Universidad de Oviedo
Universidá d'Uviéu
University of Oviedo

Escuela de Ingeniería Informática
Universidad de Oviedo

Proyecto de Desarrollo

AUTOMATIZACIÓN DE LOS INTERMITENTES DE VEHÍCULOS

MÁSTER EN INGENIERÍA WEB

TRABAJO DE FIN DE MÁSTER

AUTOR

David Sánchez Luis

DIRECTOR

Cristian González García

Junio 2021

Resumen

El avance de la tecnología de asistencia a la conducción está creciendo mucho en los últimos años, con la incorporación de vehículos autónomos, llegando incluso a realizar una conducción autónoma supervisada. Este trabajo plantea el estudio, viabilidad y solución de automatizar el uso de intermitentes, bien como asistencia a la conducción humana, o como utilización en sistemas de conducción autónoma como notificación al resto de conductores humanos.

Para resolver este problema se han capturado la velocidad y ángulo del volante durante la conducción con el objetivo de ser capaces de automatizar el uso de intermitentes basándonos en dichos datos.

Para tratar la información recogida se han utilizado diferentes ramas de la inteligencia artificial: redes neuronales y aprendizaje automático, con el fin de predecir cuándo hay que activar un intermitente. Se han construido diversas redes neuronales basándonos en la arquitectura CNN con diferentes parámetros de entrenamiento, así como un subconjunto de algoritmos de la rama de aprendizaje automático para estudiar qué combinación obtiene mejores resultados.

Índice de contenido

Índice de Figuras

Índice de Tablas

Escuela de
Ingeniería Informática
Universidad de Oviedo

Capítulo 1 PLANIFICACIÓN DEL SISTEMA DE INFORMACIÓN

FASE DE PLANIFICACIÓN

PSI

PSI 1: Inicio del Plan de Sistemas de Información

Hoy en día existen muchas soluciones tecnológicas cuyo objetivo principal es el aliviar al usuario de tener que realizar acciones o estar pendiente de una tarea repetitiva que se le puede olvidar. El presente Trabajo Fin de Máster busca analizar la viabilidad e implementación de la automatización de los intermitentes de un vehículo en movimiento.

PSI 1.1: Análisis de la Necesidad del PSI

Los coches autónomos han sido una de las revoluciones automovilísticas de la última década. Gracias a la incorporación de nuevos sensores y cámaras se han podido crear vehículos que apenas necesitan intervención humana para circular.

Diversas empresas como Nvidia, Tesla, Uber o Google han estado desarrollando y vendiendo coches autónomos [2] [3], llegando a crear flotas de taxis sin pilotos, sin embargo, todavía falta tiempo hasta que el 100% de los conductores sean reemplazados [4]. Por ello, además de vehículos autónomos, también hay que tener en cuenta posibles tecnologías que ayuden al conductor humano en diversos ámbitos, como aparcamiento automático, controles de crucero o seguridad pasiva.

Bajo este pretexto se presenta este Trabajo Fin de Máster, con el objetivo de detectar cuándo hay que poner o no los intermitentes, así como posibles usos derivados de saber cuándo hay que utilizarlos.

PSI 1.2: Identificación del Alcance del PSI

Los objetivos estratégicos para lograr que el proyecto sea un éxito son los siguientes:

- Analizar alternativas y soluciones ya existentes al problema planteado.
- Ser capaz de saber con cierto grado de antelación cuándo hay que activar un intermitente durante la conducción.

PSI 1.3: Determinación de Responsables

El proyectante se encargará del estudio, análisis e implementación de los módulos necesarios para alcanzar el objetivo.

El tutor se encargará de dar apoyo en ámbito de investigación, validar los módulos implementados por el proyectante, intentando desbloquear ante situaciones de análisis-parálisis y ofreciendo consejos y alternativas ante los problemas que vayan surgiendo.

PSI 2: DEFINICIÓN Y ORGANIZACIÓN DEL PSI

En esta sección se definirá el alcance, responsables y planificación del proyecto. Durante todas ellas se tendrá en cuenta el tiempo de aprendizaje de nuevas tecnologías y los recursos que serán necesarios para completar el proyecto.

PSI 2.1: Especificación del Ámbito y Alcance

Se han definido las siguientes fases por las que pasará el proyecto.

Fase 1: Estudio del estado del arte

Se analizarán las soluciones existentes y se investigarán diversos artículos de investigación relacionados con la conducción autónoma, detección de cambios de carriles, activación automática de intermitentes y cualquier otro tema que pueda aportar ideas y soluciones al problema original.

Objetivos de la fase:

- Obtener un listado de artículos de investigación relacionados con el tema.
- Analizar soluciones existentes que traten este mismo tema.
- Determinar qué posible información necesitamos obtener como mínimo para poder predecir el uso de los intermitentes.

Fase 2: Obtención de información

Se planteará la definición de un sistema hardware/software que sirva para extraer la información definida.

Objetivos de la fase:

- Obtener los datos necesarios y transformarlos a un formato definido con el que trabajaremos en fases posteriores.

Fase 3: Implementación de sistema predictivo

Se implementará el software en la tecnología necesaria, que, utilizando los datos de fases anteriores, sea capaz de automatizar el uso de intermitentes.

Objetivos de la fase:

- Obtener un modelo predictivo, intentando conseguir un alto porcentaje de acierto sobre un set de datos.

Fase 4: Servidor y clientes

Se cargará el modelo predictivo en un servidor y se plantearán los clientes necesarios a incluir en el vehículo para poder obtener predicciones en tiempo real.

Objetivos de la fase:

- Cargar un modelo predictivo en un servidor.
    - Ofrecer un servicio REST que se comunique con dicho modelo.
- Crear clientes que envíen los datos necesarios en tiempo real para poder obtener una predicción durante la conducción.

PSI 2.2: Organización del PSI

La única organización será con mi tutor. Se plantearán comunicaciones vía email, con alguna reunión presencial para validar el funcionamiento del proceso cuando proceda; en caso de necesidad se podrá plantear videollamadas para casuísticas concretas.

PSI 2.3: Planificación del proyecto

PSI 2.3.1: Identificación de interesados

Encontramos dos principales interesados:

1. Cristian González García: El presente tutor del Trabajo Fin de Máster que lanzó la propuesta al banco de trabajos públicos.
2. David Sánchez Luis: El alumno que busca finalizar los estudios de máster mediante el presente trabajo.

PSI 2.3.2: OBS, PBS

PSI 2.3.2.1: OBS

Al tratarse de un Trabajo Fin de Máster en el que solo encontramos a dos personas involucradas (tutor y ponente), sin haber terceras empresas o cotutores podemos definir una jerarquía muy sencilla en la que el ponente esté subordinado al tutor.

Aunque es cierto que se podría ampliar este marco englobando diferentes departamentos de la Universidad de Oviedo para darle una estructura mucho más clara, no es lo adecuado ya que ninguno de esos agentes formara parte del WBS.

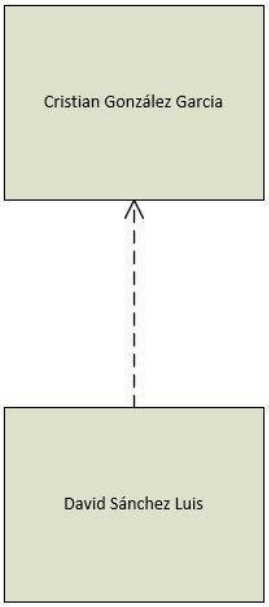


Ilustración 1 OBS

PSI 2.3.2.2: PBS

Como hemos orientado nuestro WBS en bloques entregables, el PBS puede ser creado en una relación casi directa.

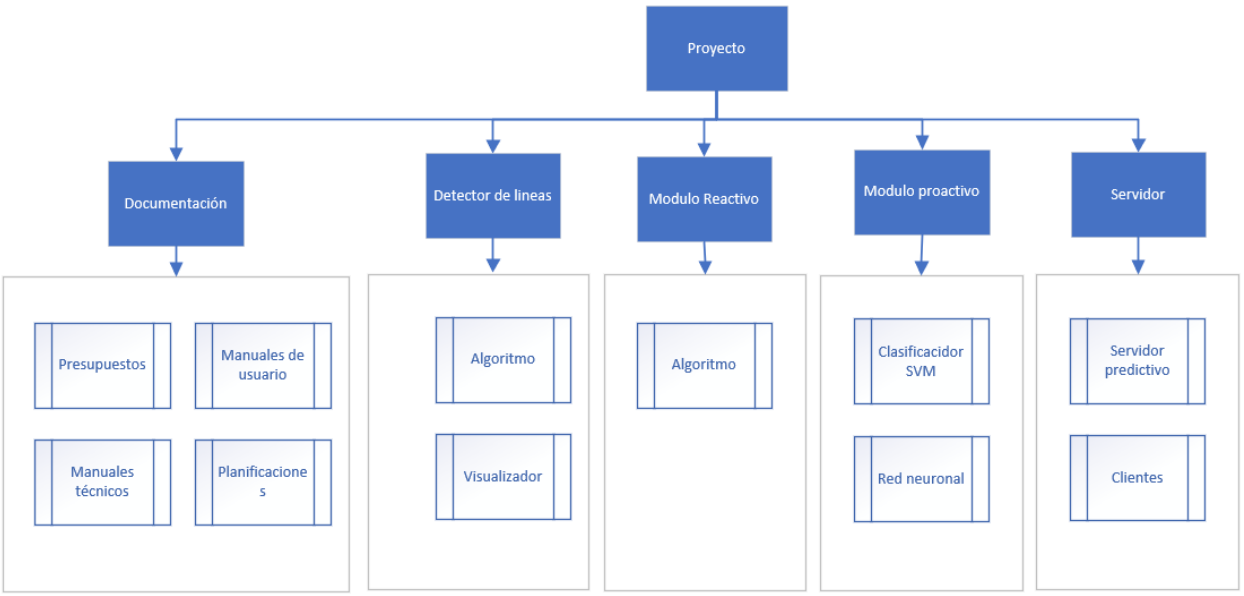


Ilustración 2 PBS

PSI 2.3.3: Planificación inicial. WBS, CBS

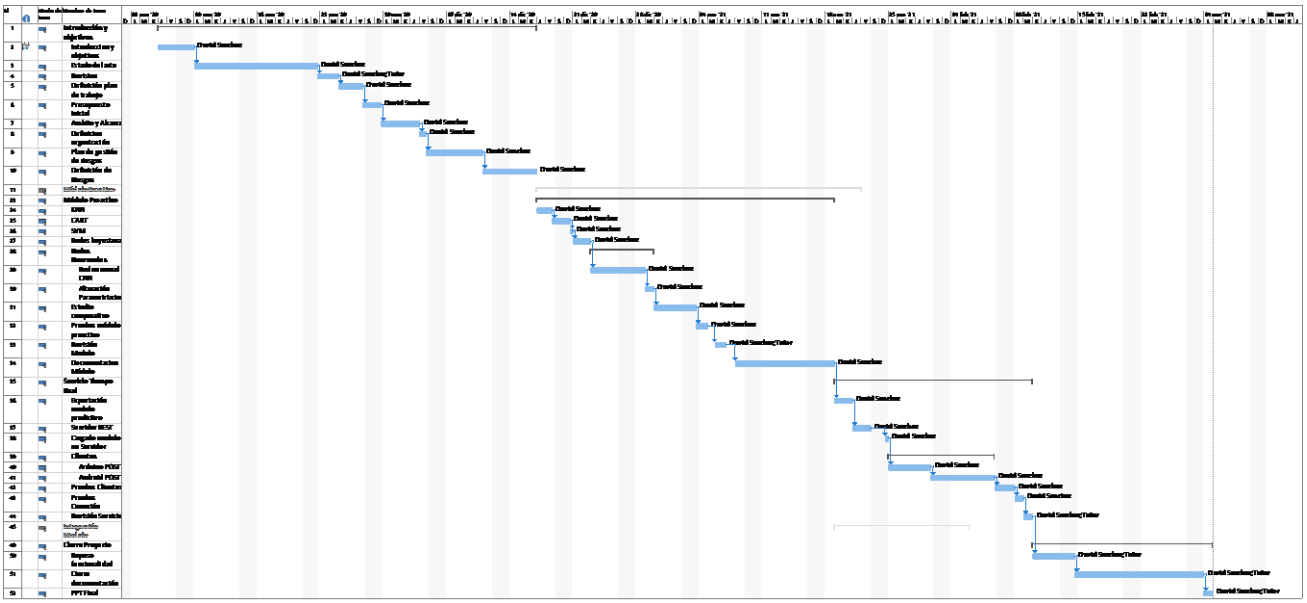


Ilustración 3 Diagrama de GANT

Las tareas mostradas en la Ilustración 3 Diagrama de GANT son las mismas que las detalladas a continuación. El propósito de dicha ilustración es mostrar el diagrama de GANT en sí. En los anexos se incluye una copia de los ficheros de planificación en caso de querer visualizarlo con alguna otra herramienta.

Tabla 1 WBS

Nombre de tarea	Duración	Comienzo	Fin	Predecesoras	Nombres de los recursos
Introducción y objetivos	**21,81 días**	**mié 04/11/20**	**mié 16/12/20**		
Introducción y objetivos	15 horas	mié 04/11/20	dom 08/11/20		David Sánchez
Estado del arte	40 horas	dom 08/11/20	dom 22/11/20	2	David Sánchez
Revisión	12 horas	dom 22/11/20	mar 24/11/20	3	David Sánchez; Tutor
Definición plan de trabajo	10 horas	mar 24/11/20	vie 27/11/20	4	David Sánchez
Presupuesto Inicial	4 horas	vie 27/11/20	dom 29/11/20	5	David Sánchez
Ámbito y Alcance	15 horas	dom 29/11/20	jue 03/12/20	6	David Sánchez
Definición organización	4 horas	jue 03/12/20	vie 04/12/20	7	David Sánchez
Plan de gestión de riesgos	20 horas	vie 04/12/20	jue 10/12/20	8	David Sánchez
Definición de Riesgos	20 horas	jue 10/12/20	mié 16/12/20	9	David Sánchez

Módulo Reactivo	**19,13 días**	**mié 16/12/20**	**jue 21/01/21**		
Captura de datos	12 horas	mié 16/12/20	dom 20/12/20	10	David Sánchez
Categorización manual de datos	16 horas	dom 20/12/20	vie 25/12/20	12	David Sánchez
Detección de líneas	4,75 días	vie 25/12/20	dom 03/01/21		
Revisión Detección de líneas	6 horas	dom 03/01/21	lun 04/01/21	17	David Sánchez
Detección de cambio de carril	20 horas	lun 04/01/21	dom 10/01/21	18	David Sánchez
Implementación visual	15 horas	dom 10/01/21	vie 15/01/21	19	David Sánchez
Pruebas detección	5 horas	vie 15/01/21	dom 17/01/21	20	David Sánchez
Documentación Módulo Reactivo	15 horas	dom 17/01/21	jue 21/01/21	21	David Sánchez
Módulo Proactivo	**17,19 días**	**mié 16/12/20**	**lun 18/01/21**		
KNN	5 horas	mié 16/12/20	vie 18/12/20	22	David Sánchez
CART	5 horas	vie 18/12/20	dom 20/12/20	24	David Sánchez
SVM	5 horas	dom 20/12/20	dom 20/12/20	25	David Sánchez
Redes bayesianas	5 horas	dom 20/12/20	mar 22/12/20	26	David Sánchez
Redes Neuronales	**3,94 días**	**mar 22/12/20**	**mar 29/12/20**		
Red neuronal CNN	20 horas	mar 22/12/20	lun 28/12/20	27	David Sánchez
Alteración Parametrizaciones	5 horas	lun 28/12/20	mar 29/12/20	29	David Sánchez
Estudio comparativo	10 horas	mar 29/12/20	dom 03/01/21	30	David Sánchez
Pruebas módulo proactivo	10 horas	dom 03/01/21	lun 04/01/21	31	David Sánchez
Revisión Modulo Proactivo	6,5 horas	mar 05/01/21	mié 06/01/21	32	David Sánchez; Tutor
Documentación Módulo Proactivo	40 horas	jue 07/01/21	lun 18/01/21	33	David Sánchez
Servicio Tiempo Real	**11,56 días**	**lun 18/01/21**	**mar 09/02/21**		
Exportación modelo predictivo	5 horas	lun 18/01/21	mié 20/01/21	34	David Sánchez
Servidor REST	8 horas	mié 20/01/21	vie 22/01/21	36	David Sánchez
Cargado modelo en Servidor REST	4 horas	dom 24/01/21	dom 24/01/21	37	David Sánchez

Clientes	**6,19 días**	**dom 24/01/21**	**vie 05/02/21**		
Arduino POST	16 horas	dom 24/01/21	vie 29/01/21	38	David Sánchez
Android POST	25 horas	vie 29/01/21	vie 05/02/21	40	David Sánchez
Pruebas Clientes	5 horas	vie 05/02/21	dom 07/02/21	41	David Sánchez
Pruebas Conexión Cliente-Servidor	5 horas	dom 07/02/21	lun 08/02/21	42	David Sánchez
Revisión Servicio	6,5 horas	lun 08/02/21	mar 09/02/21	43	David Sánchez; Tutor
Integración Módulo proactivo/Reactivo	**7,94 días**	**lun 18/01/21**	**mar 02/02/21**		
Integración Módulo proactivo/Reactivo	30 horas	lun 18/01/21	jue 28/01/21	34	David Sánchez
Pruebas integración	10 horas	jue 28/01/21	dom 31/01/21	46	David Sánchez
Revisión Integración	10 horas	dom 31/01/21	mar 02/02/21	47	David Sánchez
Cierre Proyecto	**10,31 días**	**mar 09/02/21**	**lun 01/03/21**		
Repaso funcionalidad	15 horas	mar 09/02/21	dom 14/02/21	44	David Sánchez; Tutor
Cierre documentación	73,5 horas	dom 14/02/21	dom 28/02/21	50	David Sánchez; Tutor
PPT Final	7,5 horas	dom 28/02/21	lun 01/03/21	51	David Sánchez; Tutor

PSI 2.3.4: Riesgos.

PSI 2.3.4.1: Plan de gestión de riesgos.

Contenido entregado en los anexos: Risk_Management_Plan

PSI 2.3.4.2: Identificación y registro de riesgos

Tabla 2 Identificación de riesgos

ID	Nombre del Riesgo	Breve Descripción	Categoría	Probabilidad	Impacto				Impacto	Respuesta al Riesgo	Estrategia
					Presup.	Planific.	Alcance	Calidad			
1	Resultados no válidos	Tras la implementación es posible obtener datos estadísticamente inservibles que no cumplan el objetivo del proyecto.	Técnico: Calidad	Media	Bajo	Bajo	Bajo	Crítico	0.45		Asumir el riesgo
2	No finalizar a tiempo	Debido a diferentes causas (internas o externas) es posible que el proyecto no se llegue a presentar a tiempo.	Gestión de Proyecto: Planificación y Control	Media	Medio	Crítico	Alto	Bajo	0.45		Asumir el riesgo
3	No aprobación de módulos por parte del tutor	Es posible que algunas de las implementaciones sean rechazadas por diversos motivos por el tutor, teniendo que retrabajar dicha parte.	Gestión del Proyecto: Comunicación	Baja	Bajo	Alto	Alto	Bajo	0.17	Definir de antemano qué es lo que se espera con cada uno de los módulos con el tutor, de la manera más detallada posible.	Mitigar el riesgo
4	Mala toma de requisitos	Realizar una mala toma de requisitos puede conllevar a realizar trabajo innecesario o tareas fuera del alcance.	Gestión del Proyecto: Comunicación	Media	Medio	Alto	Alto	Bajo	0,28	Definir de antemano qué es lo que se espera con cada uno de los módulos con el tutor, de la manera más detallada posible	Mitigar el riesgo

5	Uso de tecnología	Curva de aprendizaje alta para inteligencia artificial.	Técnico: Tecnología	Muy Alta	Medio	Alto	Alto	Alto	0.50	Dedicar tiempo extra a formación y lecturas sobre el estado del arte.	Mitigar el riesgo
6	Pérdida de información	Un fallo de hardware en el ordenador personal podría arruinar todo el desarrollo si se pierden ficheros.	Técnico: Prestaciones y fiabilidad	Muy Alta	Medio	Alto	Alto	Crítico	0.81	Se utilizará Microsoft OneDrive y Github como repositorios en la nube de copias de seguridad, así en caso de fallo de disco duro, se tendrá una copia actualizada de los archivos sin retrasar en exceso el proyecto ni perder información.	Eliminar el riesgo
7	Fracaso del proyecto	Suspender el proyecto y tener que repetirlo en otra convocatoria.	Técnico: Calidad	Baja	Inapreciable	Crítico	Inapreciable	Crítico	0.27		Asumir el riesgo
8	Time to Market	Que alguien saque un trabajo muy parecido al presente proyecto en el tiempo que se tarda en implementar y documentar.	Externo: Mercado	Media	Inapreciable	Inapreciable	Inapreciable	Crítico	0.45		Asumir el riesgo
9	Incumplir unas leyes desconocidas	Incumplir alguna ley al capturar de datos en un vehículo en conducción.	Externo: Regulación	Media	Alto	Inapreciable	Inapreciable	Inapreciable	0.28		Asumir el riesgo

10	Incapacidad de continuar económicamente proyecto	Si monetariamente no se pudiese continuar con el desarrollo del proyecto y se tuviesen que dedicar tiempo y esfuerzos en otras tareas para evitar deudas o pérdida de viviendas.	Organizacional: Financiación	Muy Baja	Crítico	Crítico	Crítico	Crítico	0.09	Tener un fondo reservado de dinero en exclusiva para la finalización de este proyecto.	Mitigar el riesgo
11	Accidente de trafico	Sufrir un accidente de tráfico durante la captura de datos.	Externo: Usuario	Muy Baja	Crítico	Crítico	Crítico	Crítico	0.09		Asumir el riesgo
12	Restricciones de movilidad	Que impidan desplazamientos en vehículo para capturar datos.	Externo: Regulación	Baja	Inapreciable	Crítico	Bajo	Bajo	0.27		Asumir el riesgo
13	Cambio de legislación	Un cambio de legislación en temas que afectan al proyecto como puede ser el procesado de datos capturados en tiempo real en un vehículo en conducción.	Externo: Regulación	Baja	Inapreciable	Inapreciable	Alto	Alto	0.17		Asumir el riesgo
14	Falta de definición de tareas	La posibilidad de obviar una tarea fundamental que haya que estimar y desarrollar a posteriori durante la ejecución del proyecto.	Gestión de Proyecto: Planificación	Media	Bajo	Alto	Crítico	Bajo	0.45	Definir de antemano cuál será el flujo final completo y los diferentes procesos para comprobar que no falta ninguno de los bloques en la planificación.	Mitigar el riesgo

15	Solicitud de cambios excesiva	Un gran número de cambios solicitados por el tutor o tras ver ciertas implicaciones técnicas que en un inicio se desconocían pueden aumentar la complejidad del proyecto y retrasarlo.	Gestión de Proyecto: Control	Baja	Medio	Alto	Alto	Bajo	0.17	Definir de antemano que es lo que se espera con cada uno de los módulos con el tutor de la manera más detallada posible.	Mitigar el riesgo

PSI 2.3.5: Presupuesto inicial

Para calcular el presupuesto se han tomado el número de horas necesarias de todas las tareas, así como el diferente material utilizado. También se han prorrateado los gastos mensuales que independiente del proyecto se tienen.

Estableciendo un precio hora de **45 euros** y sumando todos nuestros gastos obtenemos un coste total de **29540 euros**.

Como este presupuesto es un presupuesto interno y no algo que enseñaríamos a un cliente vamos a incluir un porcentaje de beneficio que queremos obtener, además de nuestro precio hora base. Queremos un **30% de beneficio sobre el precio total**, lo que lleva a un total (sin IVA) de 38402 euros. Antes de continuar, algunos autónomos deciden calcular el porcentaje de IRPF que les repercutirá ese año fiscal y repercutirlo al cliente de misma manera que estamos haciendo con el porcentaje de beneficio. Para ello habría que saber en qué tramo de IRPF estaremos aproximadamente ese año, que varía en función de la comunidad autónoma para calcular que impuestos derivarían del trabajo realizado.

En nuestro caso no vamos a repercutirlo en el presente presupuesto ya que realmente no se va a obtener ningún tipo de beneficio real.
Añadiendo el 21% de IVA correspondiente obtenemos un precio final de **46466.42 €.**

PSI 2.3.5.1: Presupuesto de costes.

Precio Hora: 45 euros

Tareas

Tabla 3 Presupuesto: Tareas

	Horas	Precio Total
Introducción y Objetivos	118	5310
Módulo Reactivo	128	5760
Módulo Proactivo	115	5175
Integración Módulos	50	2250
Cierre	102	4590
	Total	23085

Productos

Tabla 4 Presupuesto: Productos

	Precio	Unidades	Total
Arduino UNO	20	2	40
Conector potencia	3	2	6

MPU6050 Giroscopio	5	2	10
Arduino Lector SD	5	2	10
		Total	66

Gastos mensuales

Tabla 5 Presupuesto: Gastos mensuales

	€/Mes	Total
Luz	150	900
Agua	15	90
Internet	90	540
Alquiler	300	1800
Seguro coche	100	600
Gasolina	60	360
Mantenimiento	20	120
Material Oficina	10	60
	Total	4470

Software

Tabla 6 Presupuesto: Software

Windows 10	260
Project Profesional 2019	1509
Office Hogar	150
GPS Logger Android	0
Total	1919

Precio final

Tabla 7 Presupuesto: Precio final

Porcentaje Beneficio	30
Coste	29540
Beneficio	8862
Total Sin Iva	38402
IVA	0.21
TOTAL CON IVA	**46466.42**

Si este presupuesto fuese a presentarse a un cliente sería desglosado de diferente manera.

Contenido entregado en los anexos: Presupuesto Inicial

PSI 3: ESTUDIO DE LA INFORMACIÓN RELEVANTE

Actualmente existen marcas de vehículos, por lo general, tope de gama, que ofrecen la posibilidad de activar ayudas a la conducción, autónoma o no. Entre estas opciones puede estar el hecho de automatizar intermitentes [5]. Además, ya existen artículos científicos que tratan sobre este tema [6], por lo que se deduce que la creación de un sistema que los automatice es posible, la única diferencia que habrá será en porcentaje de aciertos y en los procedimientos o algoritmos utilizados para llegar al objetivo.

En nuestro caso, se busca crear un sistema que cumpla el objetivo. El vehículo del que se dispone es un Seat Córdoba Sedán del año 1996 sin casi ningún tipo de tecnología o datos que nos pueda proporcionar de manera automática.

PSI 3.1: Selección y Análisis de Antecedentes

Se parte de la base de la patente de Tesla para obtener una buena base de antecedentes y cómo funciona a nivel general un sistema que ya está en funcionamiento.

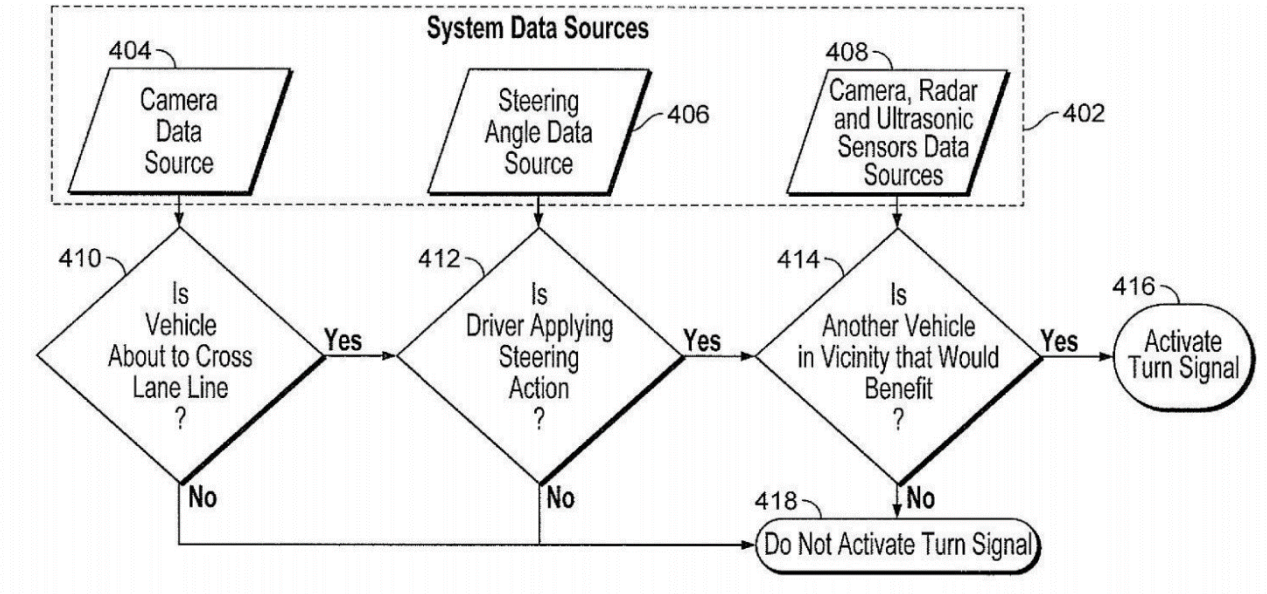


Ilustración 3 Patente de Tesla

A nivel general, se puede ver que se utiliza un sistema binario de toma de decisiones, en función de si se va a cruzar la línea de carril, si se está girando el volante y de si hay otros vehículos que se beneficien de activar los intermitentes.

Obviando que en la patente se están ignorando a los viandantes, la clave de su sistema son los elementos que aparecen en rombos. "¿Está el vehículo a punto de cruzar una línea?", esta pregunta se puede responder de diversas maneras [7] [8] [9], en función de que dato se quiera analizar: ¿Qué posición de la carretera ha estado observando el conductor durante los últimos 10 segundos?

[10]¿Cómo de cerca está el vehículo de los bordes del carril? ¿A qué velocidad está circulando? ¿Qué ángulo de giro está siguiendo el vehículo actualmente? Estas preguntas, entre otras, dan unos apuntes de diversos elementos en los que uno puede fijarse para ser capaz de establecer unos parámetros correctos para automatización.

Se pueden categorizar las posibles soluciones para activar vehículos en diferentes categorías:

PSI 3.1.1: Manualmente

- La manera tradicional, el conductor utilizando las palancas de cambio.

PSI 3.1.2 Algorítmicamente de manera reactiva

Para definir una solución algorítmica hay que definir una condición o serie de sucesos que se tienen que cumplir. Esta condición puede ser, por ejemplo, que un conductor tenga el volante girado 180º.

La gran mayoría de las veces, cuando se pisa una línea de carril se tiene que poner el intermitente. Por ello, el detectar cuando un vehículo cambia de carril sería una condición ideal para saber cuándo habría que indicar con intermitentes.

Hay muchos artículos de investigación que tratan este tema, hay dos conceptos clave a tratar:

- Detectar las líneas de carril. [11] [12].
- Cómo definir cuando se produce un cambio de carril. [13] [14].

PSI 3.1.3 Detectar las líneas de carril

Una de las técnicas más simples para detectar un carril en una carretera que involucra la utilización de procesado de imágenes es utilizar la transformada de *hough*. Esta es una técnica para encontrar formas y figuras en imágenes, en el caso de este proyecto, las líneas de carril. Esta técnica suele ir acompañada de diversas transformaciones previas o posteriores sobre la imagen que facilitan y mejoran los resultados obtenidos.

Sin embargo, no todas las carreteras tienen líneas de carril pintadas en el suelo. En estos casos se transforma la imagen y se busca una diferencia entre la carretera bien sea de tierra o pavimentada, con el terreno no circulable para poder delimitar la zona de conducción [15].

La otra opción más popular es utilizar redes neuronales. [16] Las opciones que encontramos con las diferentes arquitecturas de redes neuronales y *deep learning* son muy amplias, en *Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks* [17] dividen todas las soluciones en cuatro categorías:

- CNN codificador-decodificador
- *FCN con algoritmos de optimización*

- *CNN con redes neuronales recurrentes*
- *Redes generativas antagónicas*

La mayoría de las soluciones son diversas adaptaciones de redes neuronales convolucionales (CNN) que mejoran aspectos, como detección con baja visibilidad o eficiencia. Entre algunas de las soluciones en esta categoría encontramos *Cascaded CNNs* [11], Spatial CNN [8]. o SAD [18]

Table 4. Comparative results on BDD100K test set.

Algorithm	Accuracy	IoU
ResNet-18 [10]	30.66%	11.07
ResNet-34 [10]	30.92%	12.24
ResNet-101 [10]	34.45%	15.02
ENet [17]	34.12%	14.64
SCNN [16]	35.79%	15.84
R-18-SAD (ours)	31.10%	13.29
R-34-SAD (ours)	32.68%	14.56
R-101-SAD (ours)	35.56%	15.96
ENet-SAD (ours)	**36.56%**	**16.02**

Ilustración 4 Tabla comparativa entre diferentes soluciones. [20]

Spatial CNN es considerado, en el momento de escribir estas líneas, uno de los más modernos y precisos, con el que compiten nuevos modelos como SAD (Self Attention Distillation) que aprende sin supervisión. Podemos encontrar diversas implementaciones de SAD en Github [19].

Una red neuronal convolucionales está formada por diferentes elementos. El primero, la capa de entrada de datos. Después cuenta con un grupo de capas neuronales convolucionales, este grupo de capas son un número definido de neuronas interconectadas. Tras esta capa se encuentra un grupo de neuronas denominadas "pooling", el objetivo de esta capa es reducir, en comparación a la capa convolucional, el tamaño en neuronas. El conjunto de capa convolucional y pooling puede ser repetido varias veces. Finalmente, se encuentra una última capa de "clasificación", que proporciona el resultado.

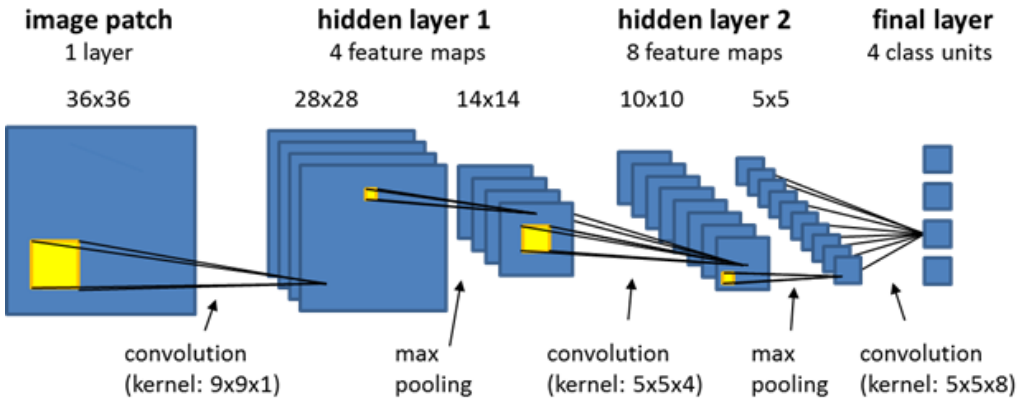


Ilustración 5 Ejemplo [20] de CNN con dos convoluciones

Cómo definir cuando se produce un cambio de carril

Una vez se ha localizado un carril, se puede trabajar en detectar en qué momento el vehículo se sale del mismo. La mayoría de las soluciones trabajan sobre la misma idea: crear un histograma de los carriles y detectar su posición respecto al centro, sabiendo así la posición relativa en la que nos encontramos respecto a los bordes. También se puede saber si el vehículo se está acercando a uno de los bordes y en qué momento se rebasa la línea del carril.

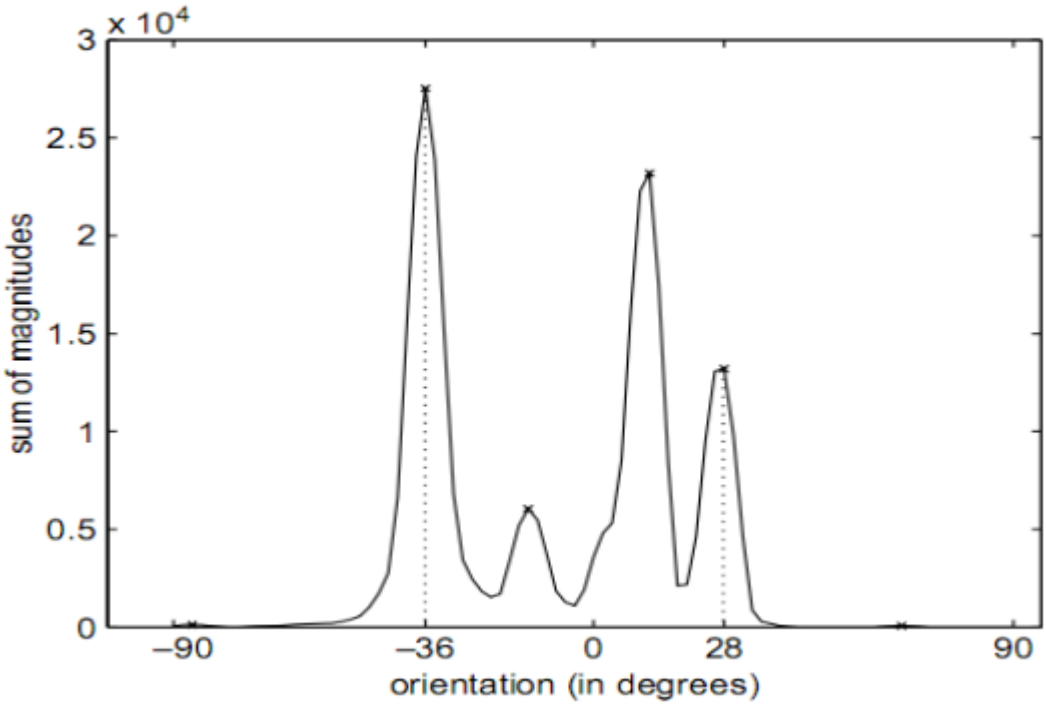
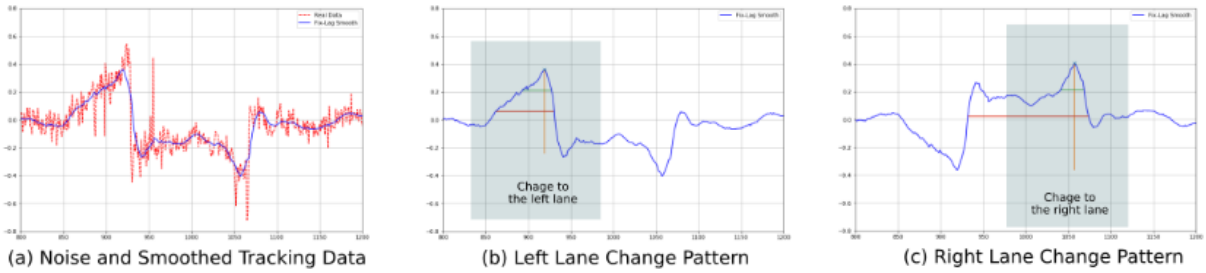


Ilustración 5 Figura obtenida de Lane change detection and tracking for a safe lane approach in real time vision based navigation systems [9]



(a) Noise and Smoothed Tracking Data (b) Left Lane Change Pattern (c) Right Lane Change Pattern

Ilustración 6 Figura obtenida de Driver Behavior Analysis Using Lane Departure Detection UnderChallenging Conditions [21]

Sobre esta idea, de manera adicional o alternativa se pueden recopilar y analizar de datos que son relevantes a la hora de activar intermitentes para utilizar redes neuronales [14]. Tres segundos antes de cualquier cambio de carril solemos mirar por los espejos para analizar el cambio [22], por ejemplo.

Los datos que se suelen recoger son:

- Proximidad y ángulo de las líneas del carril.
- Posición de los ojos.
- Velocidad y aceleración.
- Ángulo del volante [23] [24].

PSI 3.1.4 Legalidad y código de circulación

En España se considera legal la utilización de cámaras de salpicadero si se realiza durante la conducción y no se divulgan fotos o vídeos de personas o vehículos. Dejar una cámara grabando 24 horas al día sería ilegal.

Por ello los casos de uso de este proyecto se encontrarán amparados legalmente siempre y cuando se grabe solamente durante la conducción, de manera similar a la que ciertos fabricantes incluyen funcionalidades de vídeo que solo se activan en durante la conducción, como por ejemplo un sistema de aparcamiento automático.

PSI 4: DEFINICIÓN DE LA ARQUITECTURA TECNOLÓGICA

En este apartado se explicarán las diferentes alternativas encontradas, así como la solución que hemos decidido implantar inicialmente. Todos estos elementos podrán cambiar más adelante, incluso podrán ser reemplazados totalmente por otras soluciones en el caso de que se errase en su elección o haya cambio de requisitos.

PSI 4.1: Identificación de las Necesidades de Infraestructura Tecnológica

Se han detectado diferentes alternativas existentes que pueden solventar parte de las necesidades del proyecto.

PSI 4.1.1 Captura de datos

Como bien se mencionó anteriormente, hay diversos datos, como la velocidad, posición de los ojos o aceleración que pueden ser relevantes a la hora de discernir cuándo hay que activar intermitentes. Estos datos pueden obtenerse de diferentes maneras, pero a grandes rasgos hay tres categorías:

- Obtener información simulada.
- Obtener información directamente del vehículo.
- Obtener información alterando el vehículo.

PSI 4.1.1.1 Simulación

Obtener datos mediante simulación entra en la categoría de obtener datos de manera simulada. Para obtener los datos se utiliza una combinación de hardware con software que emulan la conducción. Algunos de ellos son software libre que puedes instalar en cualquier sistema operativo y manejar con teclado, puede que incluso con un volante USB. Otros son software propietario que utilizan hardware privativo, dependiendo de la empresa suministradora.

Hay varias ventajas de utilizar simuladores, da igual de que índole, para capturar información:

- La vida del conductor nunca corre peligro: Esto permite obtener datos de situaciones peligrosas o extrañas, como un accidente inminente, sin tener que depender de profesionales ni tener que arriesgar miles de euros y vidas ajenas.
- Personas sin vehículo: Aunque en España hay un 57% de personas que tienen permiso de circulación, no todas las personas tienen acceso a un vehículo que puedan utilizar. El uso de simuladores permite a cualquier persona recabar la misma información.
- Facilidad de recolección de datos: En el caso de que lo soporten, los simuladores permiten obtener datos de conducción fácilmente, solamente alterando parámetros de configuración.

Algunos de los simuladores más importantes son: NVIDIA Drive Sim [25]: Todavía en programa cerrado de acceso, seguramente requiriendo una tarjeta gráfica potente de la misma marca.

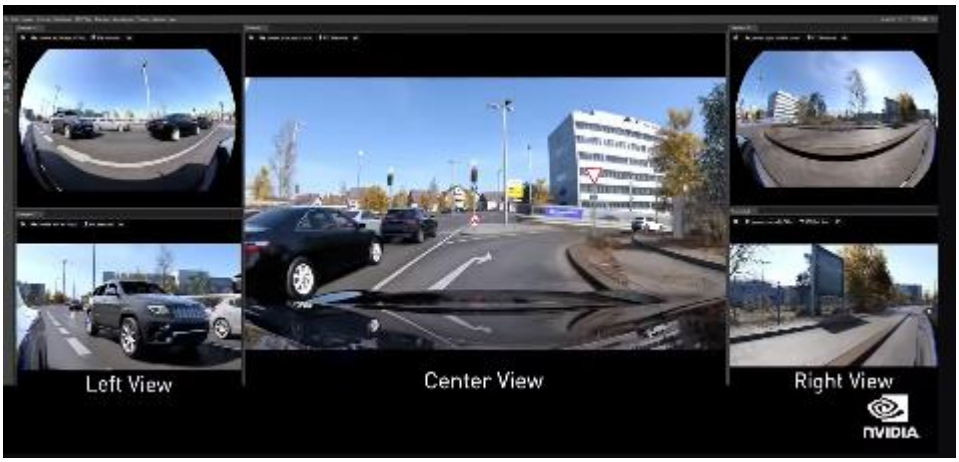


Ilustración 6 NVIDIA Drive Sim

CARLA [26]: CARLA es un simulador open source construido sobre Unreal Engine que permite incluir scripts de Python propios para añadir sensores, situaciones o capturadores de datos a la simulación, haciéndolo muy flexible.

Ilustración 7 Simulador CARLA de base y su extensión de sensor LIDARR

Lander Simulator [27]: Simulador cerrado que incluye muchos elementos físicos para asegurar una simulación lo más realista posible.

Ilustración 8 Simulando conducción con Lander Simulator

PSI 4.1.1.2 OBD-2

On Board Diagnosis. [28] Sistema de pines que permite obtener información del vehículo. Sigue unos estándares definidos en función del país. En el caso europeo denominado EOBD. Nos permite obtener como mínimo información definida en el estándar (velocidad, revoluciones del motor…) además de la información adicional que el fabricante de coches haya decidido incluir. No se puede obtener la información de cuando una persona activa los intermitentes.

PSI 4.1.1.3 Arduino

Arduino es una compañía de desarrollo de software y hardware libres. Uno de sus productos más importantes son las placas Arduino, también libres. Éstas son placas microcontroladoras con un lenguaje de alto nivel para interactuar con la misma.

Al tratarse de una placa abierta existen un número muy elevado de derivados, placas con un objetivo concreto creadas a partir de los diseños originales, o incluso tarjeas clónicas con las mismas características. Todas ellas compatibles con el mismo lenguaje de alto nivel. Esto ha permitido que existan tarjetas a muy bajo precio a costa de algunos materiales.

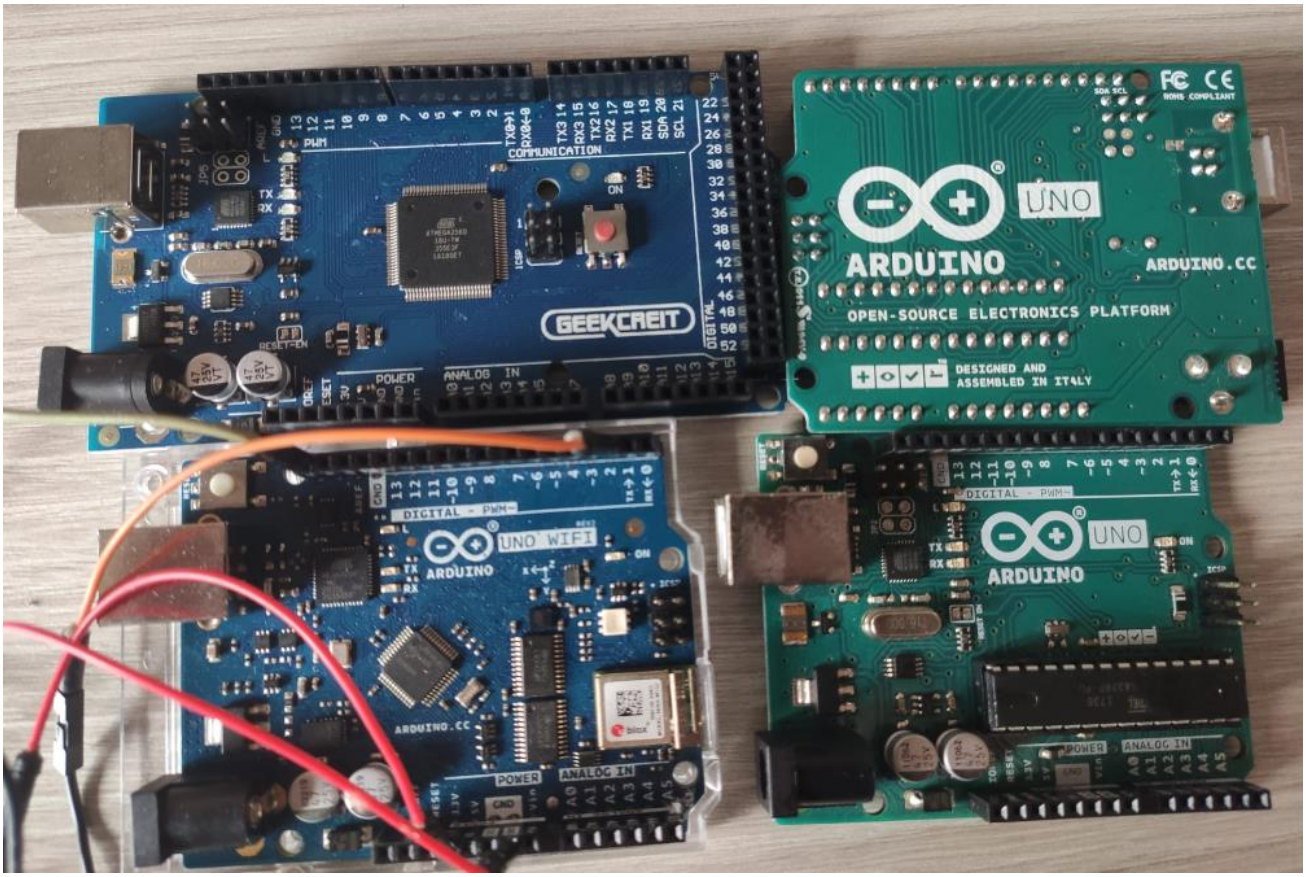


Ilustración 9 Dos Arduino UNO, una Arduino WiFiWifi y una Geekcreit MEGA 2560, basada en los diseños de Arduino MEGA

Estas placas ofrecen conexiones para cargar diversos sensores, como humedad, giroscopio, GPS… ofreciendo además una amplia biblioteca de librerías comunitarias para dichos sensores.

PSI 4.1.1.4 Raspberry Pi

Raspberry Pi, de manera similar a Arduino, es una placa de tamaño reducido, en este caso no abierta, capaz de ejecutar diferentes sistemas operativos. Tiene diferentes modelos con diferentes características y tamaños y no cuenta con clónicos.

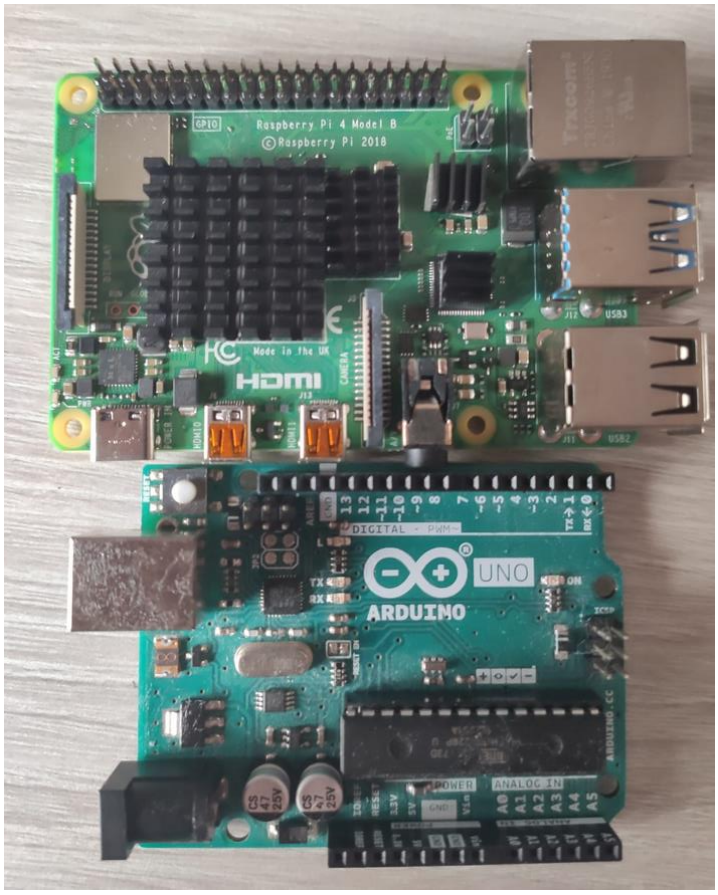


Ilustración 10 Comparativa de tamaños entre Arduino UNO y Raspberry Pi 4 B

Ilustración 11 Comparativa de entradas salidas entre Arduino UNO y Raspberry PI 4 B

Comparándola con la Arduino UNO, la Raspberry Pi 4 B ofrece más potencia computacional, diversas entradas USB pudiendo tener entradas de cámaras procesándolas en la propia placa y conectividad mucho más sencilla. Sin embargo, todos sus pines son digitales, a diferencia de la Arduino UNO u otras placas, que ofrecen diversas entradas analógicas.

PSI 4.1.1.5 Aplicaciones de móvil

Los móviles en los últimos años han ido evolucionando e incluyendo una gran cantidad de nuevos sensores, como por ejemplo el sensor *LiDAR* del iPhone.

Se pueden utilizar dichos sensores para recopilar información, tanto audiovisual, como GPS, velocidad, ángulos… También hay una gran selección de aplicaciones gratuitas.

PSI 4.1.2 Análisis y procesado de datos

Como se analizó en el estudio de la información relevante, se puede abordar el problema de diferentes maneras. En este punto se plantearán las diferentes soluciones tecnológicas de ambas categorías.

PSI 4.1.2.1 Modelo reactivo

Para implementar un modelo reactivo será necesario procesar imágenes.

PSI 4.1.2.1.1 Procesado de imágenes manual

La solución más manual, consiste en procesar los fotogramas grabados sin utilizar ningún tipo de librería.

PSI 4.1.2.1.2 OpenCV

OpenCV [29] es una biblioteca libre de visión artificial, disponible para diversos lenguajes de programación. Su uso está muy extendido teniendo incluso su propio StackOverflow [30] donde los usuarios pueden hacer preguntas. Aunque no tiene modelos específicos para trabajar elementos de conducción autónoma, sí que ofrece las implementaciones de las funcionalidades más típicas [30] para tratar vídeos o *streams* de vídeos. Estas implementaciones cuentan con su correspondiente documentación y ejemplos de uso. Además, la comunidad ha ampliado estas implementaciones a casos de uso concretos, pudiendo encontrar diversos tutoriales relacionados con la conducción autónoma (detectar carreteras, vehículos, señales…) fácilmente.

PSI 4.1.2.2 Modelo predictivo

Aprendizaje automático

Una de las ramas de la Inteligencia Artificial, que mediante utilización de algoritmos consigue extraer inferencias sobre el conjunto de datos. En nuestro caso a discernir cuándo hay que poner un intermitente.

Existen diferentes aproximaciones:

- Regresión Logística
- Análisis discriminante lineal
- KNN
- Aprendizaje basado en árbol de decisiones
- Redes Bayesianas

Sci-Kit

Sci-Kit [31] es una biblioteca en Python [32] de aprendizaje automático que ofrece ya implementados diferentes algoritmos parametrizables e Inter compatibles con NumPy [33].

NumPy es una biblioteca de funciones matemáticas de alto nivel implementada en C que además permite trabajar con matrices de gran tamaño rápidamente. Esto es especialmente útil trabajando con vídeos, pues cada fotograma de dicho video está compuesto de diversas matrices, del mismo tamaño que la resolución del vídeo, por lo que la rapidez para ser capaz de procesar y transformar dichas matrices es esencial si se quiere tener una respuesta rápida y seguir los cuadros por segundo del vídeo.

Sci-Kit además ofrece una buena documentación y ejemplos.

TensorFlow

TensorFlow [35] es una biblioteca de código abierto desarrollada por Google [36]. Tiene mucha documentación.

Un punto negativo que tiene TensorFlow es que han tenido muchísimas versiones API durante estos últimos años y la documentación está poco estructurada. Se pueden encontrar instancias en la documentación oficial para principiantes que utilizan tecnologías que abstraen del uso de TensorFlow, ya que puede ser poco legible inicialmente.

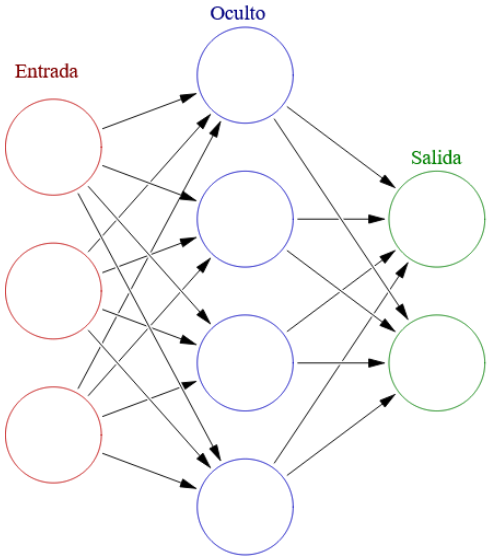
Keras

Keras [34] es una interfaz que sirve para interactuar con TensorFlow de manera sencilla e intuitiva. Es la interfaz que aparece en la documentación de TensorFlow para principiantes. Keras permite aprovecharse de permite utilizar tarjetas gráficas para realizar cómputos más rápidamente al usar TensorFlow internamente.

Keras permite abstraerse de las implementaciones, ya que ofrece interfaces con métodos parametrizables, evitando el tener que pensar cómo tener que implementar algo, pasando a pensar en que configuración es la óptima para la implementación. Por ello, es mucho más sencillo que utilizar TensorFlow directamente. Keras permite entrenar modelos estadísticos, así como redes neuronales.

Redes neuronales

Una red neuronal es un conjunto de neuronas interconectadas. Cada neurona recibe una serie de datos de entrada, bien proporcionados por datos o por otras neuronas, transforma dichos datos mediante alguna función y propaga el resultado.



En función de cuántos nodos, y cómo las conectemos obtendremos diferentes resultados. Aunque podríamos crear conexiones aleatoriamente, existen arquitecturas de redes neuronales [35] que nos pueden dar una guía base sobre la que iterar. Sobre estas bases podemos alterar cuantas capas, intermedias añadimos, numero de neuronas de una capa y cuantas interconexiones hay entre cada capa.

Tras haber creado una red neuronal, lo más habitual es alterar el valor de la función de transformación de cada neurona de manera automática, mediante entrenamiento, para obtener resultados más exactos.

Ilustración 12 Ejemplo de red neuronal de una capa

PyTorch

PyTorch es la implementación en Python de la biblioteca Torch. Además de Python, ofrece una interfaz en C++. Ha ganado mucha popularidad en ámbitos científicos y académicos y se le considera un rival directo de TensorFlow.

PSI 4.1.3 Integración en vehículo

La estrategia para implantar el proyecto en un vehículo real puede seguir las siguientes alternativas.

PSI 4.1.3.1 Módulo Independiente

Una de las posibilidades es crear un módulo independiente, esto significa tener algo que podamos quitar y poner de manera rápida y sencilla. La idea de crear este tipo de módulo es no realizar modificaciones que requieran alterar la estructura, cableado o electrónica del vehículo, ya que entonces la solución creada solo serviría para un solo tipo de vehículo, además de implicar una certificación y acreditación de la modificación realizada en la ITV [39], una revisión periódica sobre

los vehículos y su estado, en el que se determina si las condiciones del vehículo le permiten continuar circulando o no.

Para crear este módulo independiente se podría utilizar una de las placas microcontroladoras mencionadas anteriormente, almacenándolas en cualquier hueco del vehículo.

PSI 4.1.3.2 Integración parcial/total

En contraposición a crear un módulo independiente del vehículo, la idea de esta solución es integrar los módulos directamente en el vehículo, con posibles conexiones con el ordenador central del coche pudiendo recibir datos directamente de los sensores que posea el vehículo, teniendo acceso a la batería del coche y pudiendo llamar a funciones que ofrezca el vehículo, si es moderno.

Esta solución sería totalmente dependiente del modelo de vehículo, quizá siendo generalizable a la marca de fabricación, si es que esta sigue los mismos estándares entre todos los modelos que ofrece. Como no somos fabricantes, estas alteraciones deberán ser aprobadas mediante la correspondiente homologación.

PSI 4.2: Selección de la Arquitectura Tecnológica

En esta opción se definirán que soluciones de las planteadas previamente utilizaremos para desarrollar el proyecto.

PSI 4.2.1: Captura de datos

Para la captura de datos se va a utilizar un prototipado en Arduino, pues es mucho más barato, rápido y simple de implementar que en Raspberry. Una vez se obtengan unas conclusiones claras se hará un análisis a posteriori sobre si migrar a Raspberry sería sensato o no, analizando posibles problemas encontrados. Si fuese necesario, se complementarían estos datos con datos aplicaciones móviles gratuitas o, en su defecto baratas, para reducir tiempo de desarrollo.

La captura de datos de posición y velocidad se planteará utilizando el formato GPX [40], Formato de Intercambio GPS. Un formato estándar almacena la información de la posición, velocidad, trayecto, altura, dirección y hora.

PSI 4.2.2: Modelo predictivo

Aunque se podría utilizar un sistema reactivo para determinar cuándo poner intermitente, dicha activación de las luces para indicar la maniobra dejaría poco o ningún tiempo de reacción al resto de conductores, nunca indicando ni anticipando la maniobra como recomienda la DGT [40]; organismo estatal del Estado Español encargado de gestionar y establecer las políticas de las vías españolas.

Por ello se hará una comparativa de aprendizaje automático y redes neuronales para ver cuál es más exitoso o si se pueden utilizar de manera conjunta para cumplir el objetivo del proyecto. En cuanto

a las tecnologías a utilizar, se ha decidido utilizar Python como lenguaje, pues la gran mayoría de soluciones trabajan sobre él y se pueden realizar *scripts* de manera simple y sencilla.

Para aprendizaje automático se ha optado por usar SciKit, pues ya tiene plantillas con los diferentes algoritmos.

Para redes neuronales se ha decidido utilizar Keras sobre TensorFlow. Su interfaz de uso es realmente sencilla. Al disponer de una tarjeta gráfica ambas soluciones se aprovecharían de poder usar CUDA. En cualquier caso, se anotará en el apartado de conclusiones la opinión y resultados obtenidos al utilizar la tecnología seleccionada una vez comience a utilizarla.

Para cualquier otro desarrollo, y por coherencia, se utilizará Python para resolver la problemática.

PSI 4.2.3: Integración con vehículo

Finalmente, en cuanto a la integración en el vehículo; se dispone de un Seat Córdoba Sedán de 1995.

Ilustración 13 Seat Córdoba Sedán disponible y su interior

A nivel técnico, sería posible modificarlo para leer e interactuar con cualquiera de la poca tecnología que posee, no es lo óptimo para este tipo de experimento. En el caso de que se obtenga buenos resultados y/o estuviésemos en a la industria automovilística, quizá si interesase comenzar a integrar los resultados de este trabajo. Sin embargo, al ser personas individuales que tienen que pasar la inspección técnica de vehículos para cualquier modificación, lo más sencillo es un módulo independiente que coloquemos para obtener datos y retiremos para un uso diario.

Escuela de
Ingeniería Informática
Universidad de Oviedo

ASTURIAS
CAMPUS DE EXCELENCIA
INTERNACIONAL
AD FUTURUM

Capítulo 2 Análisis del Sistema de Información

Fase de Desarrollo

ASI

ASI 1: DEFINICIÓN DEL SISTEMA

Determinación del Alcance del Sistema

Como se ha definido previamente, el objetivo del presente trabajo es demostrar que se puede automatizar el uso de los intermitentes. Sin embargo, y para evitar problemas regulatorios, certificados y burocracia diversa, se han definido los siguientes límites en cuanto a alcance.

Nunca se llegará a activar el intermitente de verdad, ni puenteando la presente placa del vehículo ni activándolo con algún automatismo, se planteará un sistema que diga cuando habría que ponerlo de manera inicial.

No se plantea por el momento recoger múltiples datos de diferentes vehículos y alimentar una misma base de datos de diferentes conductores, así como no generar modelos personalizados para cada conductor.

ASI 2: ESTABLECIMIENTO DE REQUISITOS

Se han definido diferentes categorías en las que distribuiremos los requisitos.

ASI 2.1: Obtención de los Requisitos del Sistema

Requisitos: seguridad

Tabla 8 Requisitos de seguridad

CÓDIGO	***NOMBRE REQUISITO***	***DESCRIPCIÓN DEL REQUISITO***
RS1	Seguridad Vial	El sistema no debe interferir con una conducción segura según la DGT, tanto en visibilidad del conductor como en maniobrabilidad.

Requisitos: captura de datos

Tabla 9 Requisitos de captura de datos

CÓDIGO	***NOMBRE REQUISITO***	***DESCRIPCIÓN DEL REQUISITO***
RC1	Captura de datos	El sistema debe de ser capaz de capturar datos de conducción.
RC1.1	Captura de datos	El sistema debe enviar los datos a un servidor para obtener una respuesta en tiempo real.

RC1.2	Frecuencia de datos	Los datos deben tomarse o enviarse como mínimo cada segundo de conducción.
RC1.3	Almacenamiento de datos	El sistema debe permitir almacenar dichos datos para su posterior tratado.
RC1.3.1	Almacenamiento de datos	El sistema almacenará los datos en una tarjeta SD.
RC2	Dato mínimo obligatorio	En todas las capturas o envíos de datos será necesario incluir la hora.
RC3	Cambio Modo	Se debe poder cambiar el funcionamiento de la Arduino: En un modo se capturarán datos físicamente en un fichero. En otro modo se enviarán peticiones REST.

Requisitos: captura del ángulo del volante

Tabla 10 Requisitos de captura de datos: volante

CÓDIGO	***NOMBRE REQUISITO***	***DESCRIPCIÓN DEL REQUISITO***
RCA1	Ángulo de volante	Se capturará el ángulo del volante. Siendo 0º una conducción en línea recta.
RCA2	Ángulo del volante. Datos	Los datos serán almacenados en fichero CSV, representando cada entrada de CSV una medición a lo largo del tiempo.
RCA3	Ángulo del volante. Arquitectura	El sistema de detección del ángulo del volante se implementará sobre una placa Arduino con los sensores de giroscopio, escritor de SD y reloj.
RCA4	Envío del ángulo del volante	La Arduino enviará el dato de cuál es el ángulo del volante vía Internet a un servidor.

Requisitos: captura de la velocidad

Tabla 11 Requisitos de captura de datos: velocidad

CÓDIGO	***NOMBRE REQUISITO***	***DESCRIPCIÓN DEL REQUISITO***
RCS1	Velocidad/Posición	Se capturará la velocidad y la posición a la que circula el vehículo.
RCS2	Velocidad/Posición	El formato para almacenar esta información será GPX.
RCS3	Velocidad	Al enviar la información al servidor, se mandará solamente la velocidad en km/h.
RCS4	Posición	Para capturar la posición inicial se utilizará “GPS Logger” *[36]*.
RCS5	Envío de velocidad	Para el envío de la información se creará una aplicación móvil.
RCS6	Aplicación móvil	La aplicación móvil se programará para Android.

Requisitos: Generación de modelos predictivos

Tabla 12 Requisitos de generación de modelos predictivos

CÓDIGO	NOMBRE REQUISITO	DESCRIPCIÓN DEL REQUISITO
RGM1	Generación de fichero de entrenamiento	El sistema generará un CSV de entrenamiento.
RGM1.1	Generación de fichero de entrenamiento	Para poder generar dicho fichero será necesario utilizar el fichero GPX y el fichero de ángulo de volantes indicados en requisitos anteriores.
RGM1.2	Generación de fichero de entrenamiento	El CSV Generado incluirá las siguientes columnas: hora, velocidad, intermitente.
RGM1.3	Generación de fichero de entrenamiento	La columna de intermitente estará vacía y deberá ser rellenada por una persona, indicando cuando hay que activar un intermitente con valores 1,0,-1.
RGM 1.4	Generación modelo	El sistema generará un modelo predictivo utilizando los ficheros GPX, ángulo de volante y fichero de entrenamiento.
RGM 1.5	Modelo predictivo	El modelo predictivo se basará en una red neuronal CNN.
RGM 1.6	Modelo predictivo	El modelo será exportado a un fichero para su posterior utilización en el servidor.
RGM 1.7	Modelo predictivo	El modelo deberá responder, dada la entrada de ángulo y velocidad, con un valor que represente si hay que activar un intermitente.

Requisitos: Servidor

Tabla 13 Requisitos de servidor

CÓDIGO	NOMBRE REQUISITO	DESCRIPCIÓN DEL REQUISITO
RSER 1	Tecnología Servidor	El servidor será desarrollado en Python.
RSER 2	REST	El servidor tendrá disponibles dos servicios REST.
RSER 2.1	REST Velocidad	Uno de los servicios REST tendrá el endpoint /speed.
RSER 2.2	Parámetros Velocidad	El servicio REST recibirá los parámetros “time” y “speed”.
RSER 2.2.1	Parámetro time	El parámetro time tendrá formato “yyyy.MM.dd HH:mm:ss”
RSER 2.2	REST Ángulo	Uno de los servicios REST tendrá el endpoint “/angle”.
RSER 