Creador del documento: Carmen Menéndez Domínguez-Adame
Fecha de creación: 03/10/2026

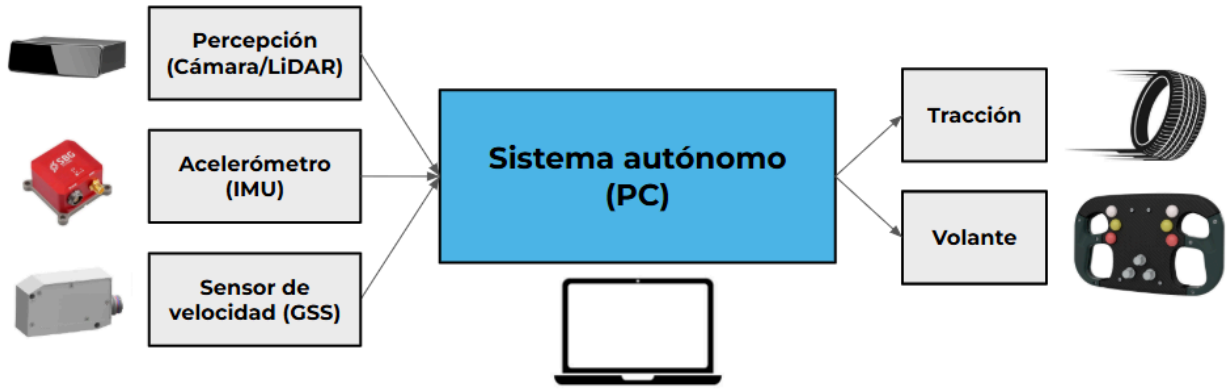
Informe Academy Driverless 2026/2027

Para cualquier duda no dudes en escribir a driverless.arus1@gmail.com

En el departamento **Driverless** nos encargamos de hacer posible la conducción autónoma del monoplaza. Sin embargo, no se trata del mismo tipo de conducción a la que se enfrenta un Tesla u otro tipo de vehículo autónomo en las vías urbanas, en la que hay que tener en cuenta muchos factores externos, sino que se trata de un **problema cerrado** bajo unas condiciones específicas que podemos aprovechar para desarrollar nuestros algoritmos.

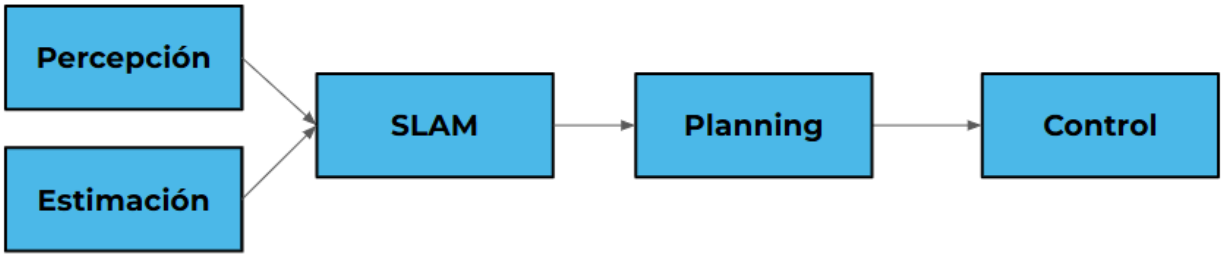
Básicamente, los circuitos están delimitados por un par de líneas de conos, azules a la izquierda y amarillos a la derecha, y el objetivo de todas las pruebas DV es recorrer dicho circuito en el menor tiempo posible (teniendo en cuenta también las penalizaciones por tirar conos o salirse del circuito).

Concretamente, hay **4 eventos DV:** Skidapd, Acceleration, Autocross y Trackdrive (10 vueltas de Autocross). En todos ellos, el coche debe poder acabar la prueba **sin recibir ningún tipo de comanda externa** más allá del botón de Start, funcionando únicamente a partir de los datos que recibe del entorno mediante sensores como el LiDAR y los algoritmos que se hayan programado en su unidad de procesamiento, que finalmente generarán una serie de comandas para acelerar y girar el volante como haría un piloto mediante el uso de actuadores (los motores que mueven cada rueda y un motor en la cremallera de la dirección).



Como hemos dicho, el coche recibe información del entorno como input mediante el uso de diversos sensores, y debe dar como output la comanda de aceleración y

ángulo de volante que le permita recorrer el circuito en el menor tiempo posible. Podemos desglosar este problema, en problemas más pequeños que podemos abordar. Este es el esquema general de la **pipeline DV** que siguen prácticamente todos los equipos de Formula Student, estructurado bajo el ecosistema de **ROS2**:



- **Percepción:** se encarga de detectar la posición de los conos en el circuito a partir de datos de algún sensor (normalmente una cámara, un LiDAR o ambos).
- **Estimación de parámetros:** se trata de calcular, con la mayor precisión posible, el valor de variables de estado del coche que necesitamos saber para el correcto funcionamiento de los demás algoritmos, como la velocidad que tiene en cada momento.
- **SLAM (Simultaneous Localization And Mapping):** como bien dice el nombre, se encarga de generar un mapa del circuito y de calcular la posición del coche dentro de ese mapa. Si la Percepción te da un mapa local de los conos (respecto a la posición actual del coche), el SLAM te da un mapa global de los mismos (respecto al punto inicial).
- **Planning:** una vez conoces tu circuito, el Planning se encarga de trazar la ruta que quieres que recorra tu coche, así como la velocidad que quieres que lleve en cada punto.
- **Control:** se encarga de generar las comandas necesarias para seguir la trayectoria que te da el Planning; es decir, cuánto tienes que acelerar en cada momento para tener tu velocidad objetivo y cuánto tienes que girar el volante para seguir tu ruta.

Nota importante: Para poder trabajar con este flujo, todo el desarrollo se realiza sobre **ROS2**. Es altamente recomendable haber completado la **tarea previa de las entrevistas**, ya que en ella se explican las bases para la correcta instalación, configuración y uso de ROS2, asegurando que estés familiarizado con los nodos, tópicos y paquetes necesarios antes de abordar tareas avanzadas de este documento.

El objetivo práctico a desarrollar a partir de la **grabación de un skidpad (rosbag)** es procesar la información bruta de los sensores hasta tener una trayectoria. Nuestra percepción está basada en la nube de puntos que nos llegan desde nuestro LiDAR, el cual tiene 288.000 puntos por iteración, 192 canales y 120° de campo de visión horizontal y 20° de vertical.

