



APORTES AL ENTORNO DOCENTE DE ROBÓTICA JDEROBOT-ACADEMY



Álvaro Villamil Vuelta
a.villamil@alumnos.urjc.es

13 de julio de 2017

Índice

- Introducción
- Objetivos
- Infraestructura
- Circuito de carreras de Fórmula 1
- Brazo Robótico
- Conclusiones

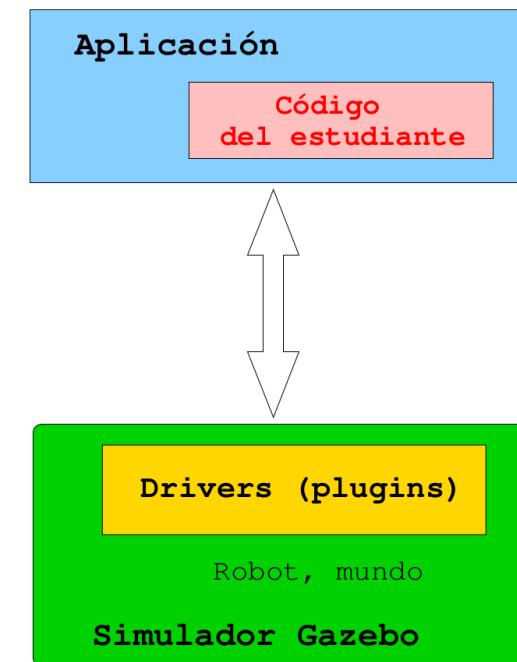
Introducción

Enseñanza en Robótica

- Secundaria: Asignatura de Tecnología; NXT, WeDo, Scratch, JdeRobot-Kids, Arduino...
- Grados: Algunas asignaturas; ROS y Matlab
- Másters: Asignaturas y cursos especializados

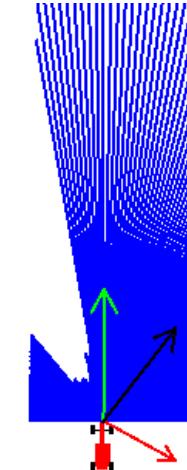
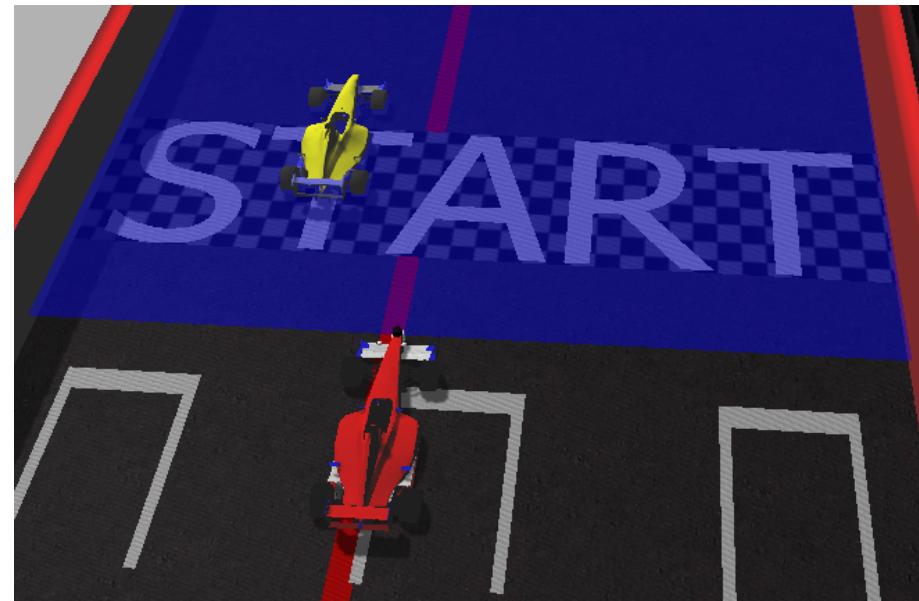
JdeRobot-Academy

- Entorno docente de robótica universitaria orientado al aprendizaje práctico.
- Énfasis en la programación de la inteligencia de los robots.
- Colección de prácticas variadas.
- Utiliza Gazebo y Python.



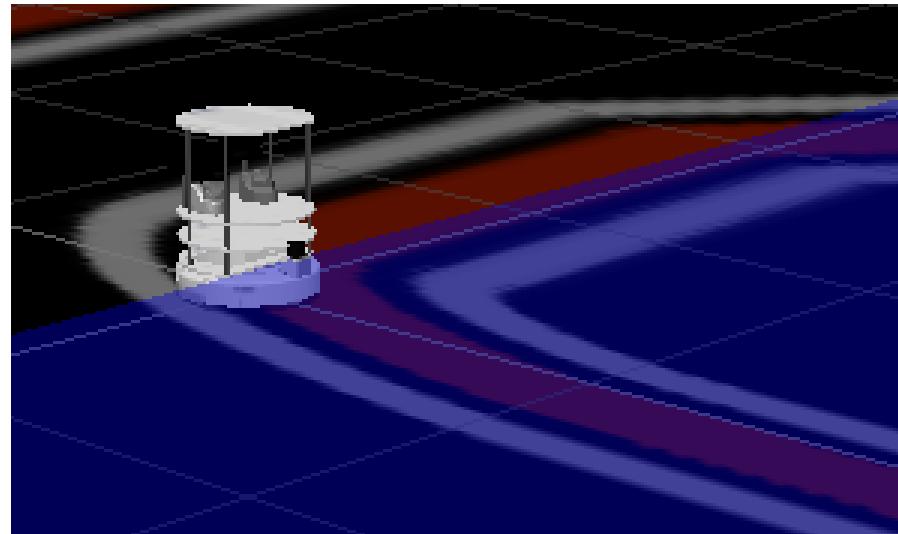
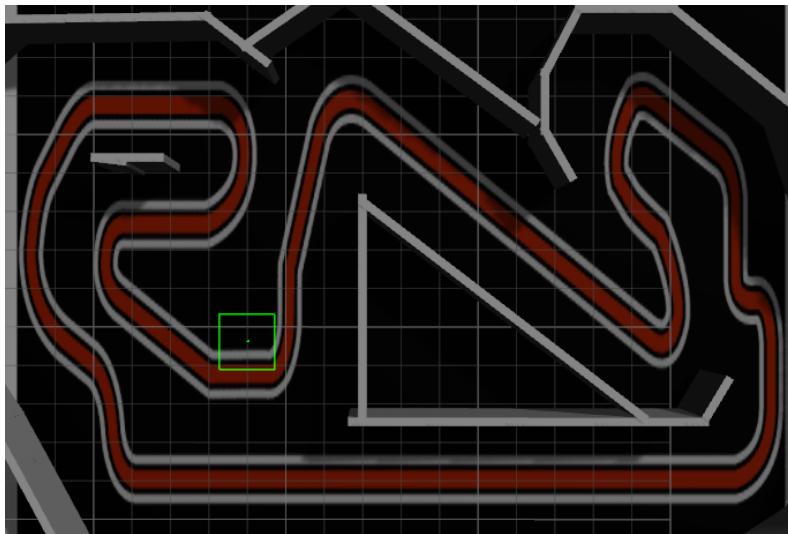
Práctica navegación local Fórmula 1.

- Coche con sensores
- Esquivar obstáculos.
- Algoritmo de navegación local (VFF).



Práctica sigue-líneas.

- kibuki con cámara
- Seguir línea roja.
- Procesamiento de imagen.



Objetivo

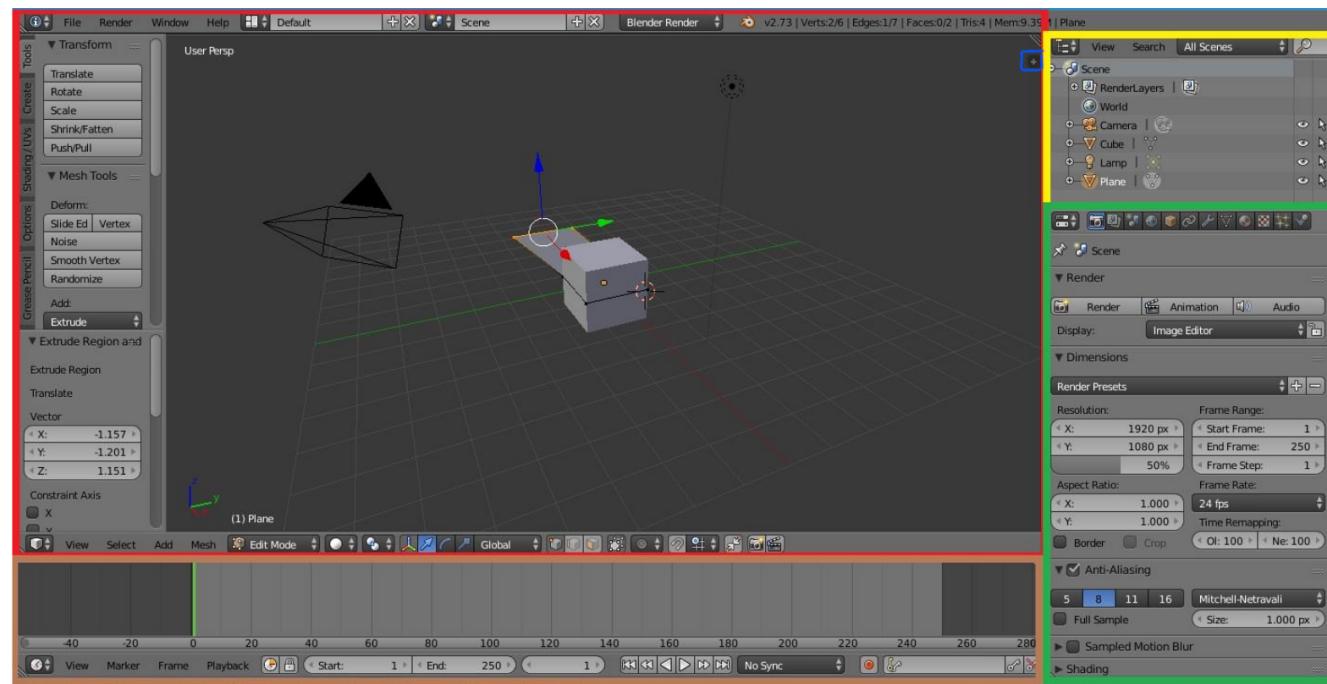
Mejorar y ampliar la colección de prácticas de JdeRobot-Academy, enriqueciéndolas y aumentando el abanico de posibilidades que se ofrece al alumno.

- Mejorar la infraestructura en Gazebo de las prácticas de JdeRobot-Academy que usan coches de carreras.
- Diseñar y programar un teleoperador de un brazo robótico en Gazebo.

Infraestructura

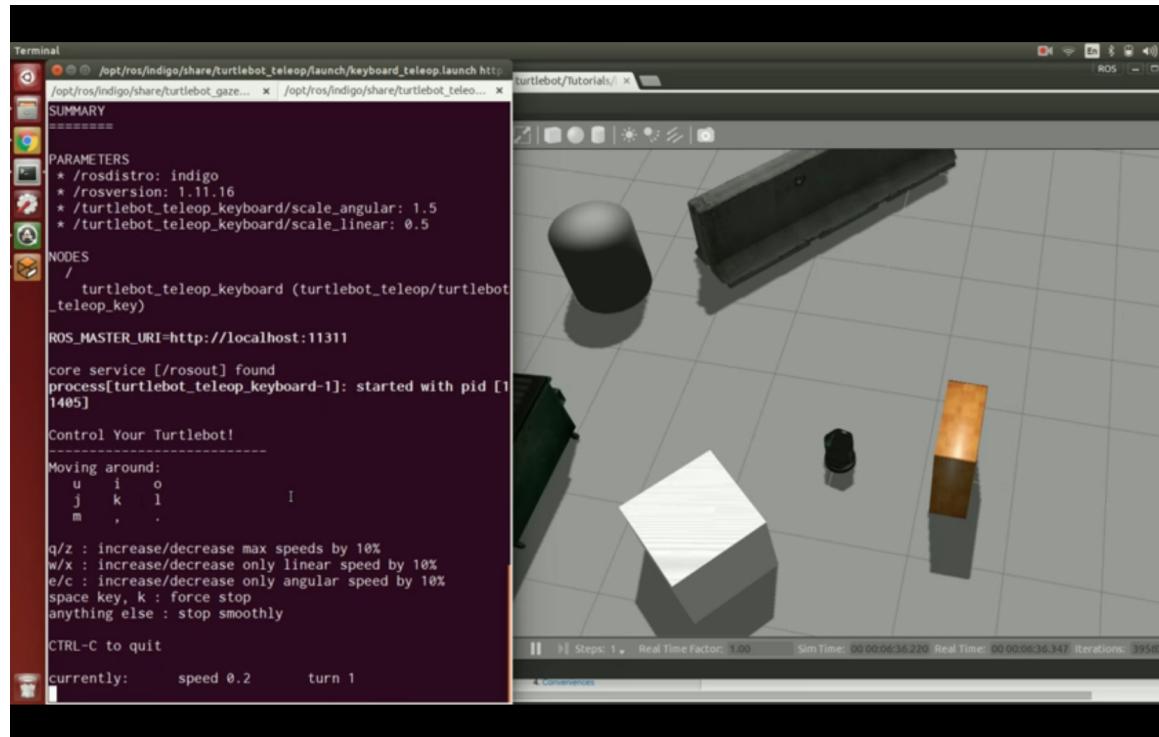
Blender

- **Blender:** Programa de modelado, iluminación, renderizado, animación y creación de gráficos tridimensionales.



Gazebo y ROS

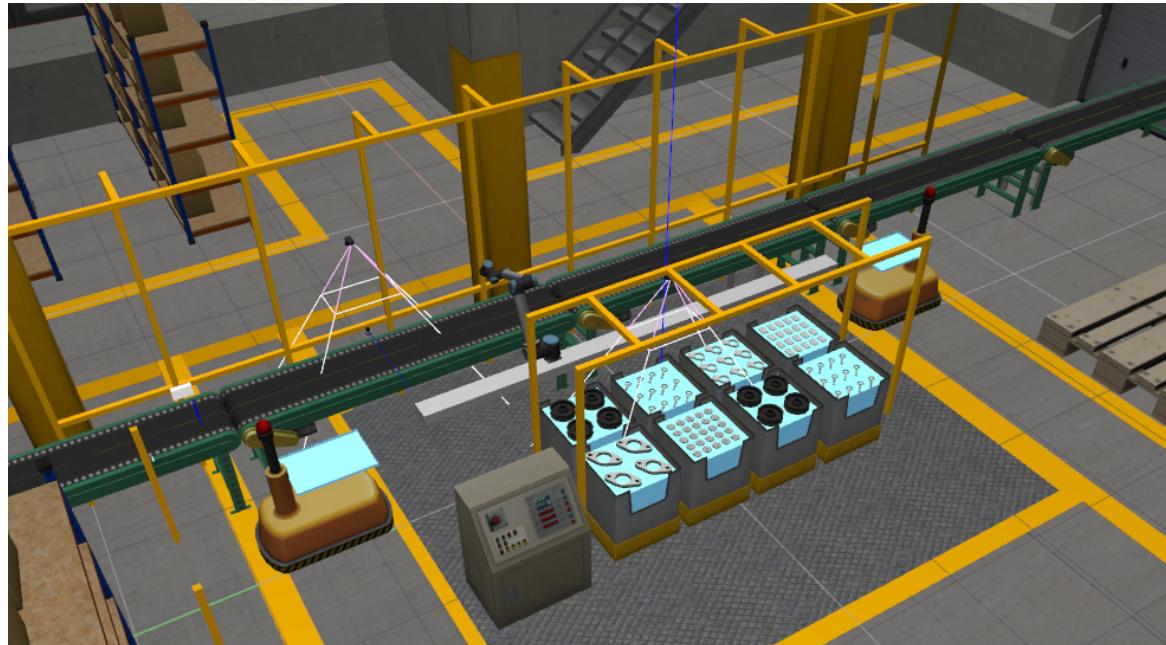
- **Gazebo**: Simula sensores, actuadores, robots,... en mundos virtuales.
- **ROS**: Es un *middleware* para robots.



ARIAC

(Agile Robotics for Industrial Automation Competition).

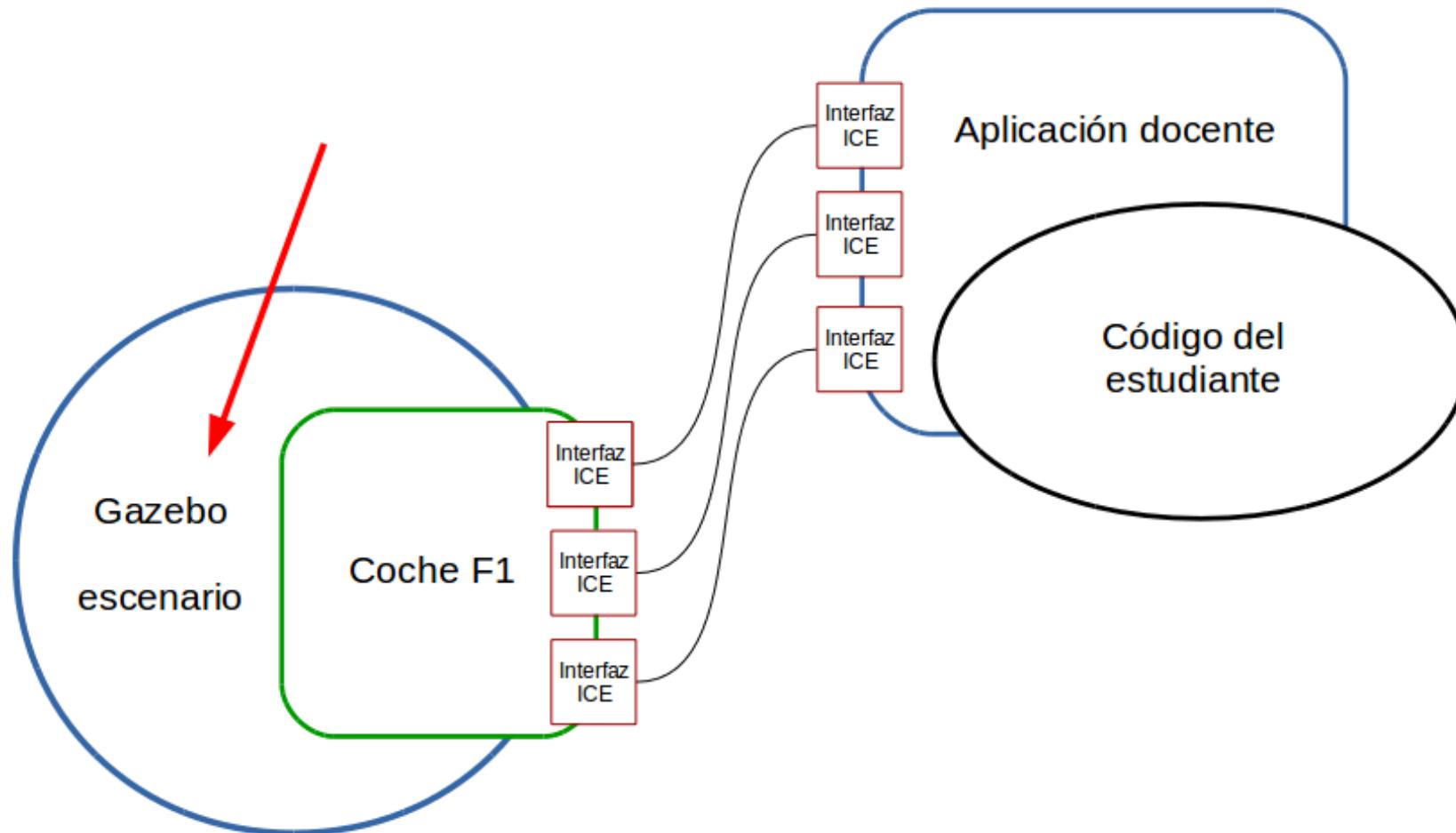
Competición para probar la agilidad de los sistemas robóticos industriales.



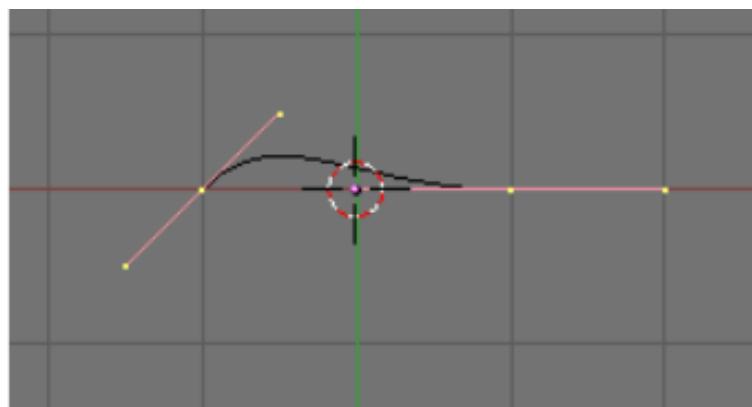
Circuito de carreras de Fórmula 1

- Reconstruir el circuito de Mónaco de forma que se pueda utilizar en:
 - Práctica sigue línea.
 - Práctica navegación local.
- Creación de mundos para Gazebo que sigan el esquema de las prácticas.

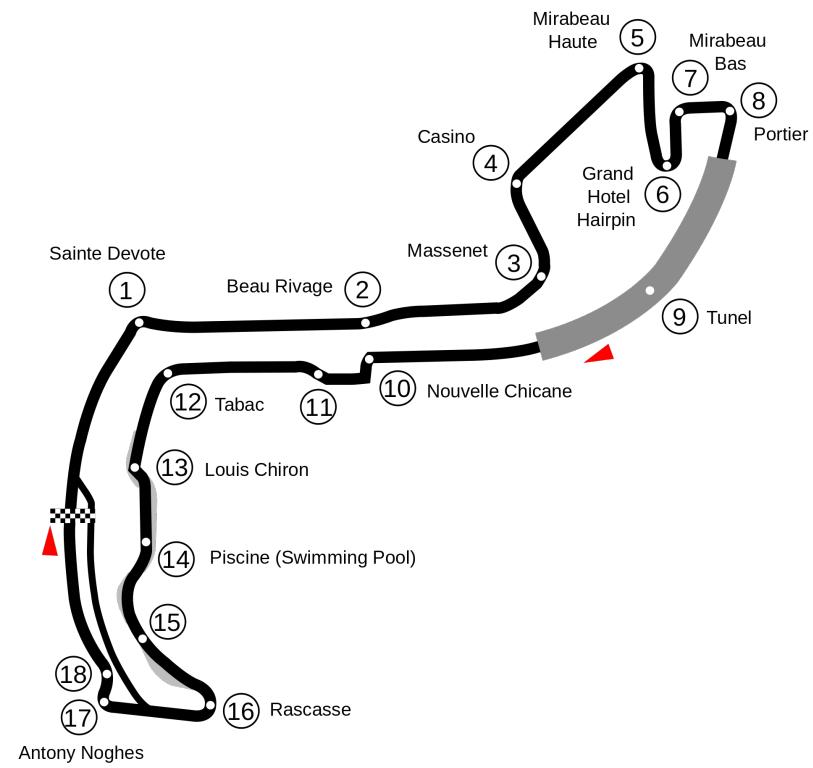
Esquema de componentes



Círculo plano

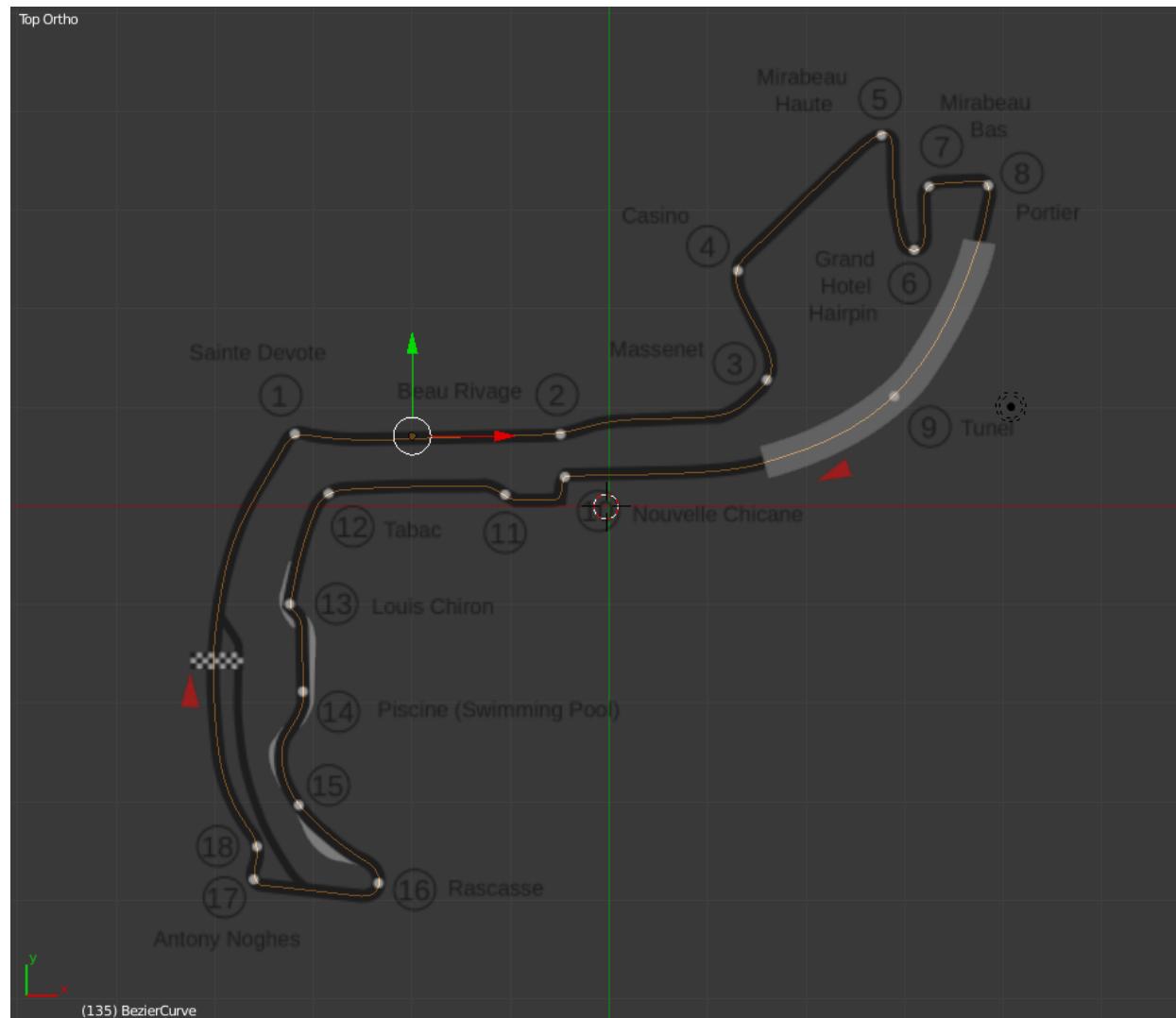


Curva Bezier



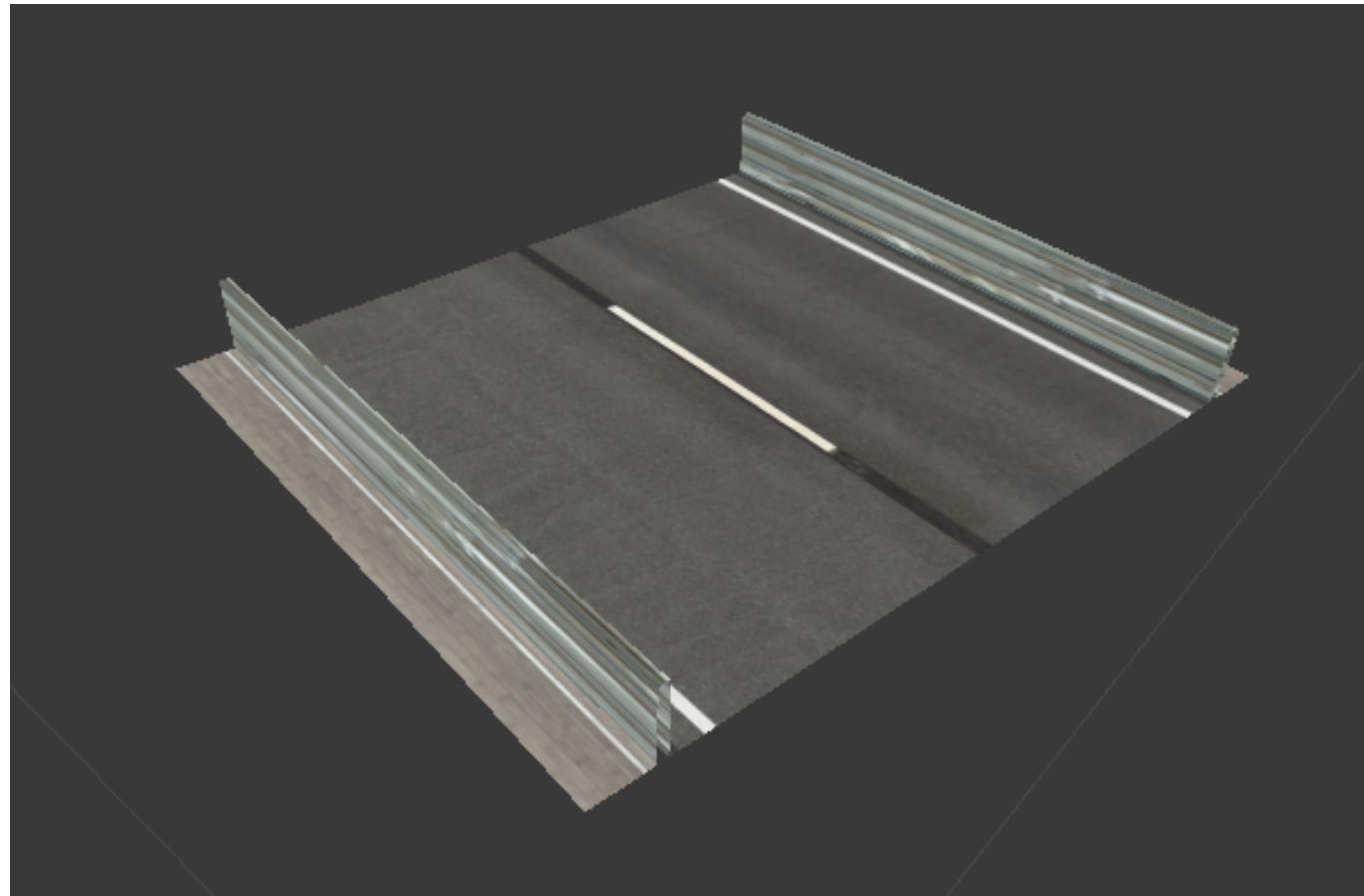
Plano del circuito

Curva del trazado

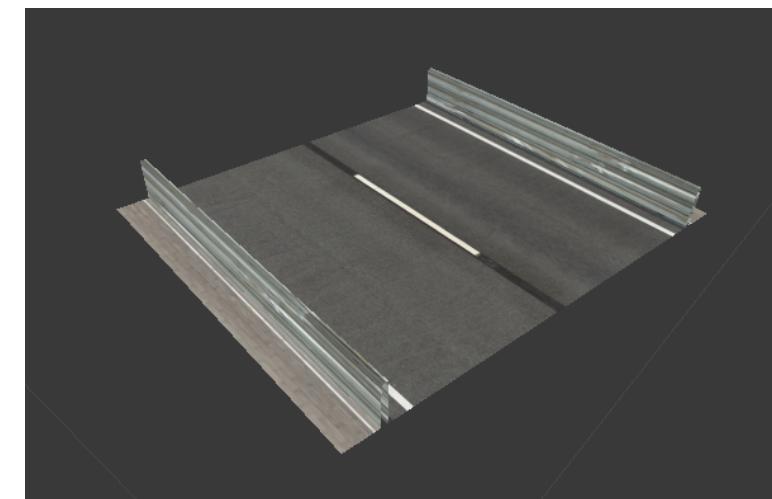
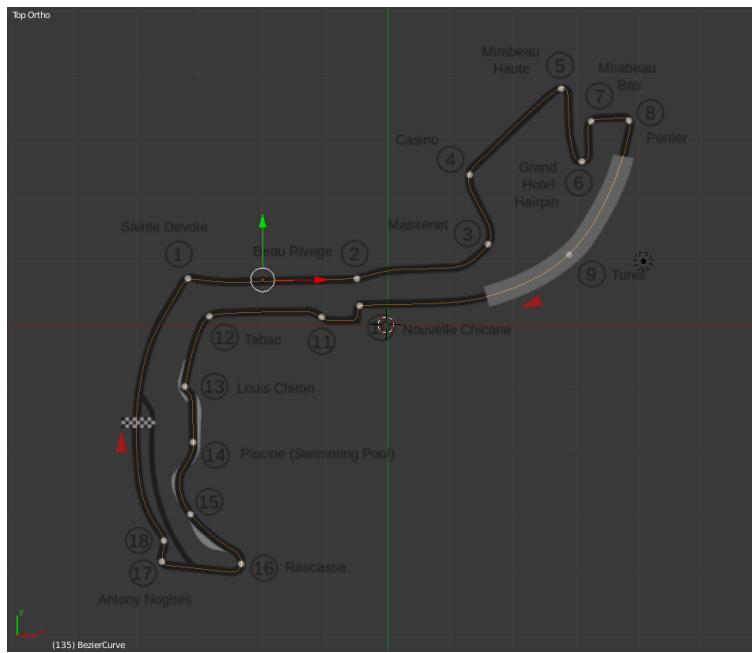


- Modelamos un segmento de circuito. Creamos:
 - Carretera.
 - Vallas.
 - Aceras.
- Texturizamos cada zona.
- Asignamos materiales a las superficies.

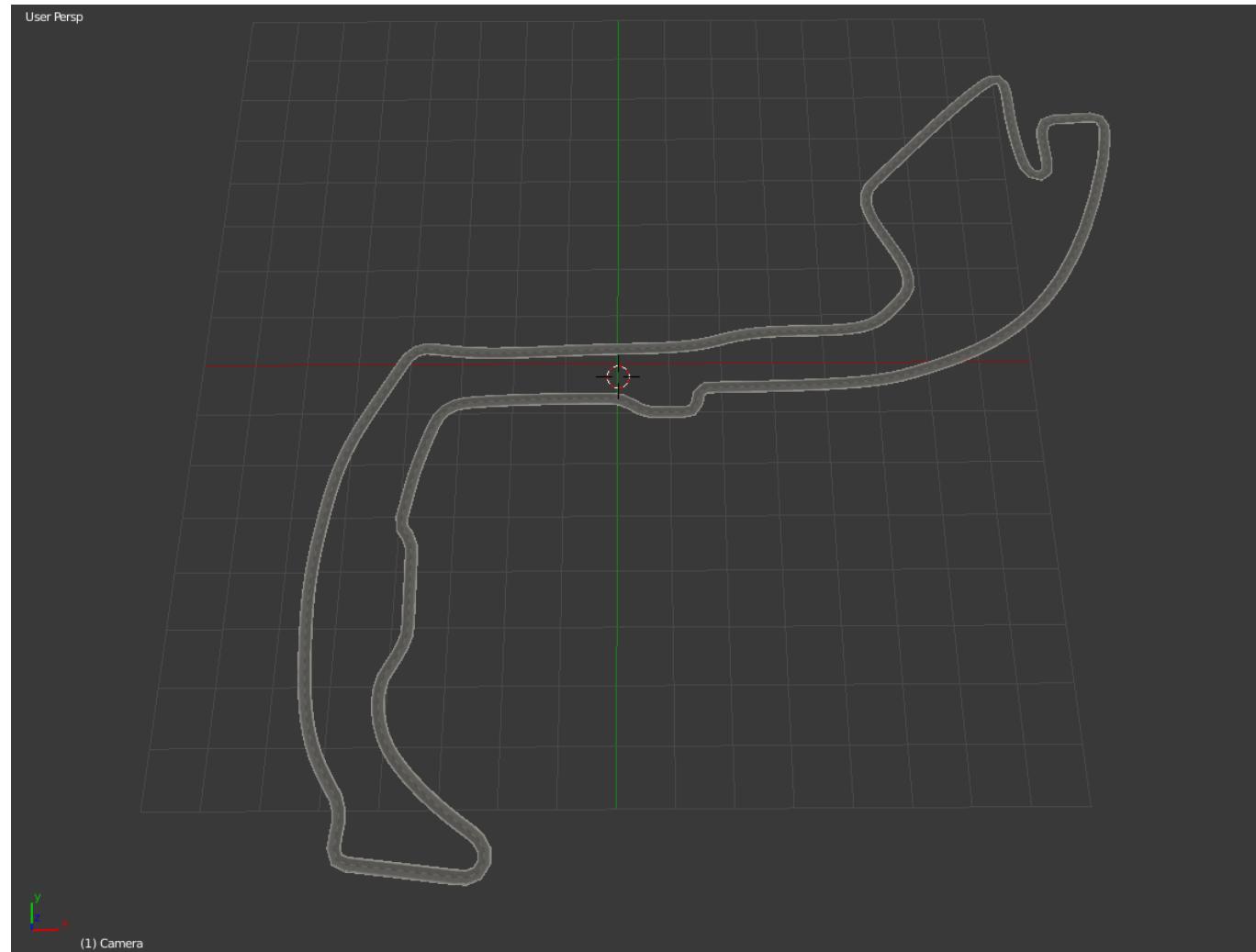
Segmento del circuito



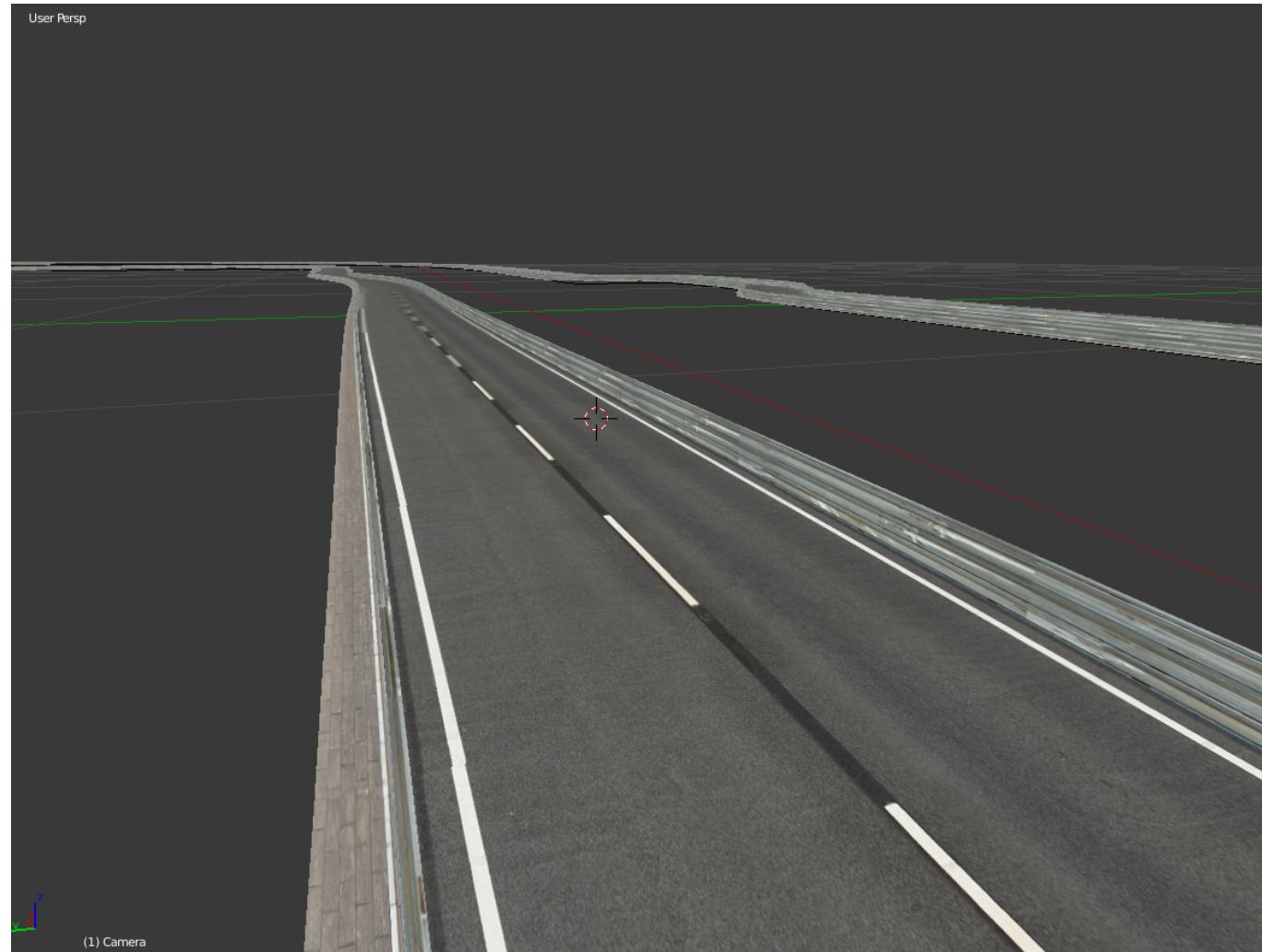
Aplicamos modificadores:



Circuito modelado:

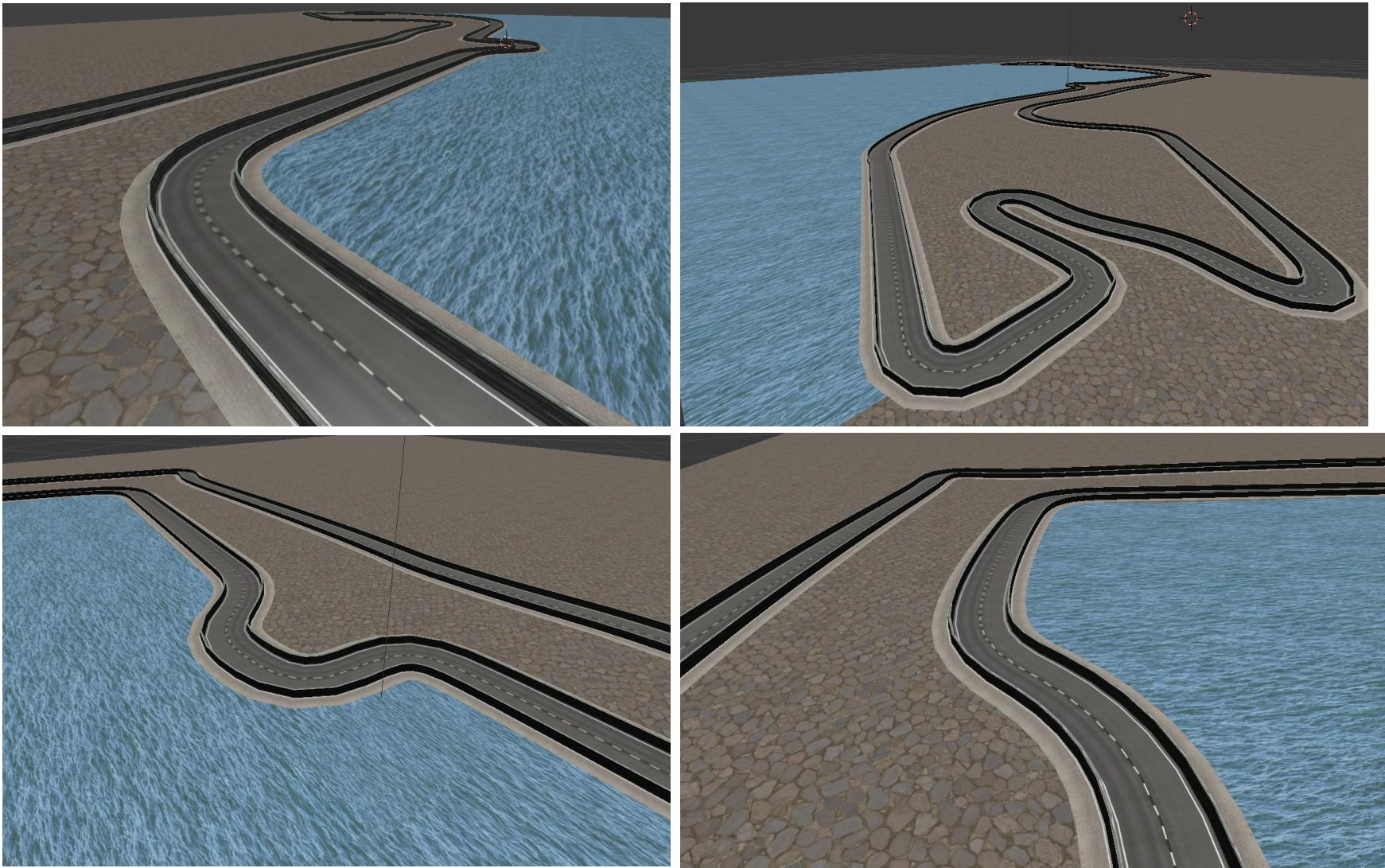


Detalle del circuito modelado:



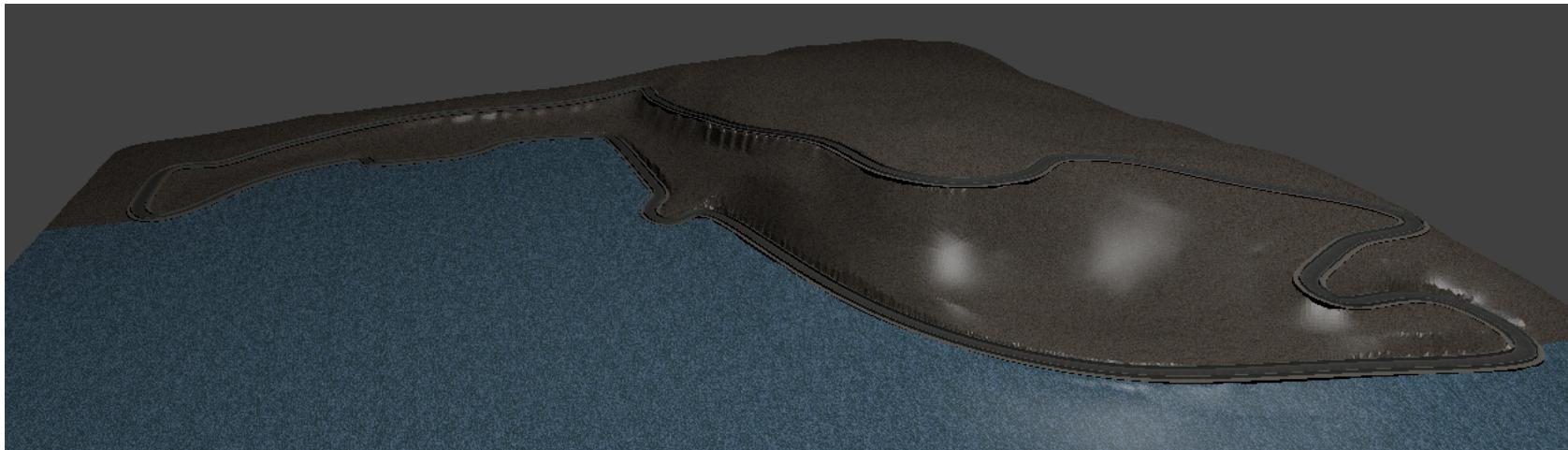
- Añadimos fondo al circuito
- Texturizamos de forma realista

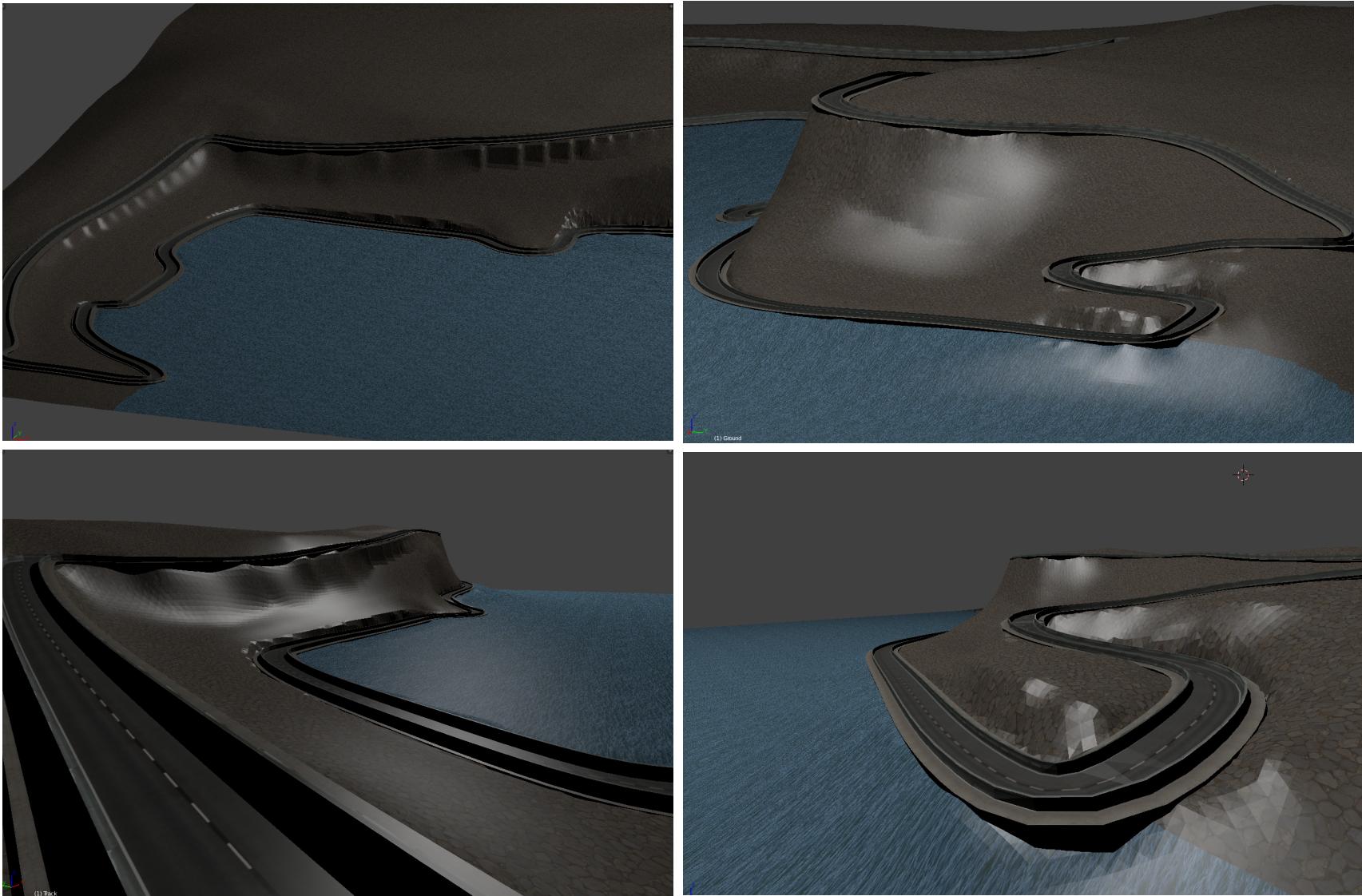




Circuito con elevaciones

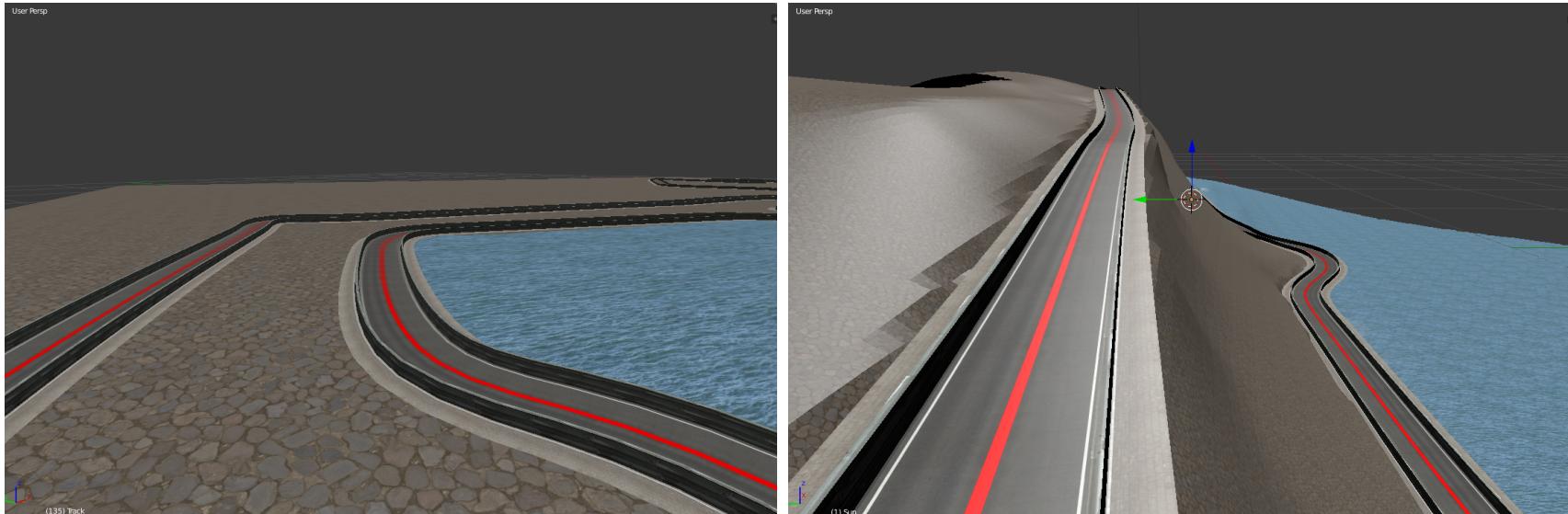
- Creamos *mesh* y esculpimos el fondo.
- Modelamos y texturizamos trazado.
- “Modificamos” el trazado con *mesh*.
- Texturizamos el circuito.



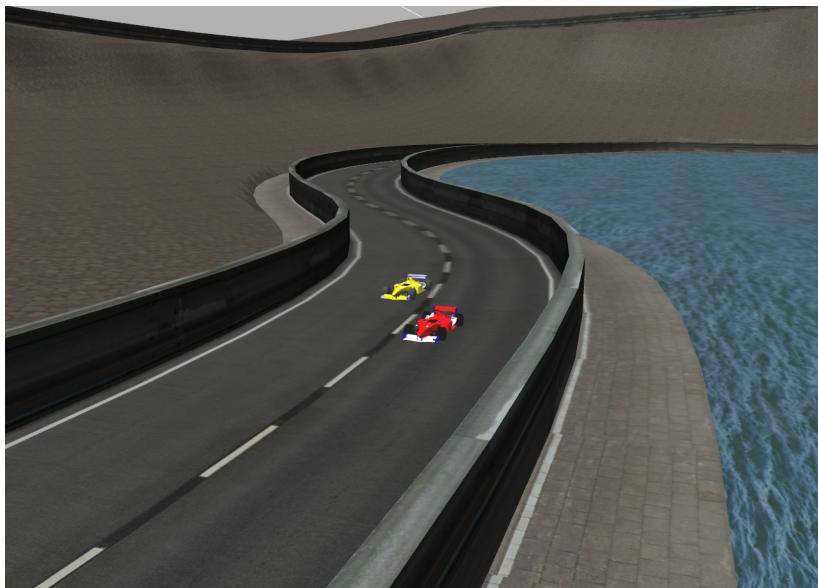


Circuitos con línea roja

Modificamos la textura del asfalto pintando la línea roja.

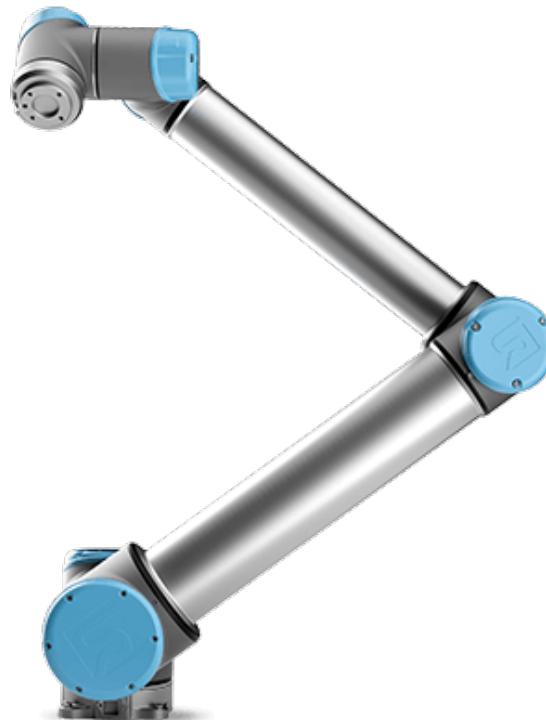


Circuitos en las prácticas

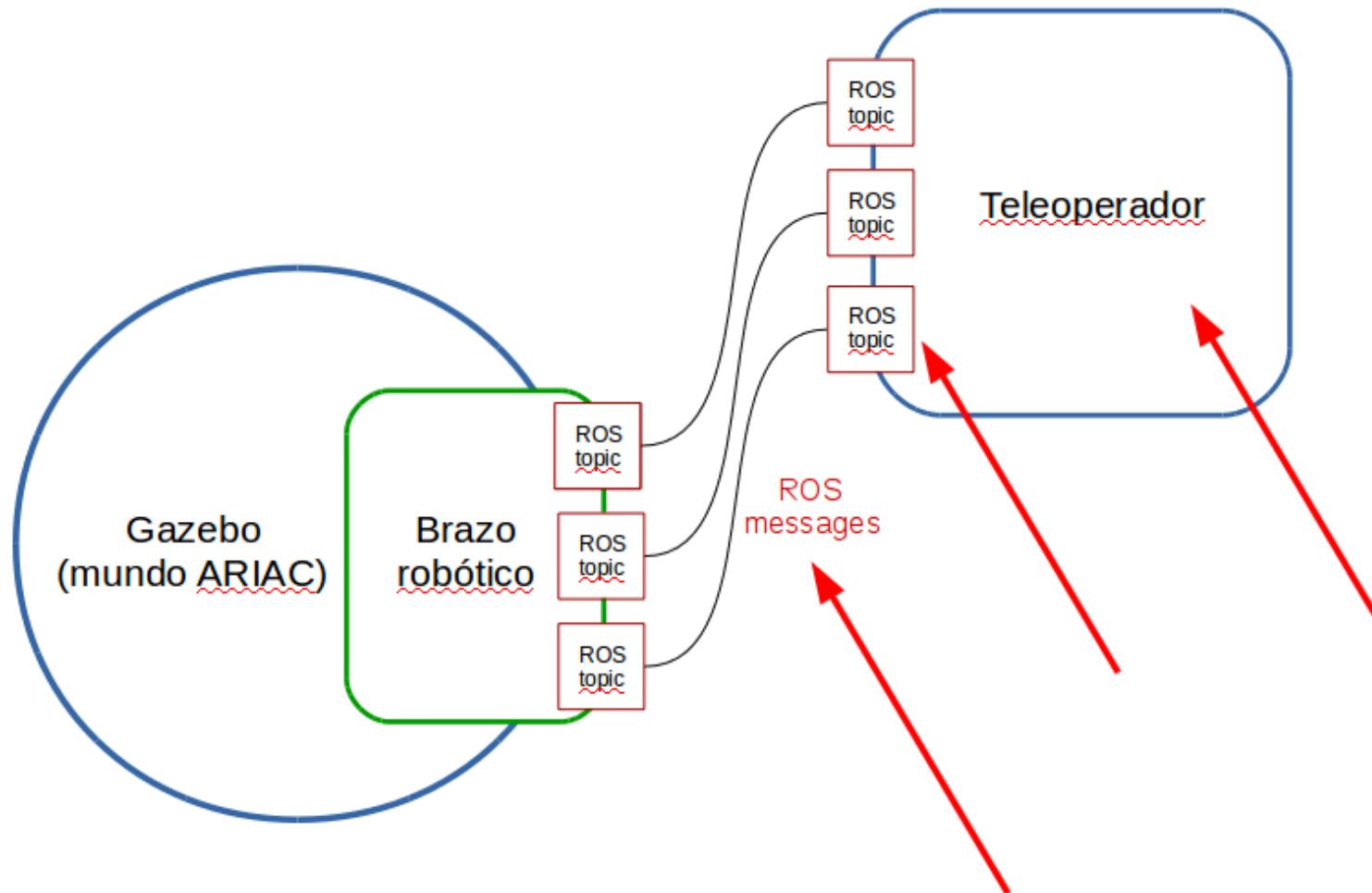


Brazo robótico

- Buscamos brazo que reuna requisitos:
 - Gazebo 7.
 - ROS Kinetic.
- Elegimos escenario ARIAC:
 - UR10



Esquema de componentes



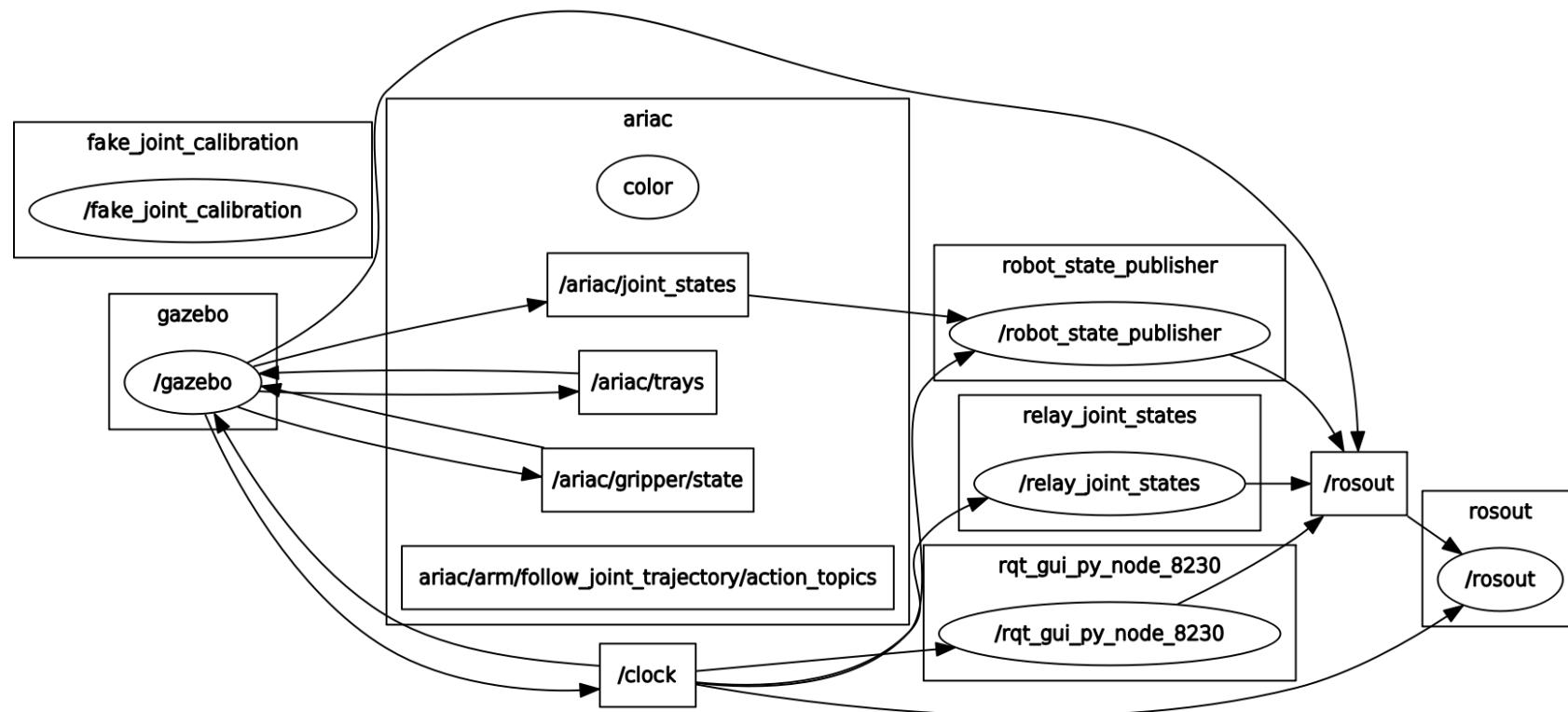
ROS

Elementos básicos:

- **Nodos**: Procesos ejecutables. Se comunican entre sí:
 - : Publica en topics.
 - *Subscriber*: Recibe de topics.
- **Topics**: Canales de comunicación entre nodos de tipado fuerte.
- **Mensajes**: Estructuras de datos que se envían por los topics.

Se organiza en grafo.

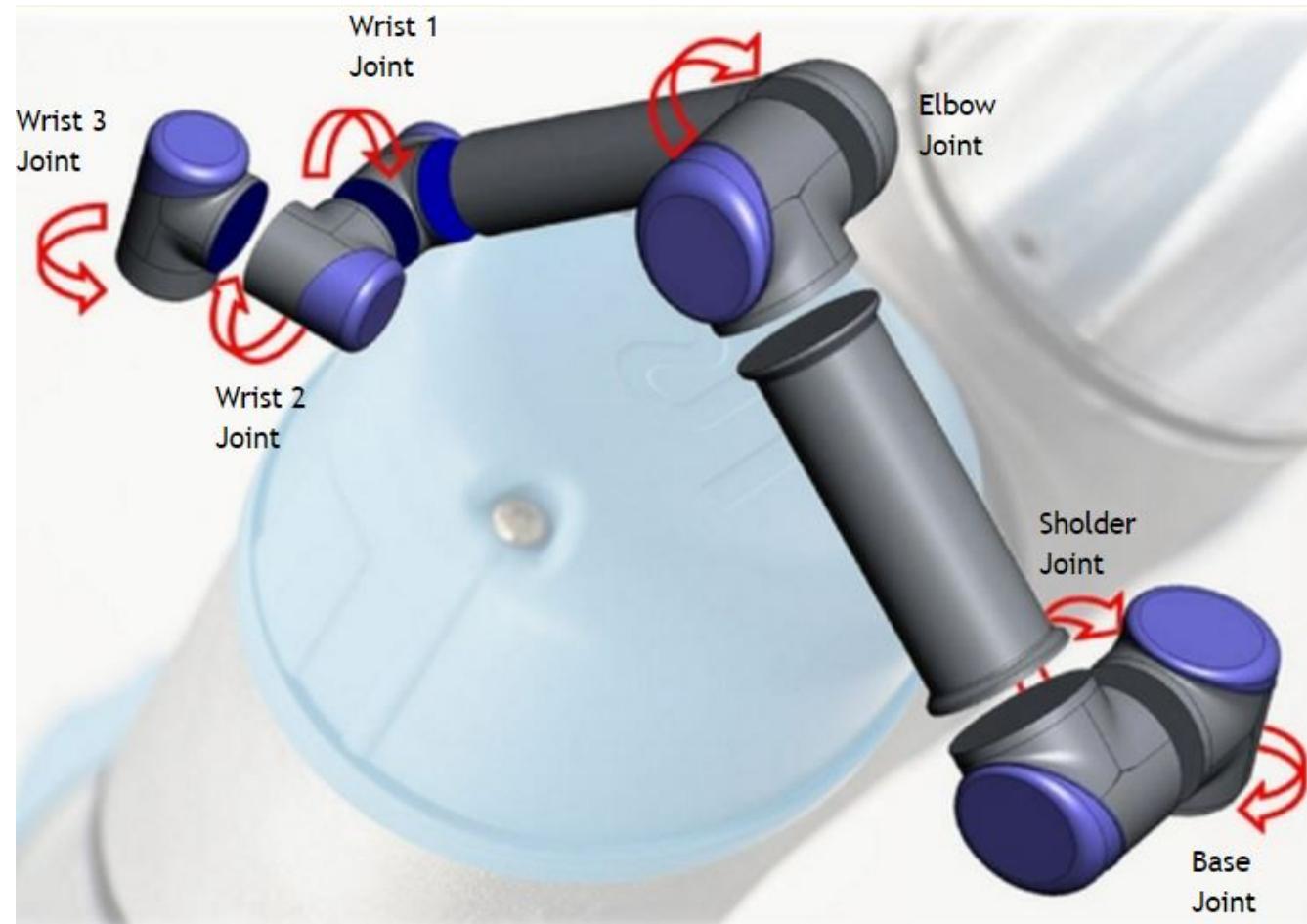
ARIAC en ROS



Topics y mensajes importantes

- **/ariac/joint_states**: Publica las posiciones de las articulaciones del brazo.
 - Mensaje: *sensor_msgs/JointState Message*.
- **/ariac/arm/command**: Recibe la posición a la que mover el brazo.
 - Mensaje: *trajectory_msgs/JointTrajectory Message*.

Articulaciones del brazo



Mensaje de posición

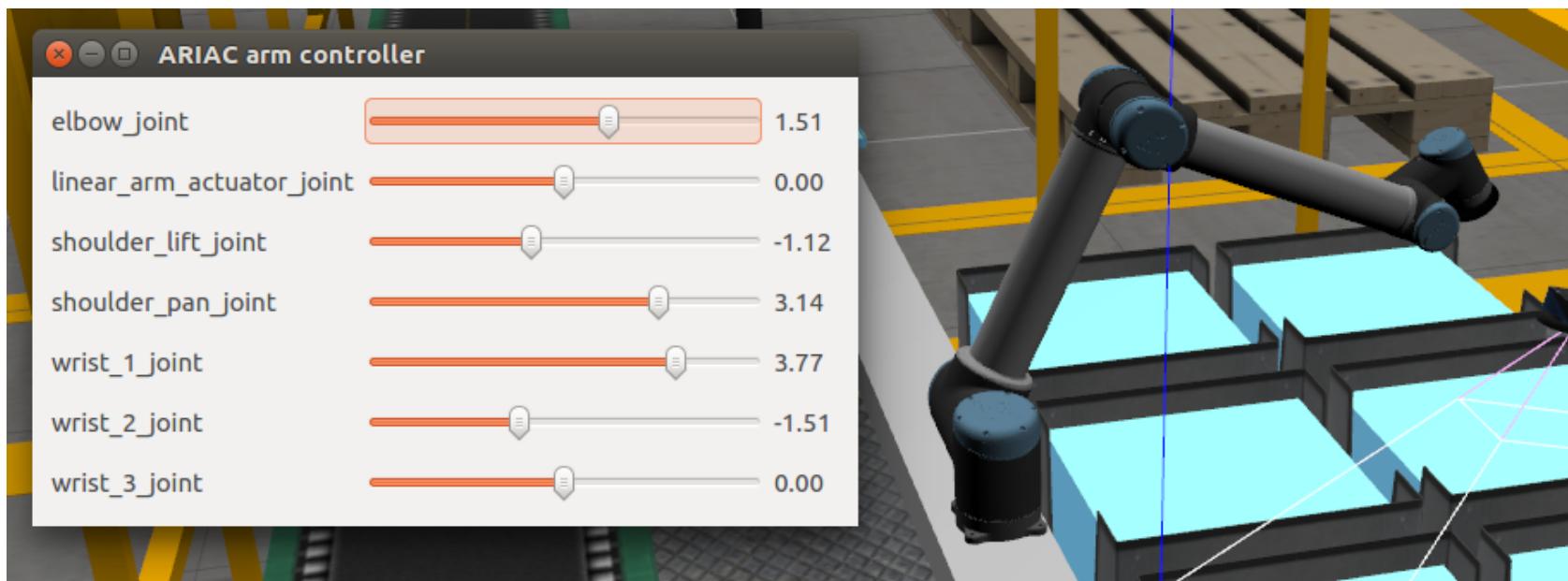
```
1 ----  
2 header:  
3 seq: 265  
4 stamp:  
5 secs: 5  
6 nsecs: 322000000  
7 frame_id: ''  
8 name: [ 'elbow_joint', 'linear_arm_actuator_joint', 'shoulder_lift_joint',  
      'shoulder_pan_joint', 'wrist_1_joint', 'wrist_2_joint', '  
      wrist_3_joint', 'vacuum_gripper_joint']  
9 position: [1.5072954794978815, 0.04169673392358113, -0.3793140712872205,  
            3.217984505432054, 3.085834205511564, -1.6130778569148077,  
            -0.00793595855568796, 0.0]  
10 velocity: [0.027474972846918494, 0.007286219434794008,  
             0.05461882849916373, 0.0316967471028743, 0.1957558318356749,  
             -1.091011699832568, -8.054309845386587, 0.0]  
11 effort: [150.0, -184.23945050318943, 330.0, 167.6704306272329, 0.0,  
           5.601291580407757, -5.541011517572109, 0.0]  
12 ----
```

Mensaje de orden

```
1 ----
2 header:
3 seq: 59
4 stamp:
5 secs: 0
6 nsecs: 0
7 frame_id: ''
8 joint_names: [ 'elbow_joint', 'linear_arm_actuator_joint', '
    shoulder_lift_joint', 'shoulder_pan_joint', 'wrist_1_joint', '
    wrist_2_joint', 'wrist_3_joint' ]
9 points:
10 -
11 positions: [1.510634135925831, 1.4199980429372933e-06,
    -1.1286545928274752, 3.140002031709801, 3.772089274964015,
    -1.5100101552695162, 4.0770707014914365e-06, 0.0]
12 velocities: []
13 accelerations: []
14 effort: []
15 time_from_start:
16 secs: 1
17 nsecs: 0
18 ----
```

Teleoperador

- **Bajo nivel:** Actúa directamente sobre las articulaciones.
- **Interfaz gráfica:** Construida con Qt, manejo intuitivo.
- **Python:** Programado en Python, compatible con ROS y JdeRobot.



Conclusiones

- **Objetivo cumplido:** Ampliar y mejorar la colección de prácticas de JdeRobot-Academy.
 - Mundo de Gazebo con el Circuito de Mónaco realista.
 - Teleoperador para brazo robótico
- Preparado para las últimas versiones de JdeRobot (5.5), Gazebo (7) y ROS (Kinetic).

Trabajos futuros:

- Mejora del diseño del Fórmula 1.
- Uso de planificadores de movimiento articulado.
- Desarrollo de práctica *pick&place*.

Enlaces

- Mediawiki: <http://jderobot.org/Avillamil-tfg>
- Repositorio:
<https://github.com/RoboticsURJC-students/2016-tfg-Alvaro-Villamil>