

Caso de Estudio:

self-adaptive AutonomousCar:
un vehículo autónomo con capacidades
de auto-adaptación

Parte 1 : Descripción del Escenario y del
Prototipo Funcional

Diseño de Sistemas Ubicuos y Adaptativos
Máster MITSS
DSIC - UPV

Profesor:
Joan Fons i Cors
jjfons@dsic.upv.es

Valencia, Marzo de 2020

Control de versiones

Fecha	Autor	Descripción
20/03/2020	Joan Fons	Versión Inicial completa del documento

Índice de Contenidos

1 INTRODUCCIÓN	5
2 SISTEMA CONDUCCIÓN AUTÓNOMA (ADS)	6
2.1 NIVELES DE AUTONOMÍA	6
2.1.1 NIVEL 0 - CONDUCCIÓN MANUAL	8
2.1.2 NIVEL 1 - CONDUCCIÓN ASISTIDA	8
2.1.3 NIVEL 2 - AUTOMATIZACIÓN PARCIAL	8
2.1.4 NIVEL 3 - AUTOMATIZACIÓN CONDICIONAL	8
2.1.5 NIVEL 4 - AUTOMATIZACIÓN ALTA	9
2.1.6 NIVEL 5 - AUTOMATIZACIÓN COMPLETA	10
3 EL VEHÍCULO AUTÓNOMO	11
3.1 DISPOSITIVOS DE MONITORIZACIÓN Y CONTROL DEL VEHÍCULO	11
3.1.1 MOTOR (ENGINE)	11
3.1.2 DIRECCIÓN (STEERING)	11
3.1.3 VELOCÍMETRO (<i>SPEEDOMETER</i>)	11
3.1.4 SENsoRES DE CARRIL (<i>LINESENSOR</i>)	12
3.1.5 SENsoRES DE DISTANCIA (<i>DISTANCESENSOR</i>)	13
3.1.6 LIDAR	14
3.2 SERVICIOS DE CONDUCCIÓN AUTÓNOMA (ADS)	16
3.2.1 L0 - CONDUCCIÓN MANUAL (<i>L0_MANUALDRIVING</i>)	16
3.2.2 L1 - SERVICIO DE CONDUCCIÓN ASISTIDA (<i>L1_ASSISTEDDRIVING</i>)	16
3.2.3 L2 - CONTROL DE CRUCERO ADAPTATIVO (<i>L2_AdaptiveCruiseControl</i>)	17
3.2.4 L2 - ASISTENTE DE SEGUIMIENTO DE CARRIL (<i>L2_LaneKeepingAssist</i>)	17
3.2.5 L3 - CHÓFER EN AUTOVÍA (<i>L3_HighwayChauffer</i>)	18
3.2.6 L3 - CHÓFER EN AUTOVÍA PARA ATASCOS (<i>L3_TrafficJamChauffer</i>)	18
3.2.7 L3 - CHÓFER EN CIUDAD (<i>L3_CityChauffer</i>)	19
3.3 PLANES DE EMERGENCIA (FALLBACK PLAN)	19
3.3.1 APARCAR EN LA CUNETA (<i>PARKINTHEROADSOULDERFallbackPlan</i>)	19
3.3.2 EMERGENCY FALBACK PLAN	20
3.4 CONTEXTO DE CIRCULACIÓN	20
3.4.1 CONTEXTO DE LA CARRETERA	20
3.4.2 CONTEXTO DEL CONDUCTOR	21
3.5 INTERACCIÓN CON EL CONDUCTOR	21
3.5.1 ATENCIÓN Y MOLESTIA	21
3.5.2 SERVICIO DE NOTIFICACIONES EN EL VEHÍCULO	23
3.5.3 MECANISMOS DE INTERACCIÓN	23
4 DISEÑO DEL VEHÍCULO AUTÓNOMO	25
4.1 DISPOSITIVOS DEL VEHÍCULO	25
4.1.1 SPEEDOMETER	25
4.1.2 ENGINE	25
4.1.3 STEERING	25
4.1.4 LINESENSORS	25
4.1.5 DISTANCESENSORS	26
4.1.6 LIDAR-DISTANCESENSORS	26
4.1.7 ROADSENSOR	26
4.1.8 HUMANSENSORS	27
4.2 MECANISMOS DE INTERACCIÓN	27
4.2.1 NOTIFICATIONSERVICE	27
4.2.2 DRIVERDISPLAY	27
4.2.3 DASHBOARDDISPLAY	27
4.2.4 SPEAKERS	28

4.2.5	DASHBOARD	28
4.2.6	DRIVER & COPILOT SEAT	28
4.2.7	STEERING WHEEL	28
4.3	FUNCIONES DE CONDUCCIÓN AUTÓNOMA	28
4.3.1	L0 MANUAL DRIVING	28
4.3.2	L1 ASSISTED DRIVING	29
4.3.3	L2 ADAPTIVECRUISECONTROL	29
4.3.4	L2 LANEKEEPINGASSIST	29
4.3.5	L3 TRAFFICJAMCHAUFFFER	30
4.3.6	L3 HIGHWAYCHAUFFFER	30
4.3.7	L3 CITYCHAUFFFER	31
4.3.8	PARKINTHEROADSHOULDERFallbackPLAN	31
4.3.9	EMERGENCYFallbackPLAN	31
4.4	MODELO DE COMPONENTES	32
5	IMPLEMENTACIÓN DEL VEHÍCULO AUTÓNOMO	33
5.1	COMPONENTES DEL SISTEMA	33
5.1.1	AUTONOMOUSCAR.INTERFACES	34
5.1.2	AUTONOMOUSCAR.INFRASTRUCTURE	35
5.1.3	DEVICE.ENGINE	35
5.1.4	DRIVING.L2.ADAPTIVECRUISECONTROL	36
5.1.5	INTERACTION.DRIVERDISPLAY	38
5.2	CONFIGURACIÓN Y PUESTA EN MARCHA	39
5.2.1	CONFIGURACIÓN DE ARRANQUE OSGI	39
6	SIMULADOR	42
6.1	AUTONOMOUSCAR.SIMULATION.SIMULATOR	42
6.1.1	CONFIGURACIÓN DEL SIMULADOR	43
6.2	COMANDOS DE SIMULACIÓN PARA LA CONSOLA DE OSGI	45
6.3	EJEMPLOS DE EJECUCIÓN DE ESCENARIOS	48
7	REFLEXIÓN FINAL	58
8	BIBLIOGRAFÍA	59

Índice de Figuras

<i>Figura 1 : Niveles de Autonomía</i>	7
<i>Figura 2 : Vehículo autónomo</i>	11
<i>Figura 3 : Sensores de carril del vehículo</i>	12
<i>Figura 4 : Lecturas de los sensores de carril del vehículo</i>	12
<i>Figura 5 : Sensores de Distancia</i>	13
<i>Figura 6 : Funcionamiento de los sensores de distancia (dedicados) del vehículo</i>	13
<i>Figura 7 : LIDAR como sensor de Distancia perimetral</i>	14
<i>Figura 8 : Lecturas de distancias proporcionadas por el LIDAR</i>	15
<i>Figura 9 : Panel de Mandos del vehículo</i>	24
<i>Figura 10 : Modelo de Componentes del Vehículo Autónomo</i>	32
<i>Figura 11 : Proyectos (bundles) del vehículo</i>	34
<i>Figura 12 : Proyecto AutonomousCar.interfaces</i>	34
<i>Figura 13 : AutonomousCar.infrastructure</i>	35
<i>Figura 14 : Proyecto device.Engine</i>	35
<i>Figura 15 : Activator del device.Engine</i>	36
<i>Figura 16 : Proyecto driving.L2.AdaptiveCruiseControl</i>	36
<i>Figura 17 : Activator del plugin L2_AdaptiveCruiseControl</i>	36
<i>Figura 18 : Implementación de L2_AdaptiveCruiseControl</i>	37
<i>Figura 19 : Implementación de interaction.DriverDisplay</i>	38
<i>Figura 20 : Configuración de arranque OSGi del vehículo autónomo</i>	40
<i>Figura 21 : Configuración Avanzada de la Run Configuration</i>	40
<i>Figura 22 : Consola de OSGi tras la ejecución inicial</i>	41
<i>Figura 23 : Bundles activo en la configuración de arranque del vehículo</i>	41
<i>Figura 24 : Configuración del Simulador</i>	43
<i>Figura 25 : Ejemplo de Configuración de un Simulador temporizado</i>	44
<i>Figura 26 : Ejecución del Simulador Temporizado en modo verboso, con periodicidad 3 segundos</i>	44

1 Introducción

En este documento se puede encontrar la descripción del caso de estudio del vehículo autónomo que se usará en la asignatura de Sistemas Ubicuos y Adaptativos (SUA) del máster MITSS del DSIC, UPV.

El objetivo de este caso de estudio es proporcionar una solución común de base a los alumnos, sobre la que se definirán trabajos para aplicar los principios de computación autónoma vistos en la asignatura. Se puede considerar pues, que este sistema ofrece una implementación funcional, sobre la que se desarrollará el bucle de control MAPE-K.

El conocimiento descrito sobre el caso de estudio está sacado por el profesor inspirándose en lecturas sobre conducción (terrestre) autónoma [1] y [2]. Por motivos académicos, se han simplificado muchos escenarios y explicaciones, y se han tomado algunas licencias (menores), siempre respetando (e indicando en el texto) tales licencias.

En este informe se introducirá a la problemática y terminología de la conducción autónoma y los niveles de autonomía y se describirá un vehículo 'conceptual' con una serie de dispositivos y funciones de conducción. A continuación se propondrá un diseño para abordar la solución, y se proporciona detalles sobre la implementación y puesta en marcha (con un simulador) del vehículo. Por último, unas reflexiones finales ayudan a entender que lo presentado en este trabajo únicamente sirve como base para el trabajo que debe realizar el alumno exteniendo todo esto con capacidades de computación autónoma / auto-adaptación.

Este documento se complementa con la implementación de un prototipo funcional de vehículo autónomo construído a partir del diseño aquí descrito.

2 Sistema Conducción Autónoma (ADS)

El Sistema de Conducción Autónoma ó *Automated Driving System (ADS)* es el conjunto de dispositivos y servicios dentro de un vehículo que permiten controlar el vehículo. Está formado tanto por dispositivos físicos (sensores, actuadores y controladores) y por programas que se diseñan para colaborar en las tareas de realización de lo que se conoce como el *Dynamic Driving Service* [1].

Dos son las funciones que se consideran primarias para la conducción: el control del movimiento longitudinal (a través del motor), y el control del movimiento lateral (a través de la dirección).

A lo largo de los años, los sistemas ADS han ido evolucionando para incorporar cada vez más ayudas a la conducción y al soporte de estas funciones primarias: pasando de estados iniciales donde a través de sensores se advertía al conductor de situaciones de riesgo o posibles incidencias; a etapas intermedias en las que estos sistemas llegan a tener el control parcial del vehículo (asistente de cambio de carril, o asistente de frenada de emergencia); hasta llegar a lo que será la conducción autónoma del futuro en el que el ADS tendrá el control completo del vehículo. La previsión es que antes del 2030 tendremos vehículos completamente autónomos [2].

El ADS es un tipo de sistema autónomo aplicado a la conducción. Pero, como veremos en la siguiente sección, el concepto de autonomía tiene que ver con responsabilidades a la hora de monitorizar, decidir, controlar y actuar sobre un sistema, y realizar tareas de supervisión. Aunque el concepto es general (se puede aplicar a múltiples dominios como la robótica, la producción, la logística, etc.), en este documento nos centraremos en las implicaciones en el dominio de la conducción (terrestre).

2.1 Niveles de Autonomía

Los procesos humanos suelen empezar por realizar tareas manuales. Con el tiempo, se construyen herramientas que asisten y facilitan la tarea al humano. Con el progreso, estas herramientas se mejoran y pueden empezar a realizar ciertas tareas bajo ciertas circunstancias por el humano, dejando que éste descargue parte de su trabajo en estas herramientas. Siguiendo con la progresión y los avances tecnológicos, se puede llegar a desarrollar herramientas que lleguen a hacer todas las tareas, y el humano pase a un rol de supervisor y/o controlador, que verifica que las decisiones y acciones tomadas proactivamente por la herramienta, corrigiendo si hace falta o en ciertas situaciones críticas. En el nivel máximo de madurez de la herramienta, puede ya no requerir la intervención del humano bajo ninguna situación.

Este podría considerarse un ejemplo de lo que, desde el punto de vista de la autonomía de los sistemas se conoce como el problema de la *transferencia del control*. Éste se halla

inicialmente en el humano, y poco a poco se desarrollan sistemas y soluciones que permite recibir este control, parcial o totalmente.

En el ámbito de la conducción y los sistemas ADS es lo que estamos viendo. Hace 50 años, la conducción se realizaba únicamente por el conductor. Hace 30 años teníamos dispositivos sensores que nos advertían de problemas. Hace 10 años que tenemos servicios que permiten gestionar alguna de las funciones primarias de la conducción, en entornos controlados. Empiezan a ofrecer control sobre funciones secundarias, como el aparcamiento asistido o autónomo. Hoy tenemos ya vehículos que permiten controlar las dos funciones primarias, también en entornos controlados, y que requieren asistencia del humano en situaciones conflictivas. En un futuro cercano tendremos vehículos que puedan realizar estas funciones primarias y ya no requerirán de la participación del humano.

Si observamos la Figura 1 podemos ver una tabla [2] con una propuesta de esta transferencia de control (en forma de monitorización, decisión y actuación), y quién es el responsable de realizar dicha tarea (conductor o ADS).

Level	Name	Narrative definition	Execution of steering and acceleration/deceleration	Monitoring of driving environment	Fallback performance of dynamic driving task	System capability (driving modes)	SAE Level	NHTSA Level
<i>Human driver monitors the driving environment</i>								
0	No Automation	the full-time performance by the <i>human driver</i> of all aspects of the <i>dynamic driving task</i> , even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a	Driver only	0
1	Driver Assistance	the <i>driving mode-specific execution</i> by a <i>driver assistance system</i> of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the <i>human driver</i> perform all remaining aspects of the <i>dynamic driving task</i>	Human driver and system	Human driver	Human driver	Some driving modes	Assisted	1
2	Partial Automation	the <i>driving mode-specific execution</i> by one or more <i>driver assistance systems</i> of both steering and acceleration/deceleration using information about the driving environment and with the expectation that the <i>human driver</i> perform all remaining aspects of the <i>dynamic driving task</i>	System	Human driver	Human driver	Some driving modes	Partially automated	2
<i>Automated driving system ("system") monitors the driving environment</i>								
3	Conditional Automation	the <i>driving mode-specific performance</i> by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> with the expectation that the <i>human driver</i> will respond appropriately to a <i>request to intervene</i>	System	System	Human driver	Some driving modes	Highly automated	3
4	High Automation	the <i>driving mode-specific performance</i> by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> , even if a <i>human driver</i> does not respond appropriately to a <i>request to intervene</i>	System	System	System	Some driving modes	Fully automated	4
5	Full Automation	the full-time performance by an <i>automated driving system</i> of all aspects of the <i>dynamic driving task</i> under all roadway and environmental conditions that can be managed by a <i>human driver</i>	System	System	System	All driving modes	-	5

Figura 1 : Niveles de Autonomía

A menor nivel de autonomía, mayor control control de todos los sistemas y tareas por parte del conductor, y el ADS ofrece asistencia. A mayor nivel de autonomía, más control por el ADS y menos asistencia requerida por parte del conductor.

A continuación se repasan cada uno de los 6 niveles propuestos.

2.1.1 Nivel 0 - Conducción Manual

El conductor es responsable de todas las tareas (primarias y secundarias) de conducción del vehículo. Desde monitorizar el entorno, decidir qué acciones correctivas tomar en base a eventos que ocurran, o actuar sobre los dispositivos de control del vehículo.

2.1.2 Nivel 1 - Conducción Asistida

El conductor sigue siendo responsable de las tareas (primarias y secundarias) de conducción. Los vehículos empiezan a tener dispositivos y sensores que aumentan las capacidades de monitorización del vehículo. Le advierten de peligros (sensor de ángulo muerto, distancia trasera para aparcar, etc.) pero no puede tomar el control de ninguna tarea, primaria o secundaria de conducción.

Pueden aparecer servicios como el control de crucero, pero en forma de mecanismo que únicamente mantiene la velocidad, sin realizar corrección alguna (es el conductor el responsable de mantener el control en todo momento).

2.1.3 Nivel 2 - Automatización Parcial

El conductor sigue siendo el responsable de las tareas (primarias y secundarias) de conducción. Sin embargo, en entornos controlados, el ADS puede tomar el control (parcial) para realizar una tarea primaria, e incluso secundarias. El ADS no puede, en este nivel, encargarse de las dos funciones primarias de conducción a la vez.

Aparecen servicios que permiten gestionar el movimiento longitudinal, y que son capaces de mantener una distancia de seguridad preconfigurada frente a obstáculos (otros vehículos) existentes en la carretera. Otros, permiten el control del movimiento lateral, y mantienen el vehículo dentro del carril.

Ambos servicios, aunque ofrecen automatización parcial, no relegan la responsabilidad de conducción al humano. Aunque sí que permiten que este 'relaje' la monitorización de ciertos parámetros (distancia o líneas de carril) y que el ADS le asista.

Aparecen también servicios que asisten parcialmente a maniobras secundarias. Por ejemplo, el asistente de parking en el que de manera colaborativa, el conductor controla el movimiento longitudinal (acelerador y freno) y el vehículo controla el lateral (dirección).

2.1.4 Nivel 3 - Automatización Condicional

En este nivel ya empieza a haber una transferencia de control relevante, bajo parámetros funcionales concretos (por ejemplo, en cierto tipo de carreteras, hasta un límite de velocidad, o según condiciones climáticas). El humano puede delegar las funciones de

conducción primarias y secundarias al ADS, si se cumplen estas condiciones, y temporalmente no ser responsable de estas funciones (ni de monitorizarlas).

Estas funciones introducen un problema que hay que atender correctamente. A pesar de que el humano no sea responsable de realizar temporalmente una función, en este nivel de autonomía el ADS puede encontrarse ante situaciones que le impidan seguir su función autónoma. Esto genera un conflicto que se afronta de varias maneras:

- 1) intentando que el conductor no 'desconecte del todo' de las funciones de conducción, y mantenga cierto nivel de atención;
- 2) estableciendo un proceso de transferencia de control (al revés) de manera que el ADS pase el control al humano (siempre que éste esté atento a la conducción);
- 3) definiendo procedimientos de emergencia para que, en el caso que cierta función de conducción autónoma no pueda seguir realizándose por el ADS y que el humano no esté preparado, el ADS pueda activar dicho procedimiento para que deje el vehículo en una situación de mínimo riesgo posible (que dependerá de la situación).