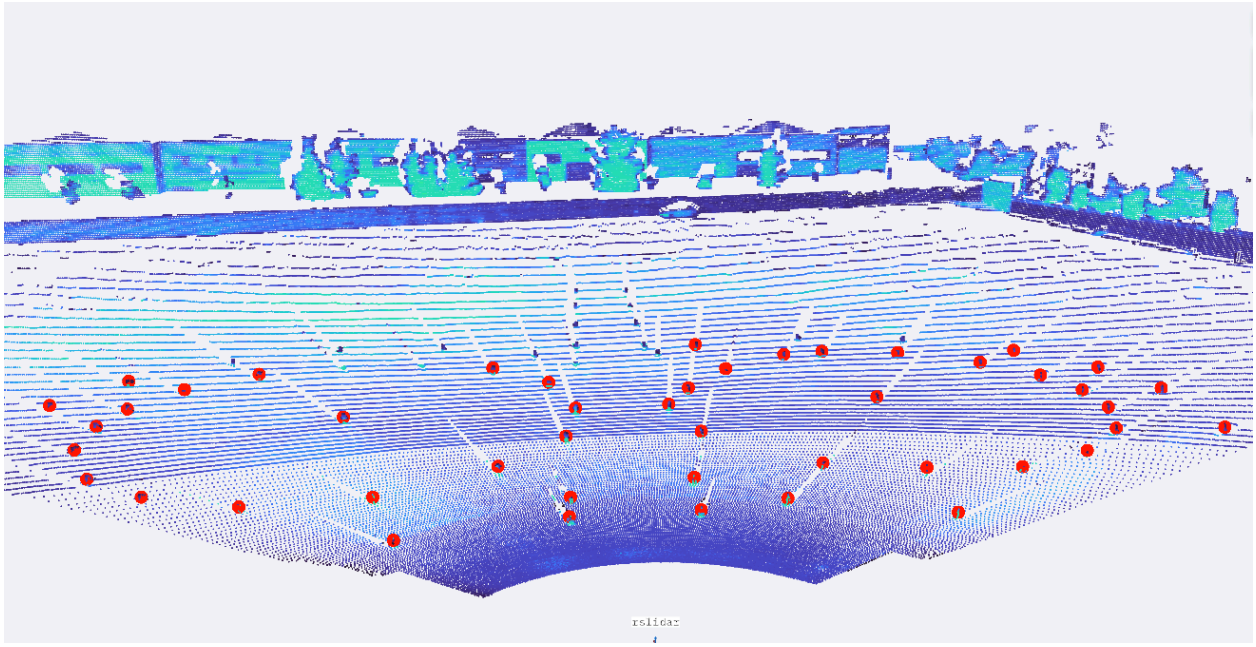
Para facilitar la inspección visual de los datos, la depuración de algoritmos y el análisis del comportamiento de la percepción, utilizaremos Foxglove Studio. Crea una cuenta e importa la grabación del link anterior desde recording -> import data. Pulsa visualize para abrirlo. Para no tener que configurar manualmente los paneles de visualización, puedes importar un archivo de diseño preconfigurado. Solo debes hacer clic en el menú desplegable de layouts en la esquina superior, seleccionar "Import from file..." y cargar el archivo .json del layout proporcionado. Para ver la parte de percepción pulsa en el recuadro de arriba a la izquierda y cambia el frame de arussim/world a rslidar y dale al icono del ojito de perception/map y rslidar_points.

Para analizar cualquier cambio:

- **Reproduce** el archivo de la grabación del *skidpad* utilizando el comando estándar de ROS2 (ros2 bag play skidpad_academy.mcap).
- **Visualización en Foxglove:** Puedes conectar Foxglove directamente a tus nodos en ejecución mediante el **Foxglove Bridge** o abriendo el fichero de grabación el comando para grabar es: ros2 bag record -s mcap -a
- **Precaución con los tópicos:** A la hora de publicar tus resultados procesados, **asegúrate de no nombrar a tus nuevos tópicos igual que los originales** (por ejemplo, añade un sufijo con tu nombre o /carmen/perception/map) para evitar colisiones.

PARTE I · PERCEPCIÓN

A continuación, puedes observar la comparativa entre la nube de puntos en bruto que recibe el sensor y la salida procesada por nuestro sistema de percepción:



Para lograr esta transformación, el subsistema de percepción ejecuta secuencialmente las siguientes etapas:

- **Ground Filter:** El ground filter como bien dice su nombre se dedica a eliminar el suelo de la nube de puntos para dejar solo los puntos que representan objetos (como conos, paredes...).
- **Clustering:** El *clustering* es un algoritmo el cual agrupa datos (en este caso puntos) mediante algún criterio como puede ser la distancia euclídea.
- **Filtrado de clustering:** Como el clustering lo que hace es sacar clusters de absolutamente todo, tenemos que ver lo que es un cono y lo que puede ser otra cosa como una persona o una mochila. Para ello nos basamos en el tamaño y forma:

			
big orange cone	small orange cone	small yellow cone	small blue cone
two white stripes	single white stripe	single black stripe	single white stripe
WEMAS 307.610500.00.00	WEMAS 400.000013.00.00	WEMAS 400.000013.01.10	WEMAS 400.000043.00.00
285 mm × 285 mm × 505 mm	228 mm × 228 mm × 325 mm		
1.05 kg	0.45 kg		

El objetivo es que investigues distintos ground filter, clustering y filtrado y te quedes con el que mejores resultados te dé de manera justificada. Además, se pueden añadir más funciones para mejorar la percepción como deskewing para tener en cuenta el movimiento o cropping y clipping.

PARTE II · SLAM

Una vez que el sistema de percepción ha extraído la posición de los conos, nos enfrentamos a uno de los mayores retos de la robótica autónoma: el SLAM (Simultaneous Localization And Mapping). Su objetivo principal es resolver el problema de construir un mapa del circuito mientras el monoplaza se localiza a sí mismo dentro de él en tiempo real. En la conducción autónoma, los sensores de estimación de movimiento acumulan pequeños errores con el paso del tiempo (***drift***). Si confiamos únicamente en la odometría, el mapa se deformará rápidamente y el coche perderá su referencia espacial. Por ello, es necesario utilizar la información de los conos detectados por la percepción para corregir tanto la trayectoria del vehículo como la posición estimada de los elementos de la pista.

Dado que existen múltiples enfoques para resolver el SLAM, te retamos a **investigar, proponer y justificar** el método que consideres más adecuado. Tu diseño debe contemplar al menos los siguientes aspectos clave:

- **Estrategia de Localización y Mapeo:** Investiga qué alternativas existen para actualizar la posición del coche y la de los conos simultáneamente, valorando sus ventajas e inconvenientes en términos de coste computacional y robustez.
- **Asociación de Datos (*Data Association*):** Cuando el coche avanza y vuelve a percibir los mismos conos, el sistema debe ser capaz de reconocerlos para evitar duplicarlos indefinidamente en el mapa. Deberás investigar y aplicar un criterio matemático o probabilístico para emparejar las nuevas observaciones con los conos ya mapeados.
- **Contador de vueltas:** Al tratarse de un circuito cerrado en forma de ocho, el sistema debe ser capaz de detectar cuándo se completa una vuelta.

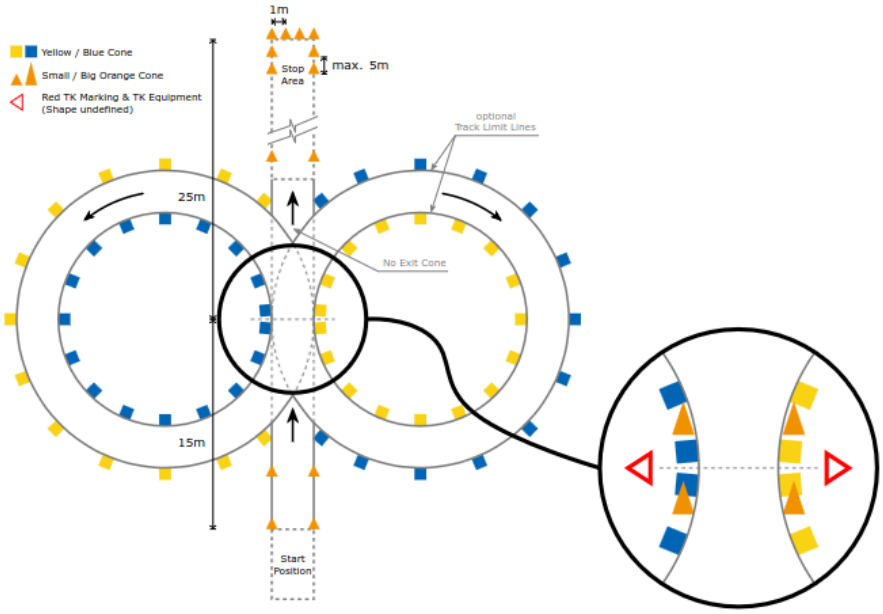
Una vez que el sistema de percepción ha detectado los conos y se han ubicado los dos círculos del circuito, el subsistema de **Planning** se encarga de decidir qué ruta exacta debe seguir el monoplaza y a qué velocidad debe recorrerla.

PARTE III · Planning

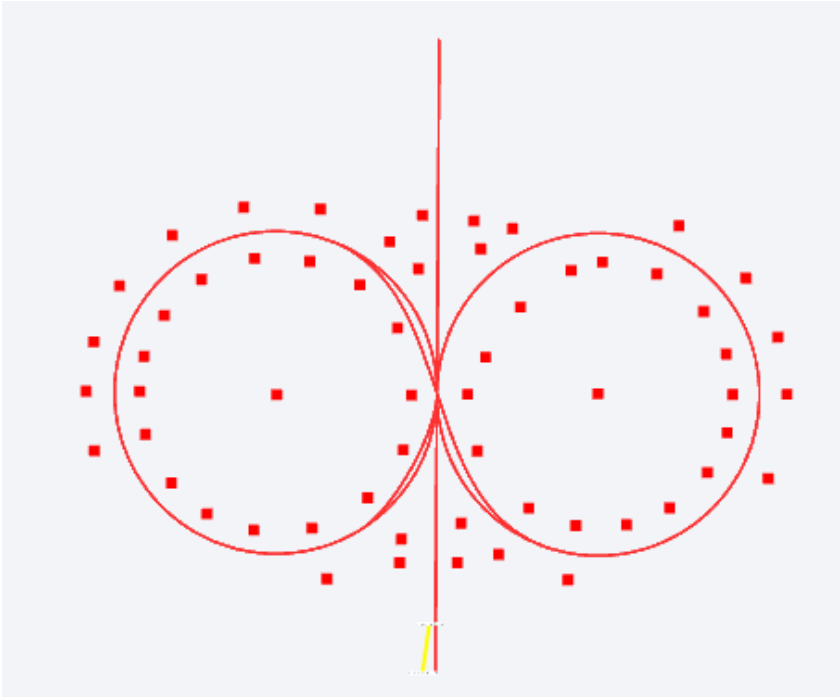
La prueba de Skidpad consiste en recorrer un circuito en forma de ocho (empezando por la derecha dos veces y luego dos a la izquierda, cronometrando solo las segundas vueltas). Dado que somos capaces de ver prácticamente todo el circuito antes de empezar y cualquier mínimo cambio de ruta perjudica directamente al control, el algoritmo debe generar una trayectoria estricta y perfiles de velocidad adaptados a cada fase de la prueba.

Te retamos a diseñar la base metodológica del planning para la prueba de Skidpad, investigando y resolviendo los siguientes aspectos clave:

- **Aprovechamiento de la Normativa:** Juega con todo lo que sabes sobre la normativa de la competición para deducir las dimensiones, radios y restricciones espaciales del circuito.



- **Generación y Ajuste de la Trayectoria:** A partir de la ubicación de los centros de los círculos del ocho, investiga cómo trazar la línea central ideal y aplicar una plantilla de ruta punto a punto que se ajuste exactamente a la geometría reglamentaria de la prueba. En la foto hay curvas de Bézier para optimizar el giro, pero no es necesario puedes hacer círculos perfectos:



- **Generación del perfil de velocidad:** Dado que la evaluación práctica del perfil de velocidad requiere de un simulador dinámico completo, en este apartado realizaremos una evaluación teórica de la estrategia de velocidad para exprimir al máximo el rendimiento del vehículo en la prueba de skidpad teniendo en cuenta que sólo se evalúan las segundas vueltas de cada lado.

PARTE IV · Control

Una vez generada la trayectoria, el sistema de Control es el encargado de calcular y enviar en tiempo real las órdenes exactas de dirección (ángulo de volante) y aceleración/frenada a los actuadores del coche para que este siga la referencia marcada sin desviarse.

Dado que el diseño del control define el comportamiento dinámico del coche frente a las órdenes del planning, te retamos a investigar y proponer soluciones (aunque, al igual que ocurre con el perfil de velocidad, no se pueda evaluar o refutar empíricamente sin un simulador)para los siguientes aspectos:

- **Control Lateral**: Investiga las diferencias y aplicaciones de los algoritmos de control geométrico y de optimización más utilizados en automoción. Justifica cuál se adaptaría mejor para el evento del skidpad.
- **Control Longitudinal**: Investiga cómo relacionar la velocidad objetivo marcada por el planning con la entrega de par en los motores que puede ser positiva (para acelerar) o negativa (para frenar), asegurando transiciones suaves.