>

  <origin xyz="0.05 0 0.1"/>
  <axis xyz="1 0 0"/>

  <limit lower="-3.14" upper="3.14" effort="10.0" velocity="0.5"/>
</joint>
```





# URDF - Gazebo

```
<?xml version="1.0"?>
<robot name="simple_camera_robot">
  <!-- Definición de un enlace simple que
  actúa como base -->
  <link name="base_link">
    <visual>
      <geometry>
        <box size="0.1 0.1 0.1"/>
      </geometry>
      <material name="blue">
        <color rgba="0 0 1 1"/>
      </material>
    </visual>
  </link>

  <!-- Definición de la cámara -->
  <link name="camera_link">
    <visual>
      <origin xyz="0.2 0 0.5" rpy="0 0 0"/>
      <geometry>
        <cylinder length="0.1" radius="0.05"/>
      </geometry>
      <material name="black">
        <color rgba="0 0 0 1"/>
      </material>
    </visual>
  </link>
```

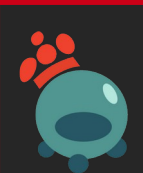
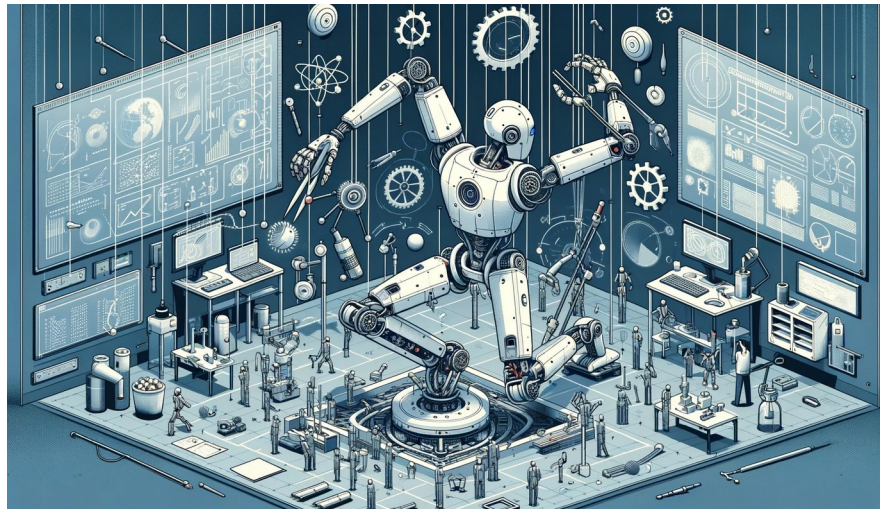
```
<!-- Articulación que conecta la base con la cámara -->
<joint name="camera_joint" type="fixed">
  <parent link="base_link"/>
  <child link="camera_link"/>
  <origin xyz="0.2 0 0.5" rpy="0 0 0"/>
</joint>

<!-- Plugin de Gazebo para la cámara -->
<gazebo reference="camera_link">
  <sensor type="camera" name="camera1">
    <camera>
      <horizontal_fov>1.3962634</horizontal_fov>
      <image>
        <width>800</width>
        <height>600</height>
      </image>
      <clip>
        <near>0.02</near>
        <far>300</far>
      </clip>
    </camera>
    <plugin name="camera_controller" filename="libgazebo_ros_camera.so">
      <!-- Configuraciones adicionales del plugin aquí -->
      <cameraName>camera1</cameraName>
      <imageTopicName>camera1/image_raw</imageTopicName>
      <cameraInfoTopicName>camera1/camera_info</cameraInfoTopicName>
      <frameName>camera_link</frameName>
    </plugin>
  </sensor>
</gazebo>
</robot>
```



# URDF

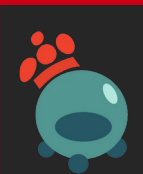
- Limitaciones
  - Modelado de dinámicas complejas
  - Simulación de sensores
  - Descripción de entornos
  - Definición de efectos ambientales (viento/luz)





# SDF

- SDF es un formato desarrollado específicamente para descripciones detalladas de simulaciones y robots en el simulador Gazebo.
- Solventa algunas de las limitaciones de URDF
  - sensores, los actuadores, las luces, las cámaras, y las interacciones con entornos físicos complejos.
- Permite modelar la dinámica del robot, entornos, elementos estáticos y dinámicos.
- Estructura basada en XML
- Mucho más flexible y extensible que URDF



## SDF

```

<?xml version="1.0" ?>
<sdf version="1.6">
  <world name="r2d2_world">

    <!-- Añadir un modelo de robot desde un archivo URDF -->
    <include>
      <uri>model://r2d2.urdf</uri>
    </include>

    <!-- Definir el escenario -->
    <model name="carretera">
      <static>true</static>
      <link name="ground_plane">
        <visual name="ground_visual">
          <geometry>
            <plane>
              <normal>0 0 1</normal>
              <size>100 100</size>
            </plane>
          </geometry>
          <material>
            <script>
              <uri>file://media/materials/scripts/gazebo.material</uri>
              <name>Gazebo/GreenGrass</name>
            </script>
          </material>
        </visual>
      </link>
    </model>
  </world>
</sdf>

```

```

    <!-- Añadir iluminación ambiental -->
    <light type="directional" name="sun">
      <cast_shadows>true</cast_shadows>
      <pose>0 0 10 0 0 0</pose>
      <diffuse>0.8 0.8 0.8 1</diffuse>
      <specular>0.2 0.2 0.2 1</specular>
      <attenuation>
        <range>100</range>
        <constant>0.9</constant>
        <linear>0.01</linear>
        <quadratic>0.001</quadratic>
      </attenuation>
      <direction>-0.5 -0.5 -1</direction>
    </light>

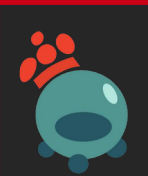
    </world>
  </sdf>

```



# Bibliografía

- Mastering ROS for Robotics Programming (2nd Edition)
- <http://wiki.ros.org/urdf/Tutorials>
- Components of SDF Models
- [SDFormat.org](http://SDFormat.org)





Escuela de Ingeniería  
de Fuenlabrada



**RoboticsLabURJC**  
Programming Robot Intelligence