El punto 1) nos lleva a pensar en introducir mecanismos orientados a mantener esta atención del conductor. Así, al ADS se le debe otorgar una capacidad más: monitorizar al humano/conductor y mantenerlo informado y preparado, en función de las posibles necesidades.

El punto 2) es lo que se conoce como el problema del '*TakeOver*'. Es el proceso de recuperación del control del vehículo por parte del conductor, pero que es el propio sistema ADS quien decide cederlo (debido a una situación/contexto de conducción) para la que no está preparado. Sólo se puede realizar este *TakeOver* si el conductor está preparado y es consciente de que esto va a ocurrir. En caso contrario, se podrían generar situaciones críticas.

El punto 3) nos lleva al concepto de *Plan de Emergencia* o *Fallback Plan*. Está relacionado con el problema de la transferencia de control del ADS al conductor, pero cuando el conductor no puede, no está atento/preparado o no es consciente de debe tomar el control. En dicha situación, el ADS, cuando ya no pueda seguir con las funciones de conducción autónoma, activará dicho plan.

2.1.5 Nivel 4 - Automatización Alta

El nivel 4 es en realidad una mejora del nivel 3, reduciendo los contextos donde las funciones autónomas no puedan operar, y mejorando las existentes. Lo que se consigue con este nivel es que el vehículo pueda estar más tiempo en situaciones de conducción autónoma, y que requiera en menos situaciones al humano.

Sin embargo, siguen siendo necesarias las transferencias de control hacia el conductor (*TakeOver*) y los planes de emergencia (igual que en el nivel 3).

A nivel conceptual no aporta un cambio significativo sobre el nivel anterior. A nivel práctico, estos vehículos van a permitir que la conducción sea prácticamente autónoma en la mayoría de las situaciones.

2.1.6 Nivel 5 - Automatización Completa

Es el máximo nivel de autonomía, y como indica, en este nivel el ADS no requiere al conductor bajo ninguna circunstancia. Esto implica que sería factible incluso el envío de vehículos 'sin conductor', pudiendo darse tanto sin pasajeros en el vehículo, o con pasajeros sin la habilidad de conducción (niños, adultos sin carnet, etc.)

Este nivel no sufre del problema del TakeOver ni requiere Planes de Emergencia. Sin embargo, por un tema de confianza en los sistemas autónomos, seguirá siendo necesario mantener al humano/conductor informado de las principales tareas, decisiones y acciones tomadas por el ADS.

3 El Vehículo Autónomo

3.1 Dispositivos de monitorización y control del Vehículo

Para monitorizar y controlar todas las funciones, parámetros y contexto de conducción, es necesario dotar al vehículo de diferentes dispositivos. Para ilustrar el vehículo y estos dispositivos, usaremos las siguientes perspectivas del vehículo:



Figura 2 : Vehículo autónomo

3.1.1 Motor (Engine)

De combustión o eléctrico, es el principal encargado del movimiento longitudinal. Se controla principalmente a través de las RPM a las que se establezca (aceleración/deceleración), provocando que, a través del sistema de cambio (no incluido en este caso de estudio), se convierta en este movimiento longitudinal (adelante o atrás).



3.1.2 Dirección (Steering)



Controla el movimiento lateral del vehículo. Típicamente, a través del volante (*steering wheel*) se puede alterar el sistema de dirección del vehículo. La dirección ofrece transmitir movimientos laterales a la derecha, a la izquierda o centrar la dirección.

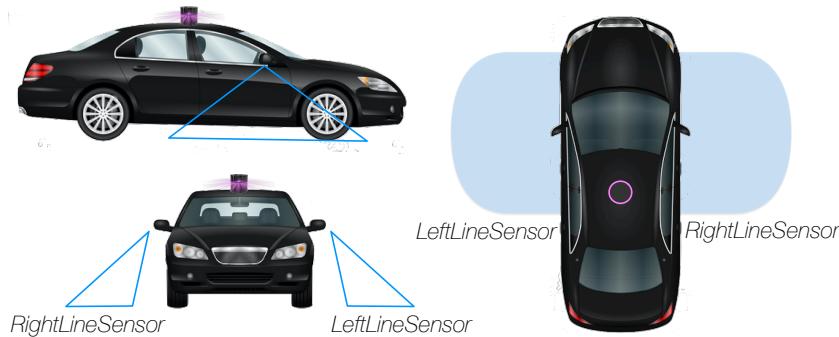
3.1.3 Velocímetro (Speedometer)

Sensor que mide la velocidad del vehículo (en Km/h). Habitualmente ubica sensores en las ruedas y permite calcular, en función del diámetro de las mismas y de un contador de vueltas, la velocidad instantánea. Suele ir acompañado de un odómetro (o medidor de distancia), aunque en este caso de estudio no se ha tenido en cuenta este otro dispositivo.



3.1.4 Sensores de Carril (*LineSensor*)

Sensor (óptico) capaz de detectar las marcas viales longitudinales que delimitan los carriles de circulación. Se ubica un sensor en cada lateral del vehículo.



Su uso permite realizar funciones de corrección de la trazada, seguimiento en adelantamientos, o para realizar maniobras de parada en el arcén (por ejemplo).

Figura 3 : Sensores de carril del vehículo

La siguiente imagen muestra varias situaciones donde los sensores reportan diferentes valores en función de la línea de carril que se 'cruce'.

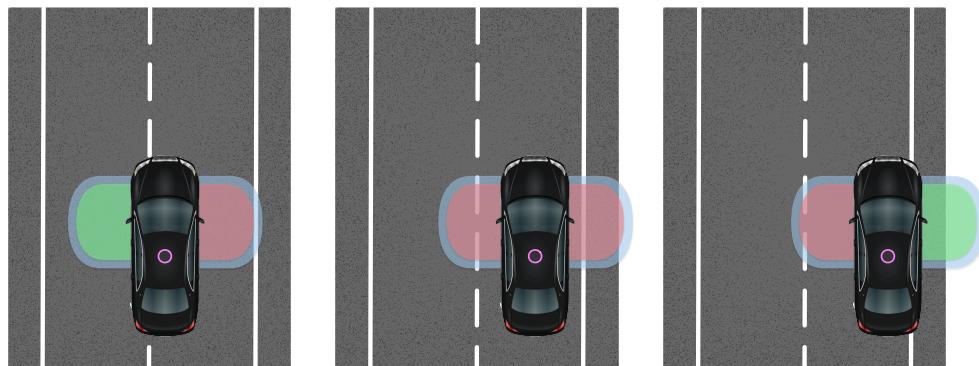


Figura 4 : Lecturas de los sensores de carril del vehículo

3.1.5 Sensores de Distancia (*DistanceSensor*)

Basados en tecnologías de ultrasonidos, ultrafrecuencias o ópticos (laser), entre otras, permiten medir distancias a obstáculos cercanos. Puede tener alcance de hasta varios cientos de metros.

Este vehículo dispone de 4 de estos sensores (

Figura 5):

- **sensor distancia frontal** (*FrontDistanceSensor*)
- **sensor distancia trasero** (*RearDistanceSensor*)
- **sensor distancia lateral derecho** (*RightDistanceSensor*)
- **sensor distancia lateral izquierdo** (*LeftDistanceSensor*)

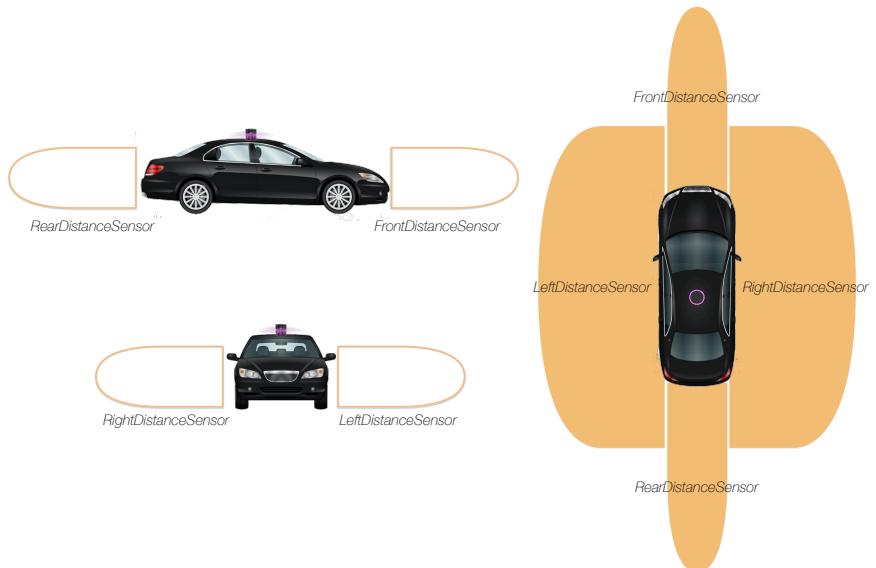


Figura 5 : Sensores de Distancia

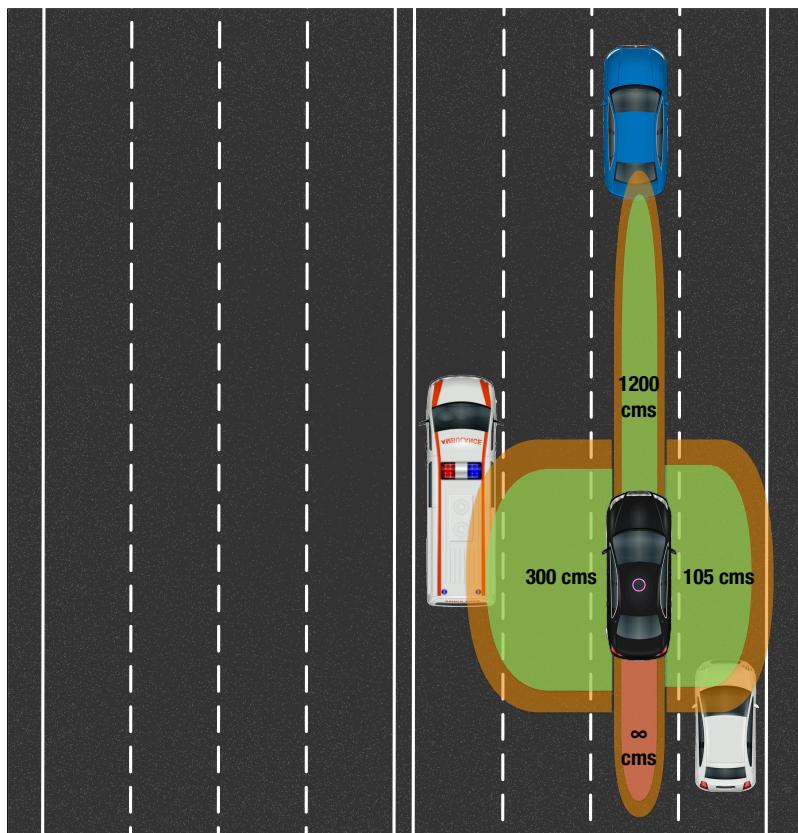


Figura 6 : Funcionamiento de los sensores de distancia (dedicados) del vehículo

La Figura 6 muestra un ejemplo de valores que pueden reportar estos sensores. Como se puede apreciar, 3 de los sensores están identificando distancias hasta los obstáculos más cercanos, mientras que el sensor de distancia trasera nos informa (∞ cms) que no se detecta obstáculo en el rango de detección del sensor (varios cientos de metros).

3.1.6 LIDAR

Es una tecnología óptica de teledetección que mide la distancia (omnidireccional) desde un punto emisor a cualquier objeto o superficie mediante el uso de un láser. Su funcionamiento es parecido al de un radar, y puede llegar a realizar mediciones de varios cientos de metros.

En el vehículo de este caso de estudio usaremos el LIDAR convirtiendo estas distancias omnidireccionales en distancias 4 direccionales (frontal, trasera, izquierda y derecha). La Figura 7 muestra el rango de trabajo del LIDAR y la división en estas 4 distancias definidas.

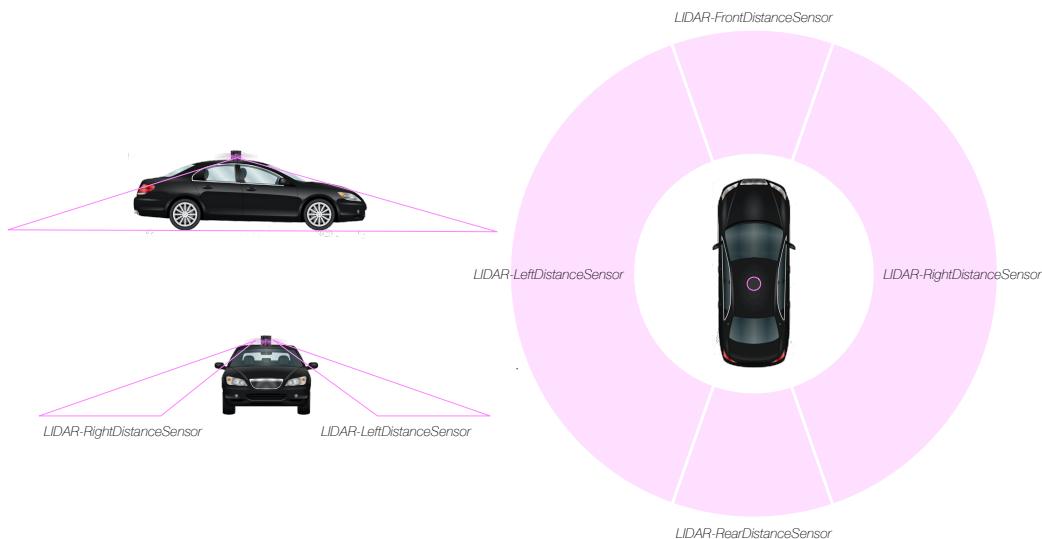


Figura 7 : LIDAR como sensor de Distancia perimetral

Este sensor LIDAR ofrece pues una duplicidad de los sensores de distancia perimetales, tanto longitudinal (frontal y trasero) como laterales (izquierda y derecha). Este hecho permitirá intercambiar, en caso necesario, los sensores de distancia dedicados (`DistanceSensor`) y utilizar los del LIDAR, permitiendo tener un mecanismo de 'backup' en caso de fallo o de lecturas erróneas/impresas. Así pues, consideraremos estos sensores como alternativas viables en caso de que los `DistanceSensor` fallen.

Sin embargo, dada la naturaleza y disposición de este LIDAR (zona central superior del vehículo), junto a la tecnología usada, vamos a asumir (para este caso de estudio) que las lecturas que nos ofrece son de menor precisión que las ofrecidas por los sensores dedicados en cada dirección específica (los `DistanceSensor`). Esto significa que, en la medida de lo posible, debemos intentar usar los `DistanceSensor` al usar las funciones de conducción autónoma, en vez de los sensores que nos proporciona el LIDAR.

La Figura 8 muestra un ejemplo de valores que pueden reportar estos sensores, en una situación similar a la mostrada en el caso de los sensores de distancia (ver Figura 6).

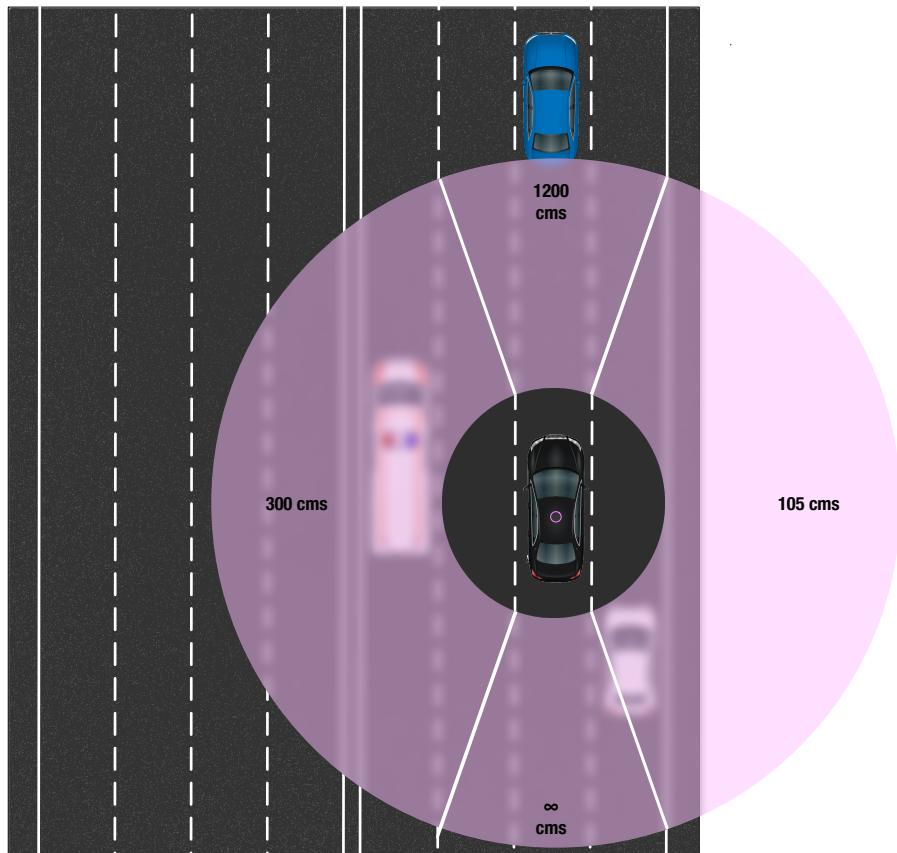


Figura 8 : Lecturas de distancias proporcionadas por el LIDAR

3.2 Servicios de Conducción Autónoma (ADS)

Siguiendo el razonamiento planteado en la secciones 2 y 2.1, a continuación se describen un conjunto de funciones de conducción autónoma que el ADS del vehículo que queremos diseñar debe tener.

Para cada una de estas funciones se indicará su nivel de autonomía, las restricciones sobre el contexto operativo que pudiera tener, y si requieren de *TakeOver* o *Fallback Plan*. También se definirá qué tipo de mensajes debe enviar al humano/conductor, tanto para mantener su atención (si así se requiere), como para informar de lo que está ocurriendo en el ámbito de la conducción.

Por último comentar que, debido a restricciones de tiempo, nos centraremos en abordar algunas funciones de conducción autónoma para los niveles L0, L1, L2 y L3. Hay que ser consciente que un vehículo autónomo ofrecerá más de estas funciones en la realidad, y que el uso y definición de las actuales persiguen objetivos académicos.

3.2.1 L0 - Conducción Manual (*L0_ManualDriving*)

Nivel de Autonomía	0
Nombre del Servicio	L0_ManualDriving
Responsabilidad ADS	<ninguna>
Responsabilidad Conductor	realizar las tareas de conducción completas, tanto primarias como secundarias
Restricciones operativas	<ninguna>
Requiere	<nada>
Notificaciones	<no genera>
Plan de Emergencia	<no procede>

3.2.2 L1 - Servicio de Conducción Asistida (*L1_AssistedDriving*)

Nivel de Autonomía	1
Nombre del Servicio	L1_AssistedDriving
Responsabilidad ADS	monitorizar algunos parámetros funcionales y avisar al conductor de ciertas situaciones de potencial peligro
Responsabilidad Conductor	realizar las tareas de conducción completas, tanto primarias como secundarias
Restricciones operativas	<ninguna>
Requiere	sensores del vehículo para monitorizar el contexto de la conducción
Notificaciones	genera notificaciones al conductor en forma, típicamente de Iconos Visuales en las consolas, 'beeps' o vibraciones.
Plan de Emergencia	<no procede>

3.2.3 L2 - Control de Crucero Adaptativo (*L2_AdaptiveCruiseControl*)

Nivel de Autonomía	2
Nombre del Servicio	L2_AdaptiveCruiseControl
Responsabilidad ADS	controlar parcialmente (en combinación con el conductor) el movimiento longitudinal, actuando sobre el sistema de frenado (motor) para mantener una distancia de seguridad frontal pre-establecida
Responsabilidad Conductor	realizar las tareas de conducción completas, tanto primarias como secundarias. Recibe asistencia en la función primaria relacionada con el control del movimiento longitudinal
Restricciones operativas	<ninguna>
Requiere	sensor de distancia frontal, acceso al control del motor y que se le indique un valor de distancia de seguridad longitudinal
Notificaciones	mostrar avisos sobre las acciones de corrección tomadas, típicamente en forma de iconos visuales o vibración en el volante
Plan de Emergencia	<no procede>

3.2.4 L2 - Asistente de Seguimiento de Carril (*L2_LaneKeepingAssist*)

Nivel de Autonomía	2
Nombre del Servicio	L2_LaneKeepingAssist
Responsabilidad ADS	controlar parcialmente (en combinación con el conductor) el movimiento lateral, actuando sobre el sistema de dirección para mantener el vehículo dentro del carril
Responsabilidad Conductor	realizar las tareas de conducción completas, tanto primarias como secundarias. Recibe asistencia en la función primaria relacionada con el control del movimiento lateral
Restricciones operativas	<ninguna>
Requiere	sensores de carril izquierdo y derecho, y acceso al control de la dirección
Notificaciones	mostrar avisos sobre las acciones de corrección tomadas, típicamente en forma de iconos visuales o vibración en el volante
Plan de Emergencia	<no procede>

3.2.5 L3 - Chófer en Autovía (*L3_HighwayChauffer*)

Nivel de Autonomía	3
Nombre del Servicio	L3_HighwayChauffer
Responsabilidad ADS	realizar tareas de conducción de las funciones primarias de movimiento longitudinal y lateral
Responsabilidad Conductor	realizar las tareas de conducción completas, tanto primarias como secundarias. Puede delegar las funciones primarias al ADS
Restricciones operativas	la carretera debe ser una vía rápida (<i>highway</i>) bien delimitada (marcas viales), y con tráfico fluído
Requiere	acceso al control del motor y la dirección, monitorización de la carretera y del conductor, acceso a sensores de carril, y distancias perimetrales (frontal/trasera, izquierda/derecha)
Notificaciones	mostrar avisos y proporcionando explicaciones sobre las acciones de corrección tomadas. Además debe mantener la atención del humano a la tarea de conducción para realizar el <i>TakeOver</i> si es necesario
Plan de Emergencia	debe indicarse un plan de emergencia

3.2.6 L3 - Chófer en Autovía para Atascos (*L3_TrafficJamChauffer*)

Nivel de Autonomía	3
Nombre del Servicio	L3_TrafficJamChauffer
Responsabilidad ADS	realizar tareas de conducción de las funciones primarias de movimiento longitudinal y lateral
Responsabilidad Conductor	realizar las tareas de conducción completas, tanto primarias como secundarias. Puede delegar las funciones primarias al ADS
Restricciones operativas	la carretera debe ser vía rápida (<i>highway</i>) bien delimitada (marcas viales), y con tráfico en situación de atasco o congestión. No puede superar la velocidad de 60 Km/h
Requiere	acceso al control del motor y la dirección, monitorización de la carretera y del conductor, acceso a sensores de carril, y distancias perimetrales (frontal/trasera, izquierda/derecha)
Notificaciones	mostrar avisos y proporcionando explicaciones sobre las acciones de corrección tomadas. Además debe mantener la atención del humano a la tarea de conducción para realizar el <i>TakeOver</i> si es necesario
Plan de Emergencia	debe indicarse un plan de emergencia

3.2.7 L3 - Chófer en Ciudad (*L3_CityChauffer*)

Este modo no existe en realidad ni en el [1] ni en [2] pero se ha decidido introducir por motivos académicos.

Nivel de Autonomía	3
Nombre del Servicio	L3_CityChauffer
Responsabilidad ADS	realizar tareas de conducción de las funciones primarias de movimiento longitudinal y lateral
Responsabilidad Conductor	realizar las tareas de conducción completas, primarias y secundarias. Puede delegar funciones primarias al ADS
Restricciones operativas	la carretera debe ser una vía dentro de una ciudad (<i>city</i>) bien delimitada (marcas viales). No puede superar la velocidad de 50 Km/h
Requiere	acceso al control del motor y la dirección, monitorización de la carretera y del conductor, acceso a sensores de carril, y distancias perimetrales (frontal/trasera, izquierda/derecha)
Notificaciones	mostrar avisos y proporcionando explicaciones sobre las acciones de corrección tomadas. Además debe mantener la atención del humano a la tarea de conducción para realizar el <i>TakeOver</i> si es necesario
Plan de Emergencia	debe indicarse un plan de emergencia

3.3 Planes de Emergencia (Fallback Plan)

Los planes de emergencia sirven para atender a situaciones críticas cuando el ADS decide que debe devolver el control del vehículo al conductor (debido a una restricción operativa de la función autónoma), pero el conductor no está en disposición de tomar el control.

El objetivo de un plan de emergencia es resolver esta situación crítica y conducir el vehículo hasta lo que se llama en [1] como la *situación de mínimo riesgo*.

3.3.1 Aparcar en la Cuneta (ParkInTheRoadShoulderFallbackPlan)

Nombre del Plan	ParkInTheRoadShoulderFallbackPlan
Descripción	realiza una maniobra de emergencia para llevar al vehículo a la cuneta, lo detiene, y desactiva la función de conducción autónoma
Restricciones operativas	la carretera debe ser una vía rápida (<i>highway</i>) o normal (<i>std</i>)
Requiere	acceso al control del motor y la dirección, sensor de distancia lateral derecho, y sensor de carril derecho

3.3.2 Emergency Fallback Plan

Este plan se puede considerar como la última opción. Se usa cuando no se puede aplicar plan de aparcar en la cuneta (ParkInTheRoadShoulderFallbackPlan), por no estar en una vía rápida (highway) o normal (std), o porque no tenemos acceso a los sensores de distancia y carril derecho (que nos impediría realizar un movimiento lateral a la derecha con garantías).

Nombre del Plan	EmergencyFallbackPlan
Descripción	realiza una maniobra de emergencia parando el vehículo en el carril actual, y pone las luces de emergencia para advertir al resto de vehículos
Restricciones operativas	<ninguna>
Requiere	acceso al control del motor y la dirección

3.4 Contexto de Circulación

Para controlar el contexto (o situación) de conducción en función del nivel de autonomía, será necesario tener en cuenta parámetros relacionados con:

- la carretera
- el conductor

3.4.1 Contexto de la Carretera

El tipo de vía es importante para atender a los diferentes servicios de conducción. Algunos sólo están disponibles para ciertos tipos o situaciones de la vía.

Así, se requiere monitorizar los siguientes parámetros:

- **Tipo de Vía** (`RoadType`) : identificará los siguientes tipos de vía
 - `STD_ROAD` : vía pavimentada con delimitación de carriles y arcén 'pequeño', con límite de velocidad moderado (100 Km/h)
 - `OFF-ROAD` : vía no pavimentada o sin delimitación de carriles
 - `HIGHWAY` : vía `STD_ROAD` con arcén grande (y habitualmente con varios carriles en cada sentido) con límite de velocidad alto (120 Km/h)
 - `CITY` : vía dentro de una ciudad
- **Estado de la Vía** (`RoadStatus`) : la vía podrá tener circulación fluída (`FLUID`), en atasco (`JAM`) o colapsada (`COLLAPSED`).

3.4.2 Contexto del Conductor

Será necesario conocer la situación del conductor para poder adaptar los servicios de conducción a dicha situación, tratando de ofrecer las mejores condiciones de seguridad y confianza posible.

Así, se requiere monitorizar los siguientes parámetros:

- **Manos en el volante** (`HandsOnWheel`): el conductor tiene o no las manos en el volante
- **Posición de la Cara** (`FaceStatus`): mirando al frente (`LOOKING_FORWARD`), distraído mirando hacia abajo, a los lados o hacia atrás (`DISTRACTED`) o dormido (`SLEEPING`)
- **Asiento del Conductor ocupado** (`DriverSeatOccupied`): el asiento del conductor está (o no) ocupado
- **Asiento del Copiloto ocupado** (`CopilotSeatOccupied`): el asiento del copiloto está (o no) ocupado

3.5 Interacción con el Conductor

El vehículo descrito ofrece funciones de asistencia a la conducción con diferentes niveles de autonomía. Tal y como se ha comentado en la sección 2, este autonomía está directamente relacionada con el cambio de responsabilidad a la hora de realizar ciertas tareas, desde la monitorización, pasando por la toma de decisiones o el control efectivo del sistema.

A mayor autonomía, más tareas de este tipo son responsabilidad del propio sistema, y el humano juega un papel de supervisor (confirmando o denegando acciones), o incluso sólo de mero observador. Será necesario pues, mantener al conductor informado de lo que ocurre en el contexto de la conducción por si fuera necesario involucrarlo, o simplemente para proporcionarle '*feedback*' (información de contexto).

3.5.1 Atención y Molestia

En todo el momento el humano debe ser *consciente* de lo que está aconteciendo en un sistema autónomo. Si este sistema autónomo no le reporta información sobre los principales procesos, acciones o decisiones que están aconteciendo, el humano puede llegar a no confiar en el sistema autónomo, o dudar de si está funcionando correctamente.

Supongamos que estando activo el servicio de conducción *L3 Highway Chauffer* (en vía rápida fluida) el vehículo detecta que la distancia frontal (`FrontDistanceSensor`) al obstáculo más cercano es inferior a la distancia de seguridad longitudinal, y decide decelerar. Si no advierte de este hecho al conductor usando algún indicador, mensaje o vibración (del volante, por ejemplo), el conductor podría pensar que algo 'raro' está pasando. Así pues, aunque sea el

vehículo el que tenga el control, debe notificar de acciones al conductor para así mantenerle informado de las principales acciones y decisiones tomadas.

Especial cuidado debe tomarse en situaciones donde el sistema autónomo es incapaz de ofrecer una autonomía completa (nivel 5 en los vehículos, según se ha comentado) y requiere de la colaboración del humano para completar ciertas tareas. Así, por ejemplo, la conducción autónoma *L3 Highway Chauffer* sólo es viable en vías rápidas. Si el vehículo abandona una de estas vías, requiere que el humano tome el control y que sea consciente de lo que está pasando y de la importancia de dicha acción. Es por esto que el vehículo debe notificar, con un nivel de alerta adecuado (por ejemplo, usando los altavoces), que se requiere que tome el control del vehículo. A esta acción de pasar de modo autónomo (controlado por el vehículo) a modo manual (controlado por el humano) se le conoce como *TakeOver*.

Como conclusión acerca de los niveles de atención y molestia para interactuar con el conductor, podemos establecer una serie de reglas/heurísticas. Por supuesto, en un entorno real y para situaciones concretas, estas 'reglas' deberían definirse con mayor precisión. Para los efectos de este caso de estudio, sirvan las siguientes recomendaciones.

Si lo que queremos es transmitir información al conductor, un mensaje de texto (escrito o hablado) ofrece más información que un ícono, y éste más que una vibración.

Feedback Información : Texto Visual ó Audición Texto (*Text-To-Speech*) > Ícono > Vibración

Si lo que queremos es alertar de algo al usuario, un sonido provoca más atención que una vibración¹, y una vibración más que algo visual.

Molestia / Atención :

Audición > Vibración > Visual

En función pues del objetivo perseguido (captar la atención o proporcionar información) en función de la necesidad concreta, se usarán unos mecanismos u otros.

Así, por ejemplo, si el conductor está con las manos en el volante, mirando hacia la carretera (atento), para notificarle que se le va a pasar el control de la conducción (*TakeOver*), a pesar de ser una tarea crítica, es suficiente con mostrarle un ícono en la consola del conductor, posiblemente combinado con una vibración en el volante.

Si, en otras circunstancias, el conductor estuviera distraído (o incluso dormido), o sin tener las manos en el volante, la misma notificación de *TakeOver* requeriría usar unos mecanismos de interacción más molestos, por ejemplo, a través de una audición de texto, una vibración en el asiento y un mensaje en la consola.

¹ Asumimos que el conductor no tiene ningún tipo de 'minusvalía', como una invidencia (por motivos obvios) o sordera. En el futuro, con mayores niveles de autonomía, éstas no deberían suponer mayores problemas.

3.5.2 Servicio de Notificaciones en el vehículo

Para satisfacer estos requisitos de interacción con el humano con el fin de mantenerlo informado de las acciones tomadas por el sistema autónomo, se define uno (o varios, si se considera) servicios de Notificación que podrán usarlos las funciones de conducción (de asistencia o de autonomía) para reportar información relevante al conductor.

Este servicio de Notificación estará configurado para usar uno o varios mecanismos de interacción. Un mecanismo de interacción es un dispositivo del vehículo (los altavoces, una pantalla/consola o un volante con vibración) que permite transmitir un mensaje o una alerta al conductor. En función del dispositivo concreto y la manera en que muestre este mensaje, podremos conseguir diferentes niveles de 'molestia' que deberán usarse adecuadamente para responder a las necesidades concretas de la notificación.

Así, por ejemplo, si se quiere advertir de que el sensor de carril izquierdo se ha activado y el vehículo ha tomado la acción correctiva de girar el volante a la derecha, bastará mostrar un ícono en la consola del conductor, o incluso una simple vibración en el volante. Dado que la advertencia no es crítica y no se desea desarrollar un sistema excesivamente molesto, se usarán mensajes de este tipo para situaciones parecidas.

Sin embargo, en el caso descrito anteriormente del *TakeOver*, se deberán usar mecanismos de interacción más 'molestanos' o 'intrusivos', que capten mejor la atención del conductor. Incluso combinándolos, para maximizar su efectividad. Así, los altavoces, los mensajes de texto explicativos en la consola del conductor (o central), combinados con la vibración en el asiento del conductor pueden tener una efectividad mayor.

3.5.3 Mecanismos de interacción

Supongamos una configuración de un panel de mandos actual de un vehículo, como el que se puede observar en la siguiente figura.

En este panel de mandos podemos observar elementos con diferentes objetivos de interacción:

- Pantallas o consolas donde mostrar información y mensajes visuales al conductor
- Botones retroiluminados (a modo de 'iconos') donde indicar el estado de ciertos dispositivos o servicios (ej. ESP, Start&Stop, ABS, etc.)
- Altavoces para indicar instrucciones, o para generar 'beeps' y/o alertas
- Zonas hapticas (volante o asientos) donde transmitir alguna notificación en forma de vibración



Figura 9 : Panel de Mandos del vehículo

Para el vehículo que estamos describiendo en este caso de estudio, se plantea usar los siguientes mecanismos de interacción. Los números permiten identificarlos sobre la figura anterior.

1. Consola del Conductor (*Driver Display*) : permite mostrar mensajes de texto visuales, e mensajes de tipo 'ícono' (información para el conductor)
2. Consola Central (*Dashboard Display*) : permite mostrar mensajes de texto visuales, e mensajes de tipo 'ícono' (información para el conductor y los pasajeros)
3. Altavoces (*Speakers*) : reproducen alertas en forma de sonido (Text-to-Speech) o 'pitidos' (*beep*).
4. Asiento Conductor y de Copiloto (*Seats*) : transmiten un mensaje (de feedback o de alerta, según el caso) en forma de vibración
5. Volante (*Steering Wheel*) : transmiten un mensaje (de feedback o de alerta, según el caso) en forma de vibración
6. Botón/Ícono Salpicadero (*Dashboard Icon*) : muestra a través de un botón o un ícono, el estado de alguna función, servicio o dispositivo del vehículo

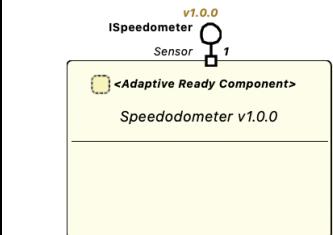
Con estos mecanismos disponibles, deberemos configurar las notificaciones que recibirán tanto el conductor como los pasajeros, de acuerdo a las necesidades antes descritas. Así pues, será posible, por ejemplo, mostrar mensajes de texto en la consola del conductor, o indicarle a través de algún ícono que algo ha sucedido, o podemos activar la vibración del asiento del conductor.

4 Diseño del Vehículo Autónomo

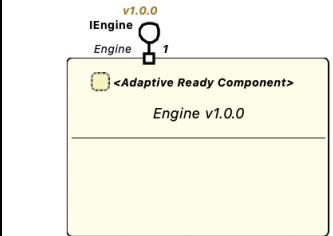
Siguiendo la notación propuesta en la asignatura [3], a continuación se propone un diseño de componentes para abordar el desarrollo del vehículo autónomo. Estos componentes se pueden agrupar según su función (dispositivos, interacción y funciones de conducción). Además de definir los componentes, es importante identificar las interfaces que ofrecen y que requieren de otros componentes para funcionar.

4.1 Dispositivos del Vehículo

4.1.1 Speedometer

	<p>Componente que ofrece la interfaz <code>ISpeedometer</code>, que permite obtener la velocidad actual</p>
--	---

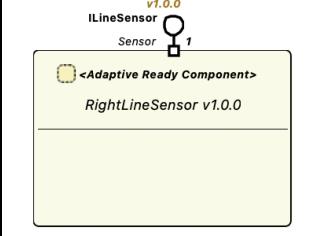
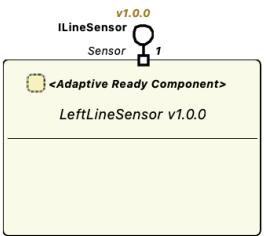
4.1.2 Engine

	<p>Componente que ofrece la interfaz <code>IEngine</code>, que permite controlar las RPM del motor (y con ellas la velocidad del vehículo)</p>
---	--

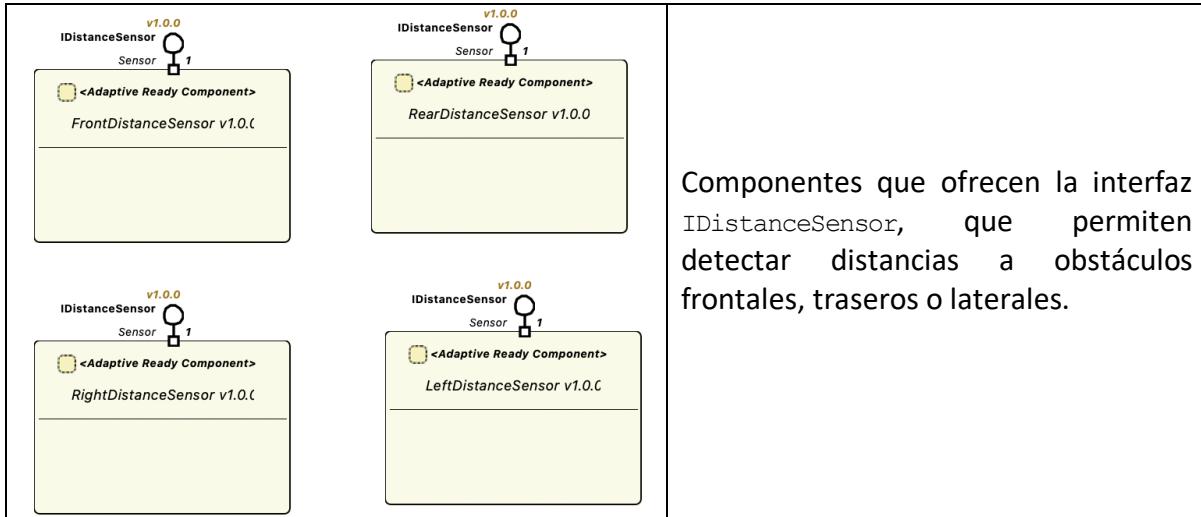
4.1.3 Steering

	<p>Componente que ofrece la interfaz <code>ISteering</code>, que permite controlar la dirección del vehículo</p>
---	--

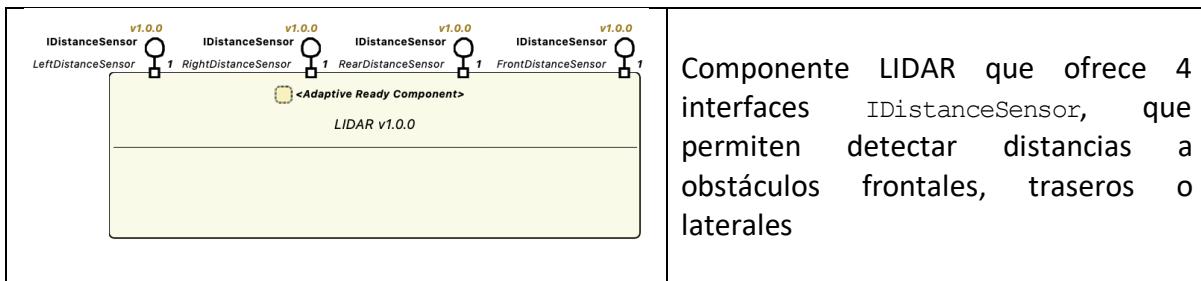
4.1.4 LineSensors

		<p>Componentes que ofrecen la interfaz <code>ILineSensor</code>, que permiten detectar cambios de carril</p>
---	---	--

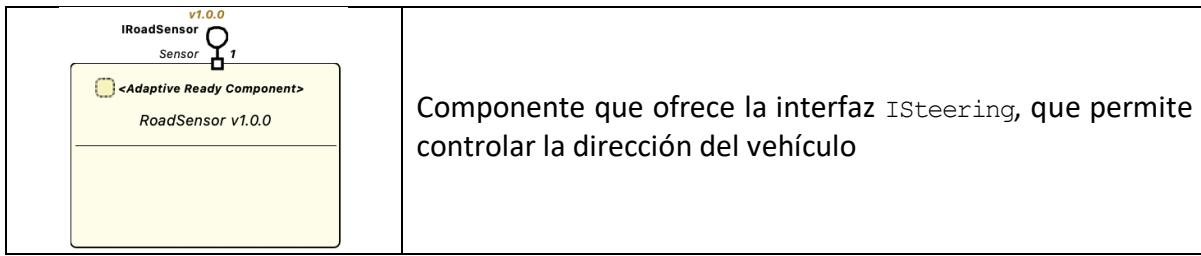
4.1.5 DistanceSensors



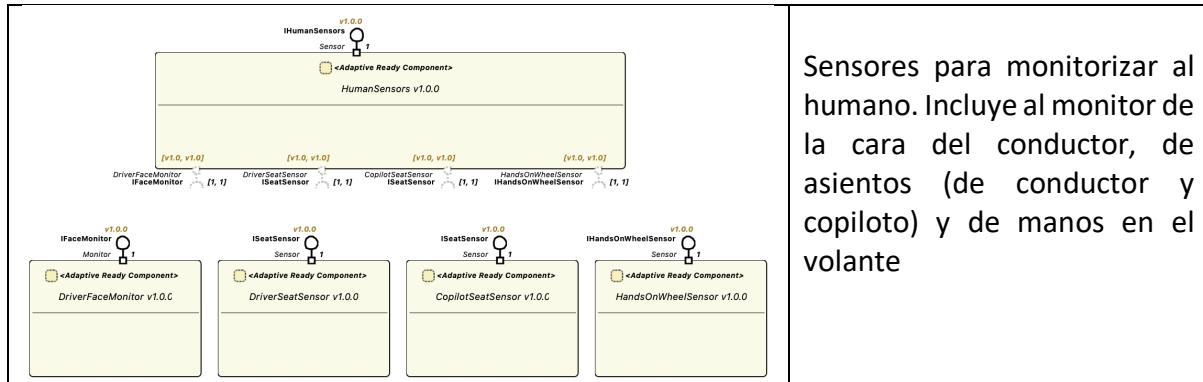
4.1.6 LIDAR-DistanceSensors



4.1.7 RoadSensor



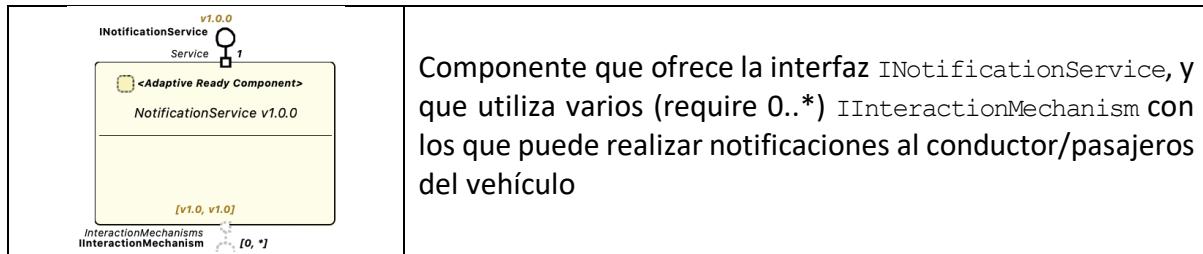
4.1.8 HumanSensors



Sensores para monitorizar al humano. Incluye al monitor de la cara del conductor, de asientos (de conductor y copiloto) y de manos en el volante

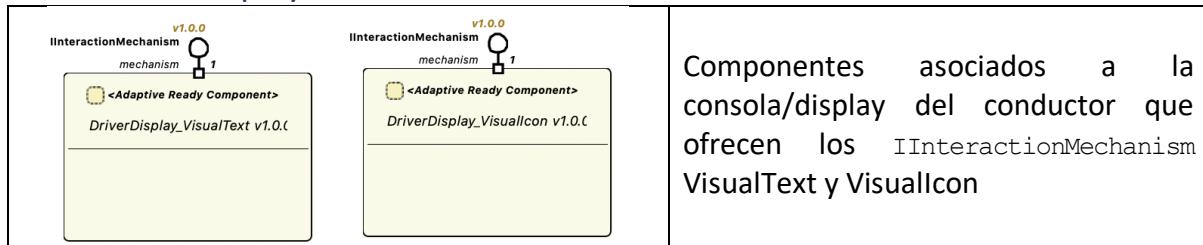
4.2 Mecanismos de Interacción

4.2.1 NotificationService



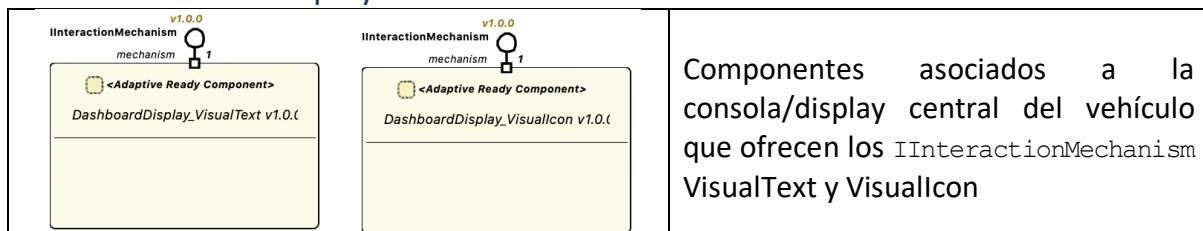
Componente que ofrece la interfaz `INotificationService`, y que utiliza varios (requiere `0..*`) `IInteractionMechanism` con los que puede realizar notificaciones al conductor/pasajeros del vehículo

4.2.2 DriverDisplay



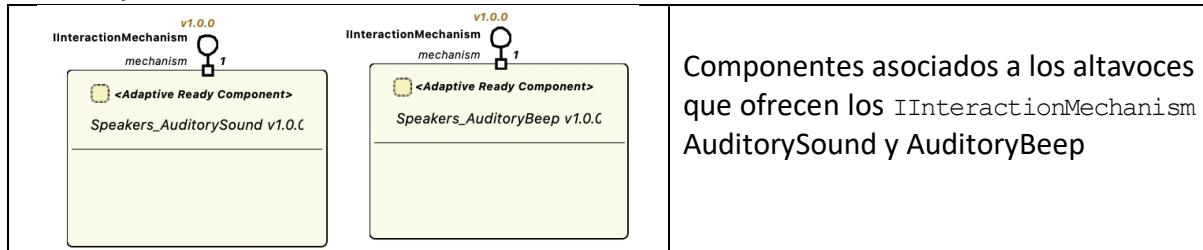
Componentes asociados a la consola/display del conductor que ofrecen los `IInteractionMechanism` `VisualText` y `VisualIcon`

4.2.3 DashboardDisplay

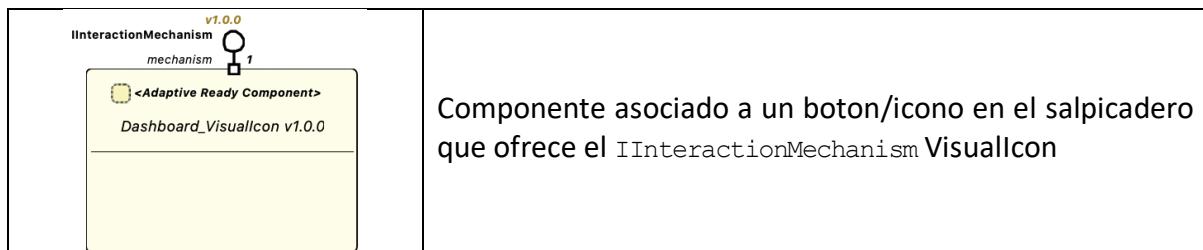


Componentes asociados a la consola/display central del vehículo que ofrecen los `IInteractionMechanism` `VisualText` y `VisualIcon`

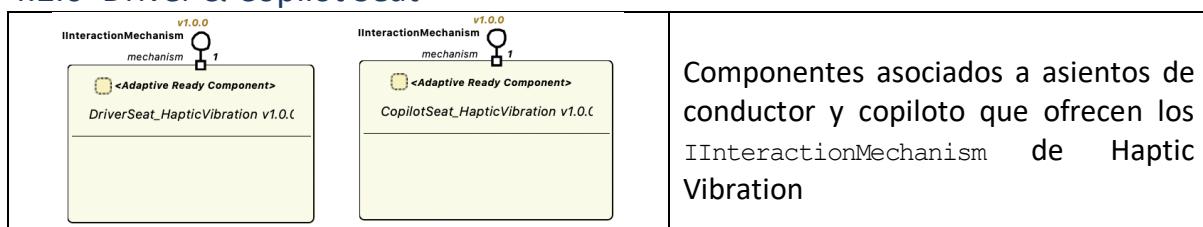
4.2.4 Speakers



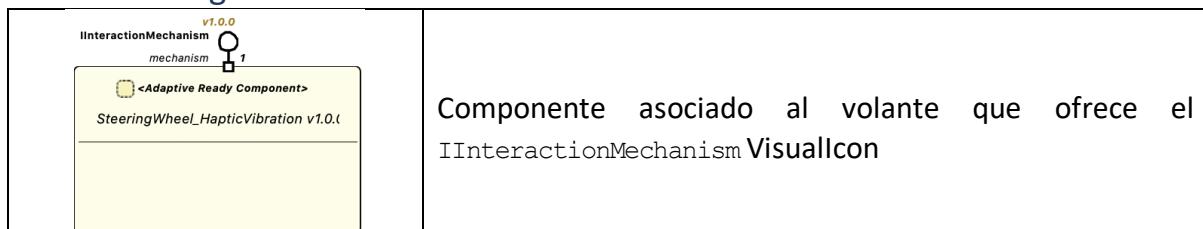
4.2.5 Dashboard



4.2.6 Driver & Copilot Seat

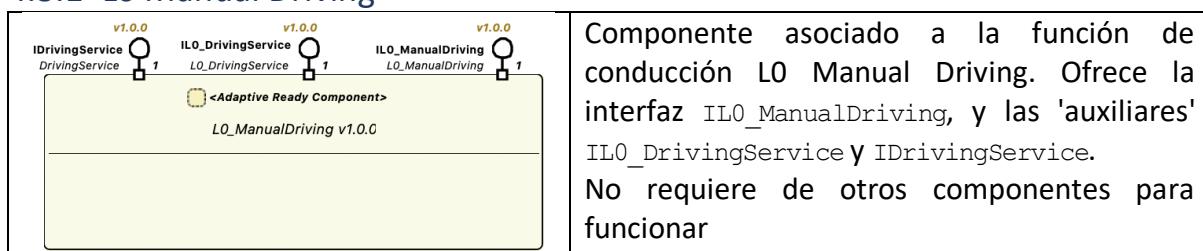


4.2.7 Steering Wheel



4.3 Funciones de Conducción Autónoma

4.3.1 L0 Manual Driving



4.3.2 L1 Assisted Driving

<pre> classDiagram class L1_AssistedDriving { <<Adaptive Ready Component>> L1_AssistedDriving v1.0.0 } interface IDrivingService { v1.0.0 } interface IL1_DrivingService { v1.0.0 } interface IL1_AssistedDriving { v1.0.0 } interface ILineSensor { [v1.0, v1.0] } interface INotificationService { [v1.0, v1.0] } IDrivingService "1" --> "1" L1_DrivingService IL1_DrivingService "1" --> "1" L1_AssistedDriving IL1_AssistedDriving "1" --> "1" LongitudinalSecurityDistance:INTEGER ILineSensor "1" --> "1" RightLineSensor ILineSensor "1" --> "1" LeftLineSensor INotificationService "0..1" --> "1" L1_AssistedDriving </pre>	<p>Componente asociado a la función de conducción L1 Assisted Driving. Ofrece IL1_AssistedDriving (y las '*Driving auxiliares'). Requiere un IDistanceSensor (front), y dos ILineSensor (right y left). Todos estos <i>require</i> son obligatorios (1..1). Requiere opcionalmente un INotificationService (0..1). Requiere un parámetro Longitudinal Security Distance con el que indicar la distancia de seguridad</p>
--	--

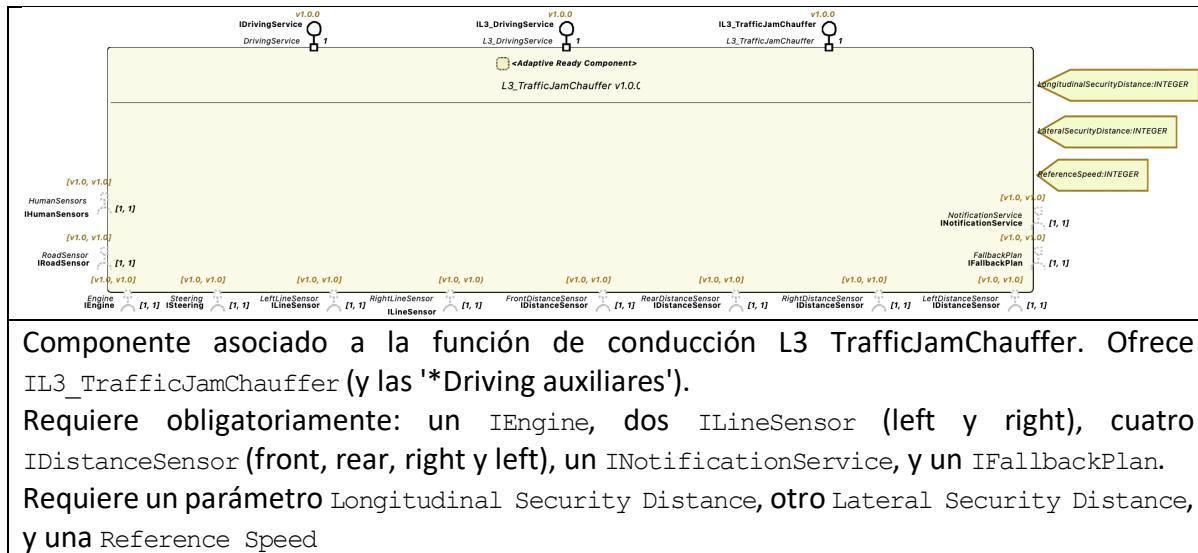
4.3.3 L2 AdaptiveCruiseControl

<pre> classDiagram class L2_AdaptiveCruiseControl { <<Adaptive Ready Component>> L2_AdaptiveCruiseControl v1.0.0 } interface IDrivingService { v1.0.0 } interface IL2_DrivingService { v1.0.0 } interface IL2_AdaptiveCruiseControl { v1.0.0 } interface INotificationService { [v1.0, v1.0] } IDrivingService "1" --> "1" IL2_DrivingService IL2_DrivingService "1" --> "1" IL2_AdaptiveCruiseControl IL2_AdaptiveCruiseControl "1" --> "1" LongitudinalSecurityDistance:INTEGER INotificationService "0..1" --> "1" L2_AdaptiveCruiseControl </pre>	<p>Componente asociado a la función de conducción L2 Adaptive Cruise Control. Ofrece IL2_AdaptiveCruiseControl (y las '*Driving auxiliares'). Requiere un IEngine, un IDistanceSensor (front), y un INotificationService. Todos estos <i>require</i> son obligatorios (1..1). Requiere un parámetro Longitudinal Security Distance con el que indicar la distancia de seguridad</p>
--	---

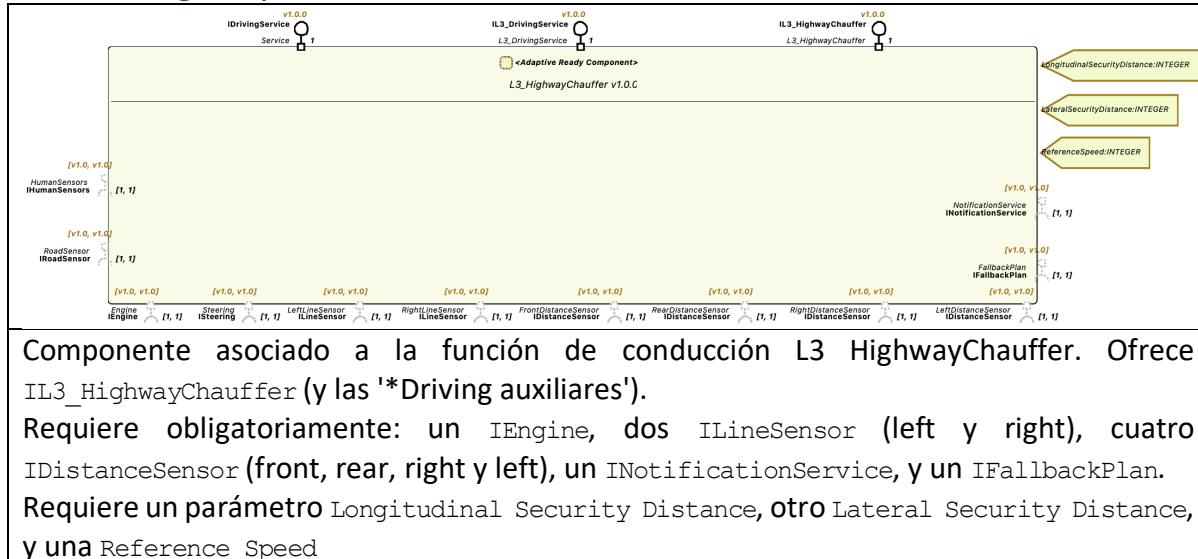
4.3.4 L2 LaneKeepingAssist

<pre> classDiagram class L2_LaneKeepingAssist { <<Adaptive Ready Component>> L2_LaneKeepingAssist v1.0.0 } interface IDrivingService { v1.0.0 } interface IL2_DrivingService { v1.0.0 } interface IL2_LaneKeepingAssist { v1.0.0 } interface ISteering { [v1.0, v1.0] } interface ILineSensor { [v1.0, v1.0] } interface INotificationService { [v1.0, v1.0] } IDrivingService "1" --> "1" IL2_DrivingService IL2_DrivingService "1" --> "1" IL2_LaneKeepingAssist IL2_LaneKeepingAssist "1" --> "1" ISteering ILineSensor "1" --> "1" LeftLineSensor ILineSensor "1" --> "1" RightLineSensor INotificationService "0..1" --> "1" L2_LaneKeepingAssist </pre>	<p>Componente asociado a la función de conducción L2 Lane Keeping Assist. Ofrece IL2_LaneKeepingAssist (y las '*Driving auxiliares'). Requiere un ISteering, y dos ILineSensor (left y right). Todos estos <i>require</i> son obligatorios (1..1). No requiere parámetros</p>
--	---

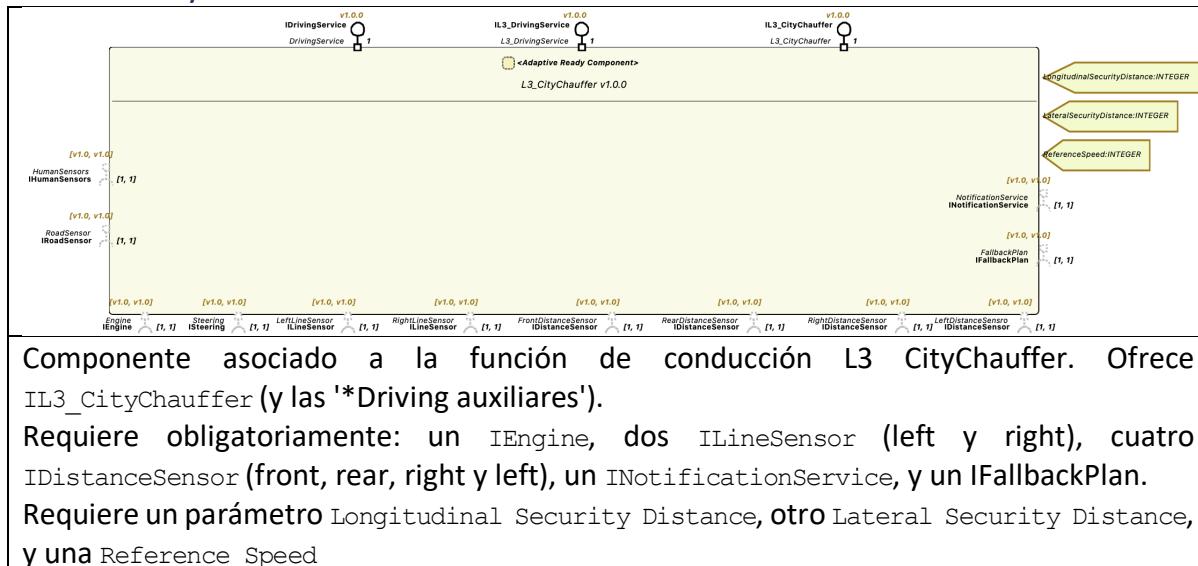
4.3.5 L3 TrafficJamChauffer



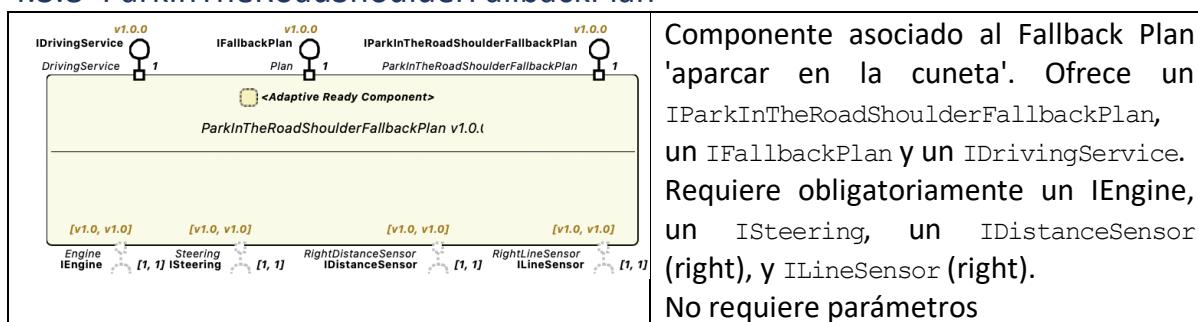
4.3.6 L3 HighwayChauffer



4.3.7 L3 CityChauffer



4.3.8 ParkInTheRoadShoulderFallbackPlan



4.3.9 EmergencyFallbackPlan



4.4 Modelo de Componentes

La Figura 10 muestra el catálogo completo de componentes² que constituyen el vehículo autónomo. En azul aparecen resaltados los dispositivos del vehículo. En naranja los mecanismos de interacción, y en verde las funciones de conducción autónoma.

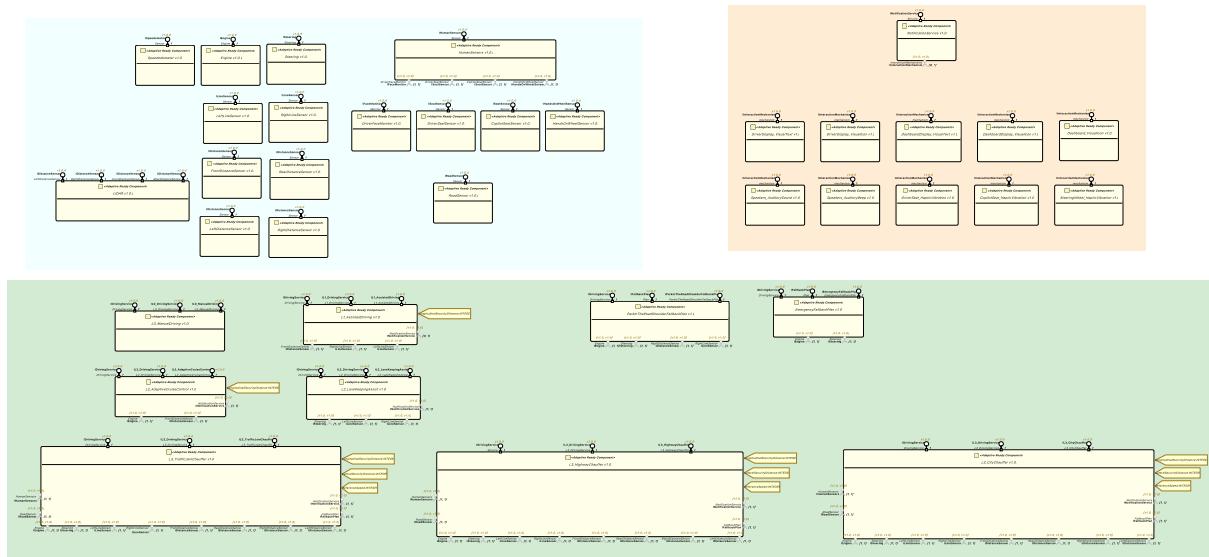


Figura 10 : Modelo de Componentes del Vehículo Autónomo

² Se adjunta imagen en fichero PDF con el modelo de componentes completo para una mejor visualización

5 Implementación del Vehículo Autónomo

Siguiendo las pautas del diseño propuesto, se ha desarrollado una implementación con Java/OSGi de los componentes identificados. Por cada componente del diagrama de componentes, se ha desarrollado un '*bundle OSGi*' que lo implementa (en forma de '*plugin project*'). Para aprender más sobre esta tecnología, ver el material dispuesto en el PoliformaT de la asignatura.

La implementación que se proporciona ofrece las funcionalidades básicas del vehículo autónomo, permitiendo realizar (simular) tareas de conducción autónoma. Sin embargo, este vehículo [1] no ofrece capacidades de auto-adaptación (o de computación autónoma), según lo visto en la asignatura. La inclusión de estas capacidades será el trabajo que deberá realizar el alumno, y se documenta en otro documento (*self-adaptive AutonomousCar: un vehículo autónomo con capacidades de auto-adaptación: Parte 2 Introducción de capacidades self-adaptive*).

5.1 Componentes del Sistema

La

Figura 11 muestra la organización de proyectos/plugins/bundles OSGi de desarrollo. Como se observa, existe un paralelismo directo con el diseño referencia descrito en la Sección 4.

Hay un conjunto de plugins de infraestructura, que contienen interfaces, o la implementación del simulador del vehículo

Existe otro grupo de proyectos que contiene la implementación de los dispositivos del vehículo (*device.**)

Los servicios de conducción autónoma están implementados en estos componentes (*driving.**)

Los mecanismos de interacción están implementados en estos componentes (*interaction.**)



Figura 11 : Proyectos (bundles) del vehículo

A continuación se describen algunos de los componentes más representativos.

5.1.1 AutonomousCar.interfaces

Ofrece las interfaces (contratos) entre los diferentes componentes. También se organizan según su objetivo.

Si se observa este proyecto/plugin podremos ver que contiene todas las interfaces definidas en el diagrama de componentes (ver Figura 10), e incluye algunas más relacionadas con soporte a implementación (`IThing`, `ISimulationElement`, ó `EFaceStatus`, entre otras).

Este plugin es vital para el resto, y es importado por todos los demás plugins OSGi (en sus respectivos `MANIFEST.MF`).

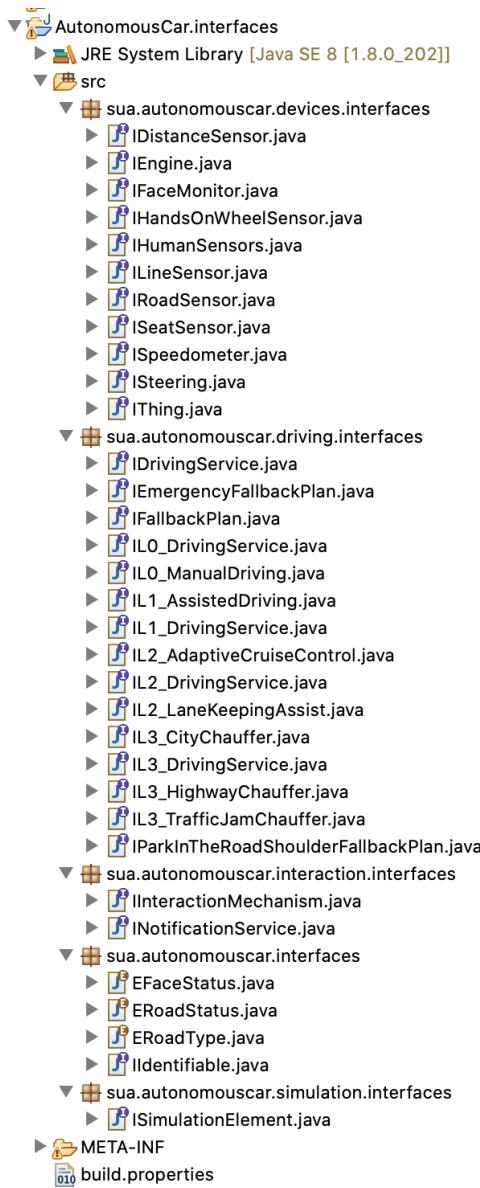


Figura 12 : Proyecto AutonomousCar.interfaces

5.1.2 AutonomousCar.infrastructure

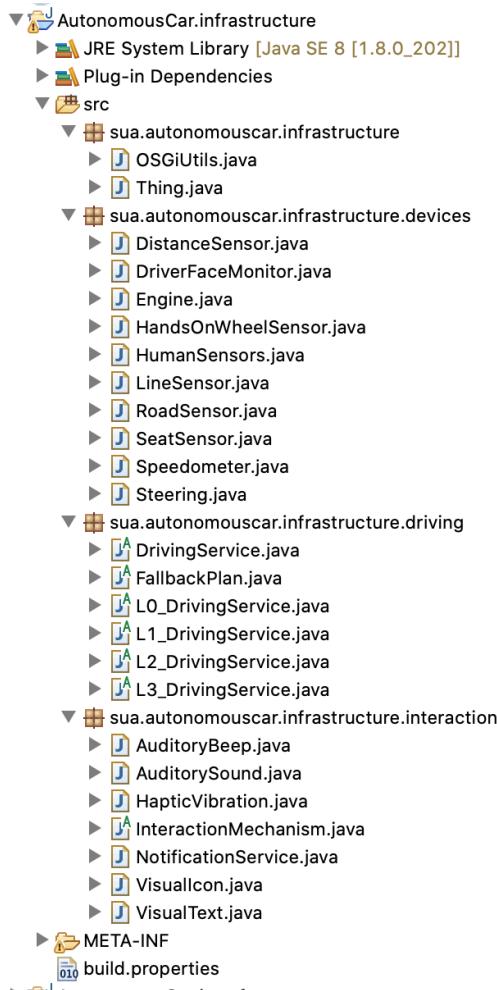


Figura 13 : AutonomousCar.infrastructure

5.1.3 device.Engine

Proyecto que arranca y detiene el componente Engine disponible en sua.autonomouscar.infrastructure.devices a través de un Activator OSGi (ver Figura 14).



Figura 14 : Proyecto device.Engine

Este Activator se encarga de registrar (en el `start()`) y desregar (en el `stop()`) el motor (`Engine`) en una solución (ver Figura 15). De esta manera, cualquier otro objeto que requiera el motor, puede encontrarlo a través de su interfaz (`IEngine`) y/o sabiendo su identificador (`id=Engine`).

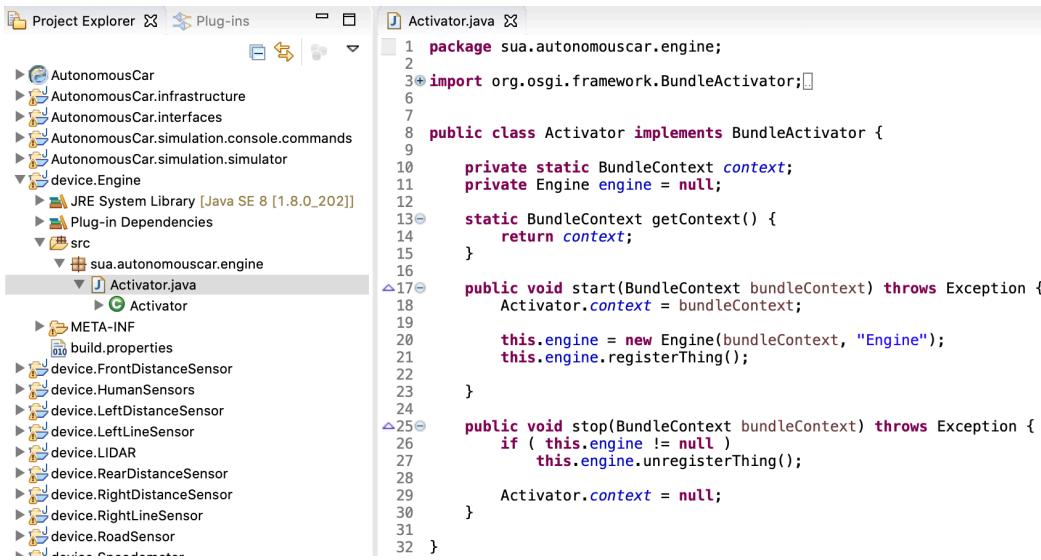
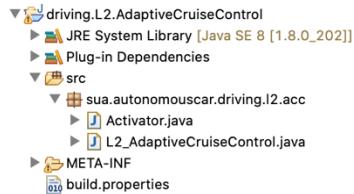


Figura 15 : Activator del device.Engine

5.1.4 driving.L2.AdaptiveCruiseControl



*Figura 16 : Proyecto
driving.L2.AdaptiveCruiseControl*

Este proyecto contiene la implementación del asistente al movimiento longitudinal en un nivel de autonomía L2 (descrito en la sección 3.2.3).

Su Activator, al igual que en el caso de Engine, se encarga de registrar/desregar el objeto (ver Figura 17).

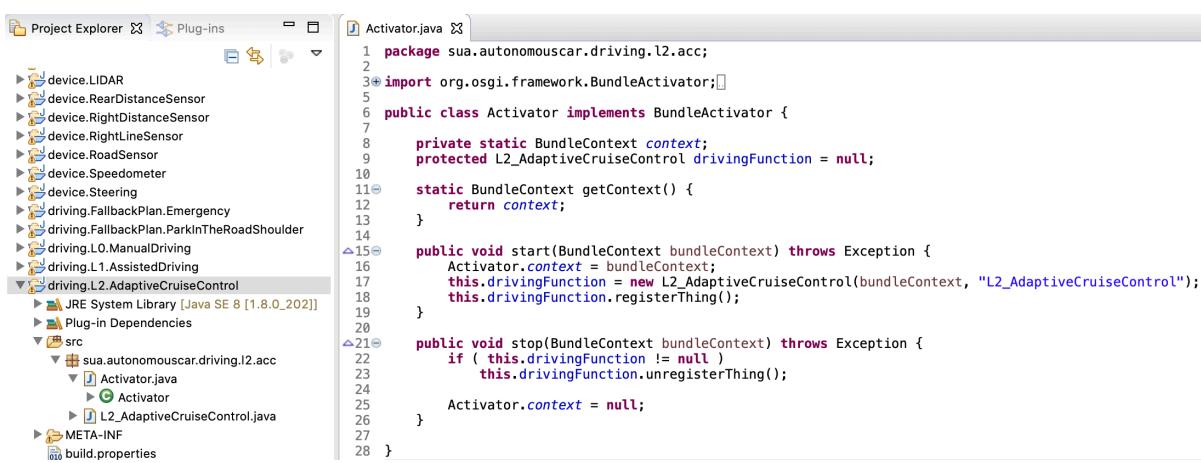


Figura 17: Activator del plugin L2_AdaptiveCruiseControl

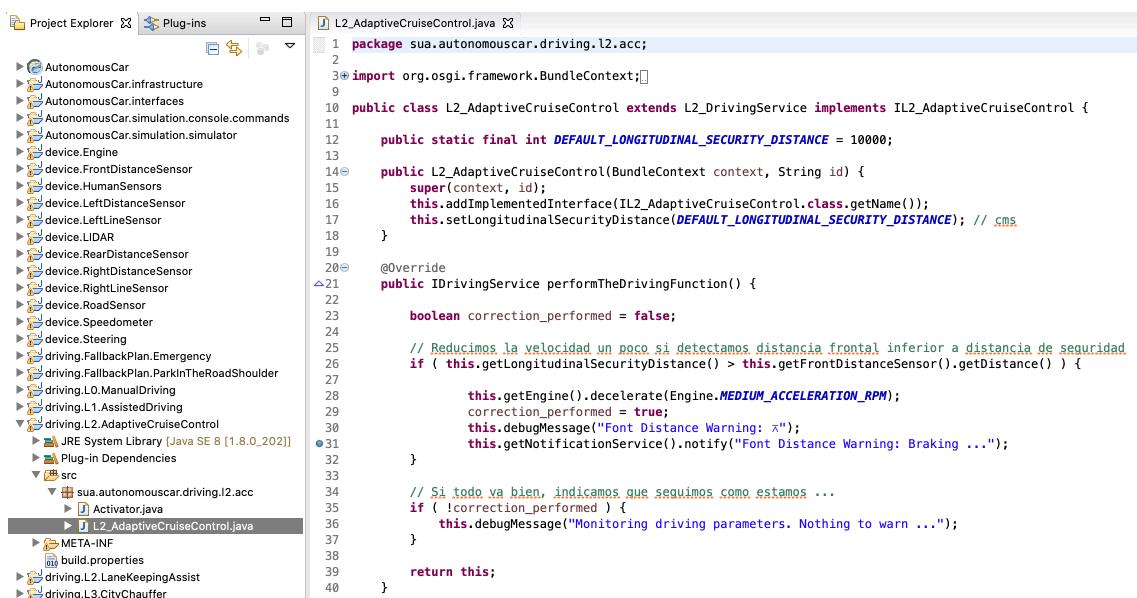
Sin embargo, este Activator no inicia el objeto `L2_AdaptiveCruiseControl`, ni lo configura. Es decir, para que este módulo de conducción autónoma pueda funcionar, se necesita que algún

otro componente lo configure (estableciendo sus *require* y *parámetros*) y lo ponga en marcha (`startDriving()` ó `stopDriving()`)³.

Siguiendo los preceptos de la asignatura, esta configuración se delega, a propósito, a otro componente que hará de controlador, y que en todo momento se encargará de que el servicio esté configurado de manera adecuada (*self-configuration*). Este es uno de los requisitos de la computación autónoma. Este razonamiento se aplica a todos los componentes del sistema, sólo que algunos, por su configuración, no requieren auto-configuración. Por ejemplo, el `Engine` no requiere de otros servicios/interfaces, y no posee parámetros, por lo que no es necesario cambiar dinámicamente su configuración.

En la implementación que se proporciona en esta entrega, sólo atiende a aspectos funcionales, por lo que no realiza esta tarea de auto-configuración. Para suplir esta necesidad hasta que el alumno lo implemente (en futuros trabajos), la tarea de configurar los servicios y componentes está delegada a un módulo del simulador del vehículo, que se verá más adelante.

Asumiendo que el servicio `L2_AdaptiveCruiseControl` está configurado con un `IEngine`, un `IDistanceSensor` (front) y un `INotificationService` (según su especificación vista en la sección 4.3.3), la implementación de su funcionalidad se puede ver en la siguiente figura:



```

Project Explorer ▾ Plug-ins ▾
  ▾ Autonomouscar
    ▾ AutonomousCar.infrastructure
    ▾ AutonomousCar.interfaces
    ▾ AutonomousCar.simulation.console.commands
    ▾ AutonomousCar.simulation.simulator
    ▾ device.Engine
      ▾ device.FrontDistanceSensor
      ▾ device.HumanSensors
      ▾ device.LeftDistanceSensor
      ▾ device.LeftLineSensor
      ▾ device.LIDAR
      ▾ device.RearDistanceSensor
      ▾ device.RightDistanceSensor
      ▾ device.RightLineSensor
      ▾ device.RoadSensor
      ▾ device.Spedometer
      ▾ device.Steering
    ▾ driving.FallbackPlan.Emergency
    ▾ driving.FallbackPlan.ParkInTheRoadShoulder
    ▾ driving.L0.ManualDriving
    ▾ driving.L1.AssistedDriving
    ▾ driving.L2.AdaptiveCruiseControl
      ▾ JRE System Library [Java SE 8 [1.8.0_202]]
      ▾ Plug-in Dependencies
    ▾ src
      ▾ sua.autonomouscar.driving.l2.acc
        ▾ Activator.java
        ▾ L2_AdaptiveCruiseControl.java
      ▾ META-INF
      ▾ build.properties
    ▾ driving.L2.LaneKeepingAssist
    ▾ driving.L3.CityChauffer

L2_AdaptiveCruiseControl.java ▾
1 package sua.autonomouscar.driving.l2.acc;
2
3 import org.osgi.framework.BundleContext;
4
5 public class L2_AdaptiveCruiseControl extends L2_DrivingService implements IL2_AdaptiveCruiseControl {
6
7   public static final int DEFAULT_LONGITUDINAL_SECURITY_DISTANCE = 10000;
8
9   public L2_AdaptiveCruiseControl(BundleContext context, String id) {
10     super(context, id);
11     this.addImplementedInterface(IL2_AdaptiveCruiseControl.class.getName());
12     this.setLongitudinalSecurityDistance(DEFAULT_LONGITUDINAL_SECURITY_DISTANCE); // cms
13   }
14
15   @Override
16   public IDrivingService performTheDrivingFunction() {
17
18     boolean correction_performed = false;
19
20     // Reducimos la velocidad un poco si detectamos distancia frontal inferior a distancia de seguridad
21     if (this.getLongitudinalSecurityDistance() > this.getFrontDistanceSensor().getDistance() ) {
22
23       this.getEngine().decelerate(Engine.MEDIUM_ACCELERATION_RPM);
24       correction_performed = true;
25       this.debugMessage("Front Distance Warning: x");
26       this.getNotificationService().notify("Front Distance Warning: Braking ...");
27     }
28
29     // Si todo va bien, indicamos que seguimos como estamos ...
30     if ( !correction_performed ) {
31       this.debugMessage("Monitoring driving parameters. Nothing to warn ...");
32     }
33
34     return this;
35   }
36
37   // Operaciones incluidas en la interfaz IDrivingService, a la que IL2_AdaptiveCruiseControl extiende
38
39
40 }

```

Figura 18 : Implementación de `L2_AdaptiveCruiseControl`

³ Operaciones incluidas en la interfaz `IDrivingService`, a la que `IL2_AdaptiveCruiseControl` extiende

5.1.5 interaction.DriverDisplay

Contiene la implementación de los mecanismos de interacción disponibles en la consola del conductor (DriverDisplay_VisualText y DriverDisplay_VisualIcon), según lo visto en la sección 4.2.2.

```

Project Explorer ▾ Plug-ins ▾
  ▷ AutonomousCar.simulation.simulator
  ▷ device.Engine
  ▷ device.FrontDistanceSensor
  ▷ device.HumanSensors
  ▷ device.LeftDistanceSensor
  ▷ device.LeftLineSensor
  ▷ device.LIDAR
  ▷ device.RearDistanceSensor
  ▷ device.RightDistanceSensor
  ▷ device.RightLineSensor
  ▷ device.RoadSensor
  ▷ device.Speedometer
  ▷ device.Steering
  ▷ drivingFallbackPlan.Emergency
  ▷ drivingFallbackPlan.ParkInTheRoadShoulder
  ▷ driving.L0.ManualDriving
  ▷ driving.L1.AssistedDriving
  ▷ driving.L2.AdaptiveCruiseControl
  ▷ driving.L2.LaneKeepingAssist
  ▷ driving.L3.CityChauffer
  ▷ driving.L3.HighwayChauffer
  ▷ driving.L3.TrafficJamChauffer
  ▷ interaction.DashboardDisplay
  ▷ interaction.DriverDisplay
    ▷ JRE System Library [Java SE 8 [1.8.0_202]]
    ▷ Plug-in Dependencies
  ▷ src
    ▷ sua.autonomouscar.interaction.driverdisplay
      ▷ Activator.java
      ▷ Activator
  ▷ META-INF

Activator.java ▾
1 package sua.autonomouscar.interaction.driverdisplay;
2
3 import org.osgi.framework.BundleActivator;
4
5 public class Activator implements BundleActivator {
6
7     private static BundleContext context;
8     protected VisualText im_visualtext = null;
9     protected VisualIcon im_visualicon = null;
10
11    static BundleContext getContext() {
12        return context;
13    }
14
15    public void start(BundleContext bundleContext) throws Exception {
16        Activator.context = bundleContext;
17
18        String deviceId = "DriverDisplay";
19
20        this.im_visualtext = new VisualText(bundleContext, deviceId);
21        this.im_visualtext.registerThing();
22
23        this.im_visualicon = new VisualIcon(bundleContext, deviceId);
24        this.im_visualicon.registerThing();
25
26    }
27
28    public void stop(BundleContext bundleContext) throws Exception {
29        if ( this.im_visualtext != null )
30            this.im_visualtext.unregisterThing();
31
32        if ( this.im_visualicon != null )
33            this.im_visualicon.unregisterThing();
34
35        Activator.context = null;
36
37    }
38
39}
40
41}
42

```

The screenshot shows the Eclipse IDE interface with the Project Explorer view on the left and the code editor view on the right. The code editor displays the Java file 'Activator.java' which implements the 'BundleActivator' interface. The file contains code for starting and stopping the plugin, creating instances of 'VisualText' and 'VisualIcon', and registering them with a specific device ID ('DriverDisplay'). The Project Explorer view shows various Java packages and files related to the 'AutonomousCar' project, including 'AutonomousCar.simulation.simulator', 'device', 'driving', 'interaction', and 'src' (containing 'sua.autonomouscar.interaction.driverdisplay'). The 'META-INF' folder is also visible.

Figura 19 : Implementación de interaction.DriverDisplay

Como se puede apreciar en la Figura 19, este dispositivo registra/desregistra dos mecanismos de interacción (VisualText y VisualIcon) en OSGi. Estos mecanismos serán usados por el servicio de notificación que, en función de cómo esté configurado (qué mecanismos tenga asociados), conseguirá transmitir información, alertas y avisos de la mejor manera al conductor y pasajeros. Al utilizar estos mecanismos de interacción del dispositivo DriverDisplay, está intentando comunicarse directamente con el conductor.

5.2 Configuración y puesta en marcha

Deberemos crear un workspace e importar todos los proyectos que contienen la implementación. La implementación se encuentra disponible en PoliformaT, en:

Recursos > AutonomousCar > implementación

Descomprimimos el recurso `AutonomousCar.zip`, y desde Eclipse hacemos un

File > Import > Existing Projects into Workspace

, y seleccionamos los proyectos de la carpeta descomprimida.

Una vez importados, podremos revisar y explorar la implementación, tal y como se ha descrito antes.

5.2.1 Configuración de arranque OSGi

Para ejecutar la solución, primero debemos crear una configuración de arranque (Run Configuration ...). Existen dos maneras: utilizar el importador ó crearla manualmente.

Para importarla, hay que descargarse de PoliformaT el fichero con la configuración de arranque llamada `AutonomousCar.launch`⁴. Luego la importamos con

File > Import ... > Run/Debug > Launch Configurations

buscamos y seleccionamos el fichero `AutonomousCar.launch`, y le damos a OK. Nos aparecerá ahora la configuración de arranque.

Para crearla manualmente, vamos

Run > Run Configurations ...

y creamos una nueva configuración de tipo 'OSGi Framework'. Deberemos seleccionar todos los plugins del workspace, y los básicos para ejecutar una solución con OSGi/Equinox. La Figura 20 muestra esta configuración.

⁴ En realidad existen dos configuraciones: una (*manual simulator*) y otra (*timed simulator*). Recomiendo importar ambas y ejecutar primero la del simulador manual, que es a la que se refiere en este momento.

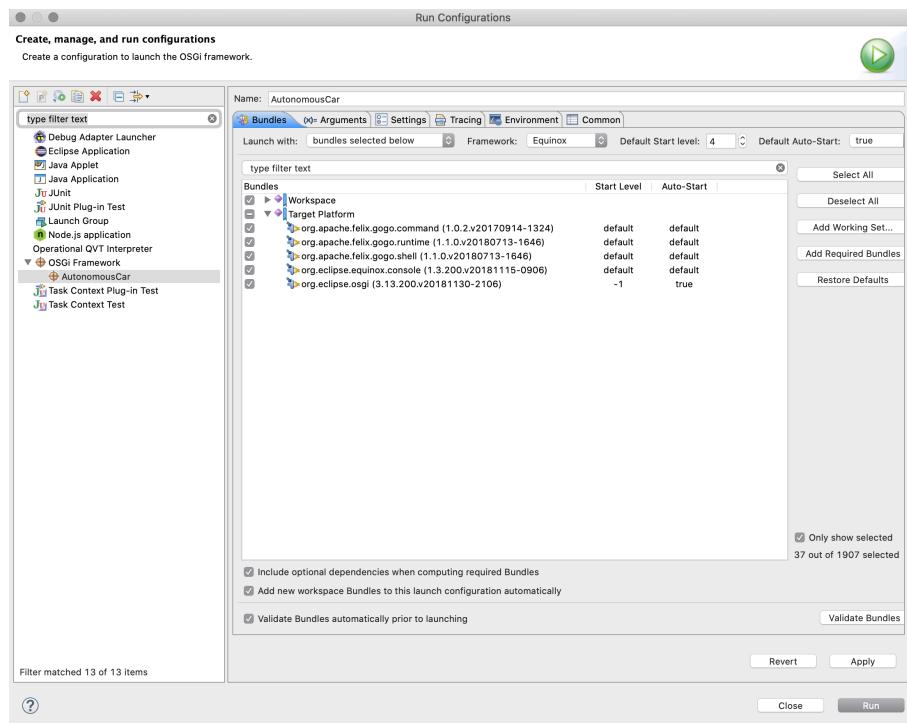


Figura 20 : Configuración de arranque OSGi del vehículo autónomo

Debemos también, por seguridad, marcar la opción 'Clear de configuration area before launching' para que cada vez que arranquemos una solución, se 'limpie' el área de ejecución y arranque 'fresca'.

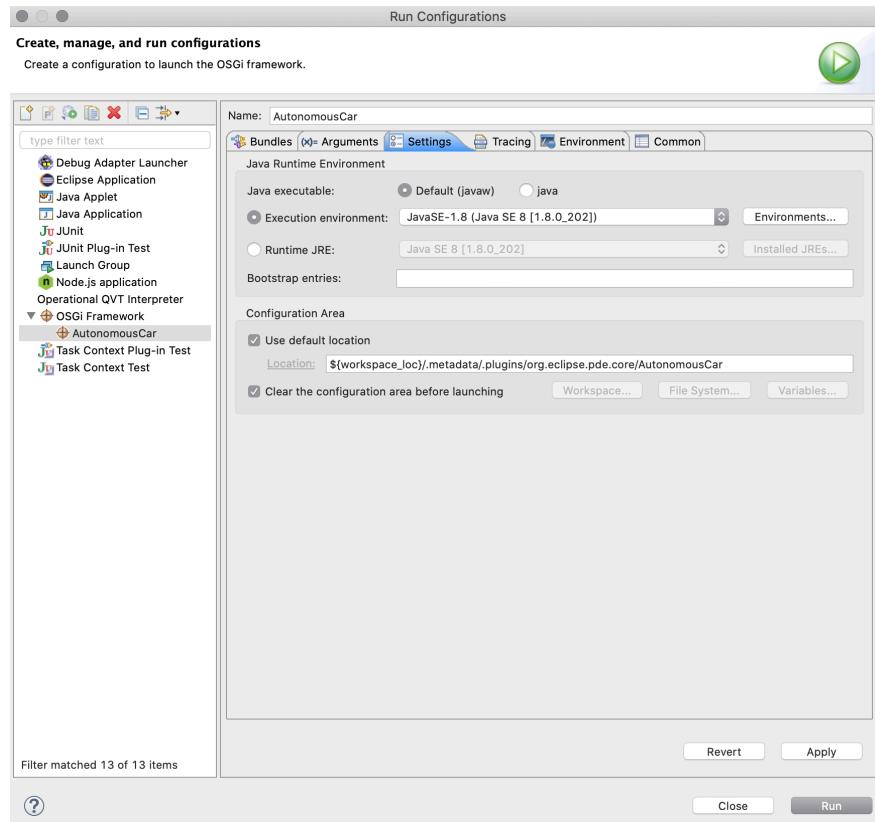


Figura 21 : Configuración Avanzada de la Run Configuration

Una vez creada la configuración de arranque, podremos ejecutar la solución (Run), y ver la consola de OSGi. Si no hay errores, veremos que la consola sale ' limpia'.

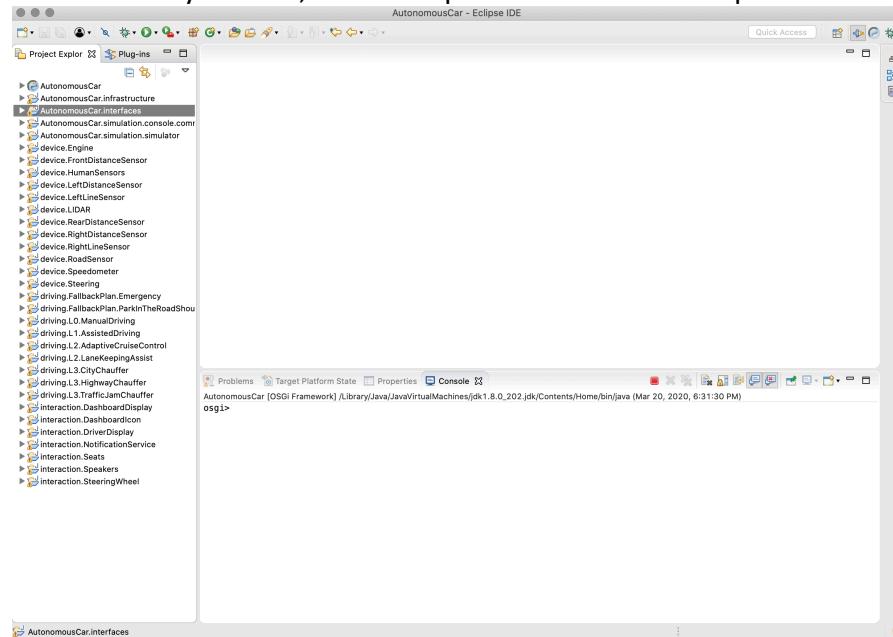


Figura 22 : Consola de OSGi tras la ejecución inicial

Si ejecutamos la orden `ss` para que nos muestre los bundles activos, obtendremos algo parecido a lo siguiente:

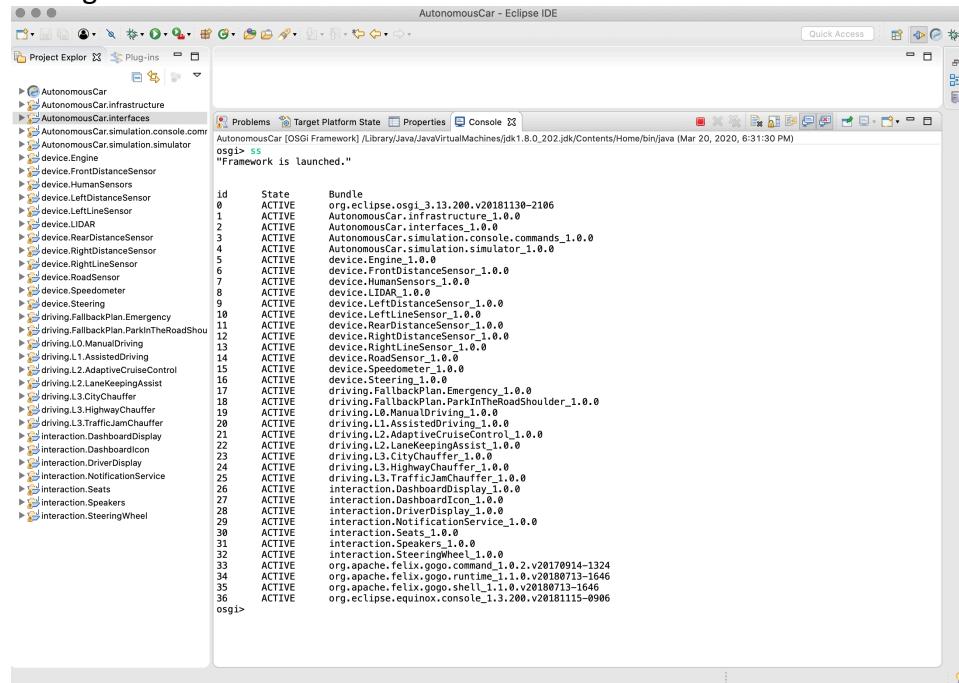


Figura 23 : Bundles activo en la configuración de arranque del vehículo

Como se puede apreciar, todos los bundles de la solución están activos.

En la siguiente sección se muestra el simulador desarrollado para poner en marcha el vehículo y configurar su contexto de conducción.

6 Simulador

Dado que no tenemos la posibilidad de desplegar esta infraestructura en un vehículo real, se ha optado por desarrollar un pequeño simulador que permita animar y ejecutar el vehículo, pudiendo observar cómo se comporta.

Este simulador permite tanto el control temporal del vehículo, así como cambiar la configuración del contexto de conducción (carretera y/o conductor), introducir valores a los sensores del vehículo (distancia, carril, ...) o controlar manualmente el vehículo (control del motor y de la dirección).

Las siguientes secciones presentan este simulador, cómo lo podemos integrar en la consola de OSGi, y cómo podemos probar diferentes escenarios de conducción.

6.1 AutonomousCar.simulation.simulator

Este proyecto contiene la implementación del simulador. Utiliza el concepto de "elemento de simulation" (vinculado a la interfaz ISimulationElement) para notificar a los diferentes componentes funcionales de los pasos (steps) de simulación.

Así, por ejemplo, las funciones de conducción autónoma implementan también ISimulationElement. Cada vez que reciben una notificación del simulador, realizan los cálculos que cada elemento debe hacer. En el caso del L2_AdaptiveCruiseControl, por ejemplo, comprobará que la distancia frontal es superior a la distancia de seguridad longitudinal, o en caso contrario, frenará (desacelerará) el motor (ver sección 5.1.4).

Existen dos implementaciones del paso del tiempo del simulador: uno manual, y otro temporizado.

El simulador manual requiere que avancemos el tiempo 'paso-a-paso'. Esto nos permite tener un control más fino sobre lo que va ocurriendo en el simulador, y poder configurar el entorno (distancias, estado del conductor, tipo de vía, etc.) antes de avanzar un paso el simulador.

El simulador temporizado avanza el tiempo automáticamente (podemos indicar el periodo de cada paso). Nos permite simular una situación más real, y automáticamente el vehículo va realizando las operaciones que debe, sin necesitar que avancemos 'paso-a-paso' el tiempo. Sin embargo, puede resultar complicado probar ciertos escenarios que impliquen configurar el entorno de conducción.

6.1.1 Configuración del Simulador

Si iniciamos una solución como hemos hecho en la sección 5.2.1, se nos iniciará un simulador manual. Podemos cambiar el tipo de simulador, junto a otros parámetros modificando los Argumentos de Arranque de la Run Configuration.

Para cambiar esta configuración, debemos abrir la Run Configuration y seleccionar la pestaña 'Arguments'. La siguiente figura muestra la configuración del simulador manual⁵.

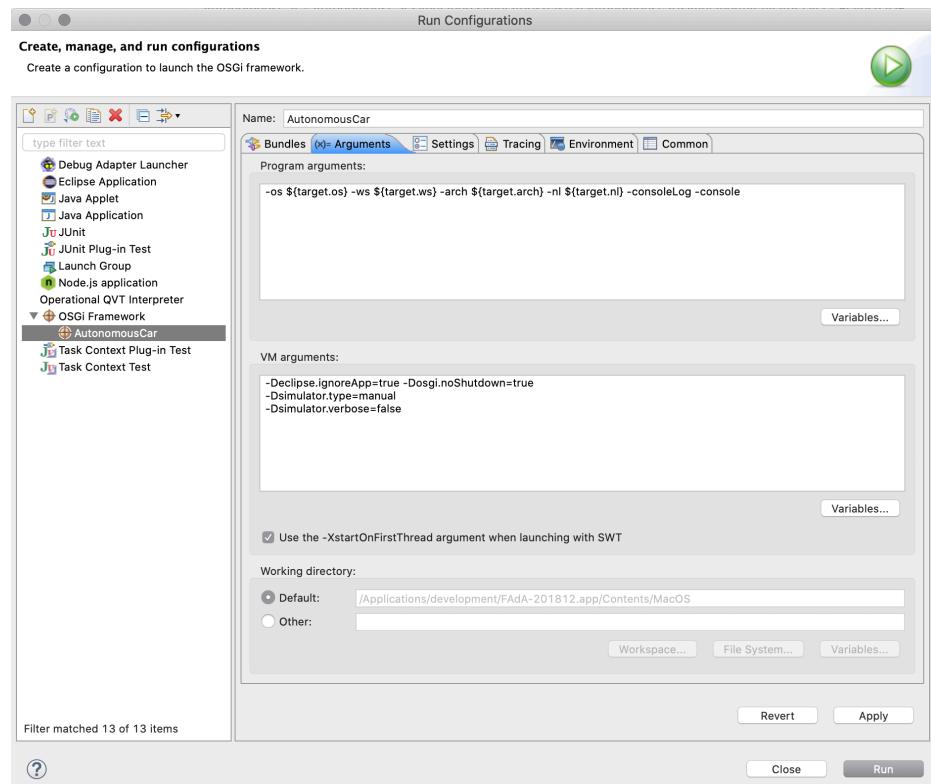


Figura 24 : Configuración del Simulador

Como se puede observar, se deben introducir estos parámetros de configuración en forma de:

`-Dparametro=valor`

Los parámetros de configuración del simulador son:

- **simulator.type** : puede tomar valores `manual` (por defecto) o `timer`
- **simulator.period** : en caso de elegir un simulador temporal (`timer`), aquí podemos indicar (en milisegundos) el `periodo` de avance (`pasos`) del simulador
- **simulator.verbose** : podemos activar (`true`) o desactivar (`false`, por defecto) la visualización de los pasos del simulador

⁵ La figura muestra los valores por defecto. Si no se indica nada, se toman estos valores.

La Figura 25 muestra un ejemplo de configuración de un simulador temporizado, con una periodicidad de pasos de 3 segundos, y 'verboso'.

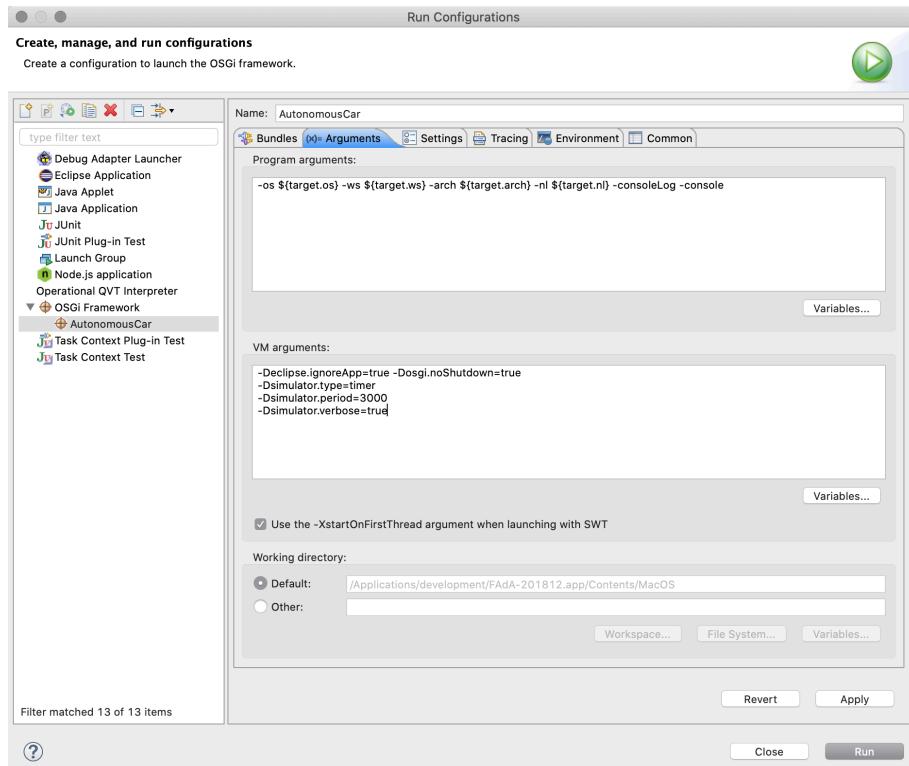


Figura 25 : Ejemplo de Configuración de un Simulador temporizado

Si ejecutamos ahora la configuración de arranque, obtendremos lo siguiente:

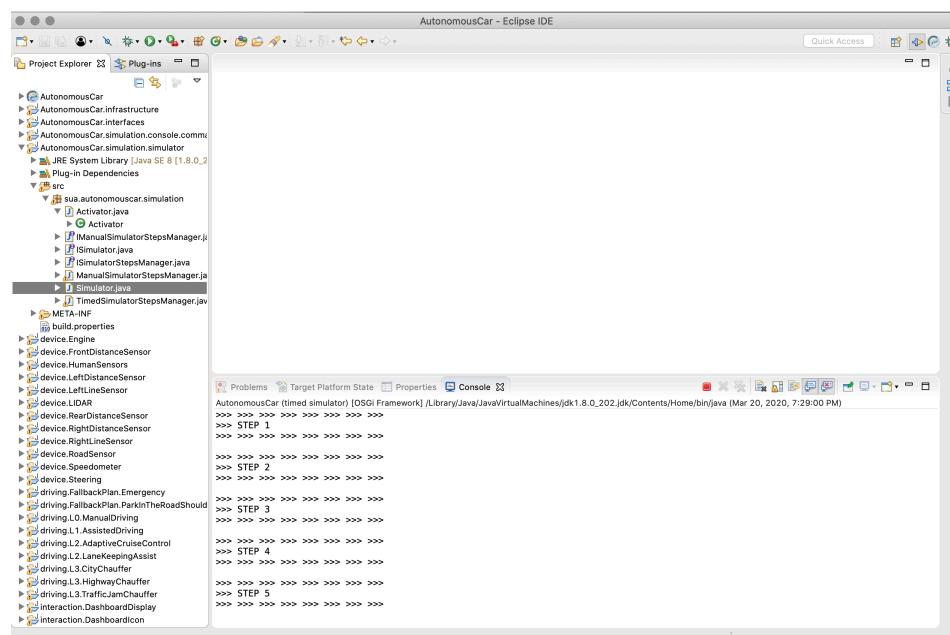


Figura 26 : Ejecución del Simulador Temporizado en modo verboso, con periodicidad 3 segundos

6.2 Comandos de simulación para la Consola de OSGi

Una vez tenemos el simulador en marcha, debemos habilitar comandos que podamos ejecutar sobre la consola, que nos permita realizar acciones sobre el vehículo.

OSGi nos proporciona el concepto de CommandProvider, que permite extender los comandos que se pueden definir. No es el objetivo de este documento explicar cómo funciona este mecanismo, pero si se observa el proyecto `AutonomousCar.simulation.console.commands`, se puede ver que el `Activator` define una serie de operaciones, y registra clase `MyCommandProvider` como `CommandProvider`, la cual implementa dichas operaciones declaradas.

Dado que se ha documentado el código con bastante detalle, pasamos a copiar el contenido del `Activator`, donde se encuentra la declaración.

Aparecen, además, un conjunto de clases del estilo `*Configurator` (ejemplo, `L0_ManualDrivingConfigurator`, `L2_AdaptiveCruiseControlConfigurator`, `EmergencyFallbackPlanConfigurator` o `NotificationServiceConfigurator`) que ofrecen operaciones para configurar y arrancar estos componentes (todos los servicios y dispositivos que requieren configuración). Esto es lo que permite que el sistema arranque con una configuración inicial (auto-configuración) y pueda funcionar bajo un contexto determinado.

```

// CONFIGURACIÓN
// 

    // configure : realiza una configuración inicial de servicios y
    // dispositivos (de prueba)
    //
    // Modo uso
    //         configure
    //
    "configure",

    // show : muestra la configuración actual de conducción
    //
    // Modo uso
    //         configure
    //
    "show",

// SIMULACIÓN LECTURAS DE SENSORES
//
    // line : sensores de carril (LineSensor)
    //
    // Modo uso
    //         line [ right | left ] [ true | false ]
    //
    // Ejemplo: detección de línea derecha de carril
    //         line right true
    //
    "line",

```

```

//  distance : sensores de distancia (DistanceSensor)
//
//  Modo uso
//      distance [ front | rear | left | right ] valor-distancia
//
//  Ejemplo: distancia frontal de 150 m (15000 cms)
//          distance front 15000
//
//  "distance",
//
//  lidar : sensores de distancia en LIDAR (LIDAR-DistanceSensor)
//
//  Modo uso
//      lidar [ front | rear | left | right ] valor-distancia
//
//  Ejemplo: distancia a obstáculo en lateral izquierdo de 70 cms
//          lidar left 70
//
//  "lidar",
//
//  PARÁMETROS DE CONTEXTO
//
//  driver : situación del conductor. Permite varios usos para
//          indicar la situación de la cara (face) y las manos en el
//          volante
//
//  Modo uso
//      driver face [ looking_forward | distracted | sleeping ]
//      driver hands [ on-wheel | off-wheel ]
//
//  Ejemplo: conductor distraído
//          driver face distracted
//
//  Ejemplo: conductor distraído
//          driver hands on-wheel
//
//  "driver",
//
//  seat : sensor de asiento de conductor y copiloto
//  Modo uso
//      seat [ driver | copilot ] [ true | false ]
//
//  Ejemplo: asiento del conductor ocupado
//          seat driver true
//
//  "seat",
//
//  road : sensor de carretera. Permite indicar tanto el tipo
//        como el estado
//
//  Modo uso
//      road type [ std | highway | off-road | city ]
//      road status [ fluid | jam | collapsed ]
//
//  Ejemplo: circulamos por ciudad
//          road type city
//
//  Ejemplo: hay atasco
//          road status jam
//
//  "road",

```

```

// CONTROL MANUAL DEL VEHÍCULO
//  

// engine : control de las revoluciones (rpm) del motor
// Modo uso
// engine [ rpm | accelerate | decelerate ] numero-rpm
// Ejemplo: Poner las revoluciones a 2500 rpm
// engine rpm 2500
// Ejemplo: Aceleramos 1000 rpm
// engine accelerate 1000
// "engine", // controlar manualmente las funciones primarias de conducción

// FUNCIONES DE CONDUCCIÓN
//  

// driving : activa un nivel de conducción autónoma
// Modo uso
// driving [ 10 | 11 | 12 | 13 ]
// Ejemplo: Activar nivel de autonomía 3
// driving 13
// * NOTA: en función del tipo de vía y su estado
// se activará el servicio adecuado para el
// nivel indicado
// "driving",  

// SIMULADOR
//  

// next ó n : da un paso de simulación manual
// Modo uso
// next
// n
// "next",
// "n"

```

6.3 Ejemplos de Ejecución de Escenarios

Las operaciones anteriores nos permiten definir la ejecución de los escenarios como una secuencia ordenada de operaciones de simulación. A continuación se muestran varios ejemplos de ejecución.

Ejemplo 1 - Probar configuración sensores

- Ejecutar la solución (simulador manual),
- ver el estado del vehículo,
- cambiar el tipo de vía a 'ciudad',
- el sensor de distancia frontal a 50 metros,
- y el sensor de carril izquierdo a true
- Mostrar la configuración final

```
osgi> show
-----| DRIVER
-----| Hands on Wheel: true
      |   Driver Face: LOOKING_FORWARD
      |   Driver Seat: true
      |   Copilot Seat: false
-----| ROAD INFO
-----| Road Type: STD_ROAD
      | Road Status: FLUID
-----| CAR INFO
-----| Speed: 0 Km/h
      | Engine: 700 rpm
      | Steering: 0 °
-----| DRIVING SERVICE
      | <none>
-----| DISTANCES
      | Front: ∞ cms
      | Rear: ∞ cms
      | Right: ∞ cms
      | Left: ∞ cms
-----| LIDAR
      | Front: ∞ cms
      | Rear: ∞ cms
      | Right: ∞ cms
      | Left: ∞ cms
-----| LINE SENSORS
      | Right: false
      | Left: false
-----|
```

```

osgi> road type city
osgi> distance front 5000
osgi> line left true
osgi> show
-----
|           DRIVER
| -----
|   Hands on Wheel: true
|     Driver Face: LOOKING_FORWARD
|     Driver Seat: true
|     Copilot Seat: false
| -----
|
|           ROAD INFO
| -----
|   Road Type: CITY
|   Road Status: FLUID
| -----
|
|           CAR INFO
| -----
|   Speed: 0 Km/h
|   Engine: 700 rpm
|   Steering: 0 °
| -----
|           DRIVING SERVICE
|   <none>
| -----
|           DISTANCES
|   Front: 5000 cms
|   Rear: ∞ cms
|   Right: ∞ cms
|   Left: ∞ cms
| -----
|           LIDAR
|   Front: ∞ cms
|   Rear: ∞ cms
|   Right: ∞ cms
|   Left: ∞ cms
| -----
|           LINE SENSORS
|   Right: false
|   Left: true
| -----

```

Ejercicio 2 - Activar una función de conducción autónoma y que el vehículo avance

- Ejecutar la solución (simulador manual)
- configurar los servicios de conducción
- cambiar el tipo de vía a 'highway'
- cambiar el estado de la vía a atasco
- indicar que el conductor está distraído
- solicitar el servicio de conducción autónoma nivel 3 (se activará el Chófer para atascos en Autovía, L3_TrafficJamChauffer, dado el tipo de carretera y su estado)

Como se podrá observar, el simulador lanzará mensajes para que el conductor mire al frente. El servicio de notificaciones está configurado ahora mismo para usar los mecanismos DriverDisplay_VisualText, Speakers_AuditoryBeep y Steering_HapticVibration.

- avanzar un paso de simulación ('next' ó 'n')
- avanzar otro paso
- indicar que la distancia frontal es de 20 metros
- paso
- indicar que el sensor de carril derecho ha detectado la linea
- paso
- mostrar el estado del vehículo

```

osgi> configure
osgi> road type highway
osgi> road status jam
osgi> driver face distracted
osgi> driving l3
[ L3_TrafficJamChauffer ] Starting the driving function ...
[ L3_TrafficJamChauffer ] Current Speed: 0 Km/h
[ L3_TrafficJamChauffer ] Accelerating (aggressive) to get the reference speeed of 60 Km/h
##           DriverDisplay_VisualText ## Please, look forward!
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
osgi> next
[ L3_TrafficJamChauffer ] Current Speed: 13 Km/h
[ L3_TrafficJamChauffer ] Accelerating (aggressive) to get the reference speeed of 60 Km/h
##           DriverDisplay_VisualText ## Please, look forward!
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
osgi> n
[ L3_TrafficJamChauffer ] Current Speed: 27 Km/h
[ L3_TrafficJamChauffer ] Accelerating (aggressive) to get the reference speeed of 60 Km/h
##           DriverDisplay_VisualText ## Please, look forward!
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
osgi> distance front 2000
osgi> n
[ L3_TrafficJamChauffer ] Current Speed: 40 Km/h
[ L3_TrafficJamChauffer ] Font Distance Warning: ⚠
##           DriverDisplay_VisualText ## Font Distance Warning: Braking ...
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
##           DriverDisplay_VisualText ## Please, look forward!
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
osgi> line right true

```

```

osgi> n
[ L3_TrafficJamChauffer ] Current Speed: 22 Km/h
[ L3_TrafficJamChauffer ] Font Distance Warning: ⚠
##           DriverDisplay_VisualText ## Font Distance Warning: Braking ...
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
[ L3_TrafficJamChauffer ] Right Line Sensor Warning: >| . Turning the Steering to the left ...
##           DriverDisplay_VisualText ## Right Line Sensor Warning: Turning to the left ...
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
##           DriverDisplay_VisualText ## Please, look forward!
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
osgi> show
-----+
|          DRIVER
-----+
|  Hands on Wheel: true
|  Driver Face: DISTRACTED
|  Driver Seat: true
|  Copilot Seat: false
-----+
-----+
|          ROAD INFO
-----+
|  Road Type: HIGHWAY
|  Road Status: JAM
-----+
-----+
|          CAR INFO
-----+
|  Speed: 4 Km/h
|  Engine: 800 rpm
|  Steering: -5 °
-----+
DRIVING SERVICE
L3_TrafficJamChauffer
  Reference Speed: 60 Km/h
  Longit. Security Distance: 6000 cms
  Lateral Security Distance: 150 cms
-----+
DISTANCES
Front: 2000 cms
Rear: ∞ cms
Right: ∞ cms
Left: ∞ cms
-----+
LIDAR
Front: ∞ cms
Rear: ∞ cms
Right: ∞ cms
Left: ∞ cms
-----+
LINE SENSORS
Right: true
Left: false
-----+

```

Ejercicio 3 - Probar un TakeOver

- Ejecutar la solución (simulador manual)
- configurar los servicios de conducción
- cambiar el tipo de vía a 'highway'
- cambiar el estado de la vía a atasco
- mostrar configuración
- activar conducción nivel 3 (se activará L3_TrafficJamChauffer) y dar 3 pasos
- salimos de la autovía y nos metemos en camino off-road
- paso y mostrar configuración

En este momento, el vehículo no puede seguir conduciendo de manera autónoma, y como el conductor está atento y preparado (en el asiento del conductor, mirando hacia adelante y con las manos en el volante), el vehículo activa el TakeOver y deja de conducir

```

osgi> configure
osgi> road type highway
osgi> road status jam
osgi> show
-----
|           DRIVER
| -----
| Hands on Wheel: true
|   Driver Face: LOOKING_FORWARD
|   Driver Seat: true
|   Copilot Seat: false
| -----
|           ROAD INFO
| -----
|   Road Type: HIGHWAY
|   Road Status: JAM
| -----
|           CAR INFO
| -----
|   Speed: 0 Km/h
|   Engine: 700 rpm
|   Steering: 0 °
| -----
DRIVING SERVICE
<none>
-----
DISTANCES
Front: ∞ cms
Rear: ∞ cms
Right: ∞ cms
Left: ∞ cms
-----
LIDAR
Front: ∞ cms
Rear: ∞ cms
Right: ∞ cms
Left: ∞ cms
-----
LINE SENSORS
Right: false
Left: false
-----
```

```

osgi> driving l3
[ L3_TrafficJamChauffer ] Starting the driving function ...
[ L3_TrafficJamChauffer ] Current Speed: 0 Km/h
[ L3_TrafficJamChauffer ] Accelerating (aggressive) to get the reference speeed of 60 Km/h
osgi> n
[ L3_TrafficJamChauffer ] Current Speed: 13 Km/h
[ L3_TrafficJamChauffer ] Accelerating (aggressive) to get the reference speeed of 60 Km/h
osgi> n
[ L3_TrafficJamChauffer ] Current Speed: 27 Km/h
[ L3_TrafficJamChauffer ] Accelerating (aggressive) to get the reference speeed of 60 Km/h
osgi> n
[ L3_TrafficJamChauffer ] Current Speed: 40 Km/h
[ L3_TrafficJamChauffer ] Accelerating (high) to get the reference speeed of 60 Km/h
osgi> road type off-road
osgi> n
[ L3_TrafficJamChauffer ] Cannot drive in L3 Autonomy level ...
##           DriverDisplay_VisualText ## Cannot drive in L3 Autonomy level ...
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
[ L3_TrafficJamChauffer ] The driver is ready to TakeOver ...
##           DriverDisplay_VisualText ## The driver is ready to TakeOver ...
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
[ L3_TrafficJamChauffer ] Ending the driving function ...
##           DriverDisplay_VisualText ## Exited Autonomous Mode
##           Speakers_AuditoryBeep ## 🎙
##           SteeringWheel_HapticVibration ## ~
osgi> show
----- 
|          DRIVER
----- 
| Hands on Wheel: true
|   Driver Face: LOOKING_FORWARD
|   Driver Seat: true
|   Copilot Seat: false
----- 

----- 
|          ROAD INFO
----- 
|   Road Type: OFF_ROAD
|   Road Status: JAM
----- 

----- 
|          CAR INFO
----- 
|   Speed: 49 Km/h
|   Engine: 1800 rpm
|   Steering: 0 °
----- 
|          DRIVING SERVICE
|   <none>
----- 
|          DISTANCES
|   Front: ∞ cms
|   Rear: ∞ cms
|   Right: ∞ cms
|   Left: ∞ cms
----- 
|          LIDAR
|   Front: ∞ cms
|   Rear: ∞ cms
|   Right: ∞ cms
|   Left: ∞ cms
----- 
|          LINE SENSORS
|   Right: false
|   Left: false
----- 

```

Ejercicio 4 - Probar un Fallback Plan

- Ejecutar la solución (simulador manual)
- configurar los servicios de conducción
- cambiar el tipo de vía a 'highway'
- activar conducción nivel 3 (se activará L3_HighwayChauffer)
- dar 3 pasos
- el conductor suelta las manos del volante
- muestra configuración
- dejamos la autovía y entramos en carretera normal (std)
- paso

Llegados a este punto, el vehículo se encontraba en conducción autónoma L3 en situación fluída por autovía, y sale de ésta. Pero el conductor no tiene las manos en el volante, por lo que no le puede ceder el control. Así que el vehículo decide activar el FallbackPlan. En esta configuración, el servicio L3_HighwayChauffer usa el FallbackPlan de aparcar en la cuneta (ParkInTheRoadShoulderFallbackPlan), así que lo activa.

- paso
- indicamos que estamos cruzando la línea derecha
- paso
- paso
- indicamos que ya no detectamos la línea derecha (la hemos cruzado)
- 12 pasos hasta que frena por completo
- mostramos configuración

```

osgi> configure
osgi> road_type highway
osgi> driving l3
[ L3_HighwayChauffer ] Starting the driving function ...
[ L3_HighwayChauffer ] Current Speed: 0 Km/h
[ L3_HighwayChauffer ] Accelerating (aggressive) to get the reference speeed of 120 Km/h
osgi> n
[ L3_HighwayChauffer ] Current Speed: 22 Km/h
[ L3_HighwayChauffer ] Accelerating (aggressive) to get the reference speeed of 120 Km/h
osgi> n
[ L3_HighwayChauffer ] Current Speed: 45 Km/h
[ L3_HighwayChauffer ] Accelerating (aggressive) to get the reference speeed of 120 Km/h
osgi> n
[ L3_HighwayChauffer ] Current Speed: 67 Km/h
[ L3_HighwayChauffer ] Accelerating (aggressive) to get the reference speeed of 120 Km/h
osgi> driver hands off-wheel
osgi> show
----- 
|          DRIVER
|----- 
| Hands on Wheel: false
|   Driver Face: LOOKING_FORWARD
|   Driver Seat: true
|   Copilot Seat: false
|----- 

```

```
|           ROAD INFO
|   Road Type: HIGHWAY
|   Road Status: FLUID
|
|           CAR INFO
|
|       Speed: 90 Km/h
|       Engine: 2700 rpm
|       Steering: 0 °
|
DRIVING SERVICE
L3_HighwayChauffer
    Reference Speed: 120 Km/h
    Longit. Security Distance: 12000 cms
    Lateral Security Distance: 200 cms
-
DISTANCES
Front: ∞ cms
Rear: ∞ cms
Right: ∞ cms
Left: ∞ cms
-
LIDAR
Front: ∞ cms
Rear: ∞ cms
Right: ∞ cms
Left: ∞ cms
-
LINE SENSORS
Right: false
Left: false
-
osgi> road type std
osgi> n
[ L3_HighwayChauffer ] Cannot drive in L3 Autonomy level ...
##          DriverDisplay_VisualText ## Cannot drive in L3 Autonomy level ...
##          Speakers_AuditoryBeep ## 🎙
##          SteeringWheel_HapticVibration ## ~
[ L3_HighwayChauffer ] Activating the Fallback Plan ...
[ L3_HighwayChauffer ] Ending the driving function ...
[ ParkInTheRoadShoulderFallbackPlan ] Starting the driving function ...
[ ParkInTheRoadShoulderFallbackPlan ] Turning to the right ...
osgi> n
[ ParkInTheRoadShoulderFallbackPlan ] Waiting to detect the right line
osgi> line right true
osgi> n
[ ParkInTheRoadShoulderFallbackPlan ] Waiting to detect the right line
[ ParkInTheRoadShoulderFallbackPlan ] Line Detected ...
osgi> n
[ ParkInTheRoadShoulderFallbackPlan ] Waiting to cross the line ...
osgi> line right false
osgi> n
[ ParkInTheRoadShoulderFallbackPlan ] Waiting to cross the line ...
[ ParkInTheRoadShoulderFallbackPlan ] Reached the Road Shoulder
[ ParkInTheRoadShoulderFallbackPlan ] Centering the steering
osgi> n
[ ParkInTheRoadShoulderFallbackPlan ] Speed: 90 Km/h
[ ParkInTheRoadShoulderFallbackPlan ] Decelerating (aggressive) ...
osgi> n
[ ParkInTheRoadShoulderFallbackPlan ] Speed: 76 Km/h
[ ParkInTheRoadShoulderFallbackPlan ] Decelerating (aggressive) ...
osgi> n
[ ParkInTheRoadShoulderFallbackPlan ] Speed: 63 Km/h
[ ParkInTheRoadShoulderFallbackPlan ] Decelerating (aggressive) ...
osgi> n
```


Ejercicio Final: Identificar nuevos escenarios que puedan requerir que el sistema realice ciertas tareas. Comprender las funciones de conducción autónoma y experimentar con el simulador a través de los comandos de simulación.

Algunos pruebas:

- activamos conducción L1 donde tendremos que acelerar y girar manualmente. Ponemos obstáculos delante, o provocamos la detección del carril, y vemos cómo el vehículo sólo nos informa. Esto responde a la responsabilidad del sistema en nivel de autonomía 1 (sólo advertencias)
- activamos funciones L2 donde tendremos que acelerar y girar manualmente. Ponemos obstáculos delante, o provocamos la detección del carril, y vemos cómo el vehículo toma acciones correctivas en una función primaria (movimiento longitudinal o lateral). Esto responde a la responsabilidad del sistema en nivel de autonomía 2 (tomar acciones parciales de control)
- ¿qué ocurre si estamos en una carretera off-road y activamos conducción L3?

7 Reflexión final

Este vehículo es autónomo a nivel de conducción ya que está programado para atender a diferentes situaciones de conducción, y a través de operaciones como el *TakeOver* o los *FallbackPlans* resuelve las situaciones conflictivas, monitorizando el contexto de la conducción (carretera, pasajeros y dispositivos del vehículo), tomando decisiones y actuando sobre los dispositivos de control del vehículo.

Sin embargo, ¿qué ocurriría si le fallara un sensor de distancia? ¿Cómo consigue el vehículo estar en todo momento usando la función de conducción autónoma más alta posible? ¿Cómo se adecúan los mecanismos de interacción para mostrar los mensajes al usuario en función de su situación, si está atento o dormido, por ejemplo? ¿Cómo se consigue que los servicios de conducción autónoma de nivel 3 usen el *FallbackPlan* de 'aparcar en la cuneta' si tienen los sensores necesarios disponibles, o en caso contrario usen el plan de emergencia (sin sensores)?

Todas estas últimas consideraciones son las que nos llevarán a diseñar las capacidades de computación autónoma, para conseguir que el vehículo sea capaz de realizar operaciones de auto-adaptación: *self-configuring*, *self-protecting*, *self-healing* o *self-optimizing*.

Este trabajo de extender con capacidades de computación autónoma es el que deberá realizarse sobre la implementación (funcional) proporcionada. En breve se publicará la segunda parte de este informe, donde se describirán algunos escenarios que requerirán este tipo de capacidades de auto-adaptación descritas.

8 Bibliografía

- [1] SAE International, «Surface Vehicle Recommended Practice, J3015,» Septiembre, 2016.
- [2] ERTRACK Task Force "Connectivity and Automated Driving", «Automated Driving Roadmap (version 5.0),» Julio, 2015.
- [3] J. Fons, «Especificación de Sistemas auto-adaptativos,» Marzo, 2020. [En línea]. Available:
https://poliformat.upv.es/access/content/group/DOC_33953_2019/Notación-SAS.pdf.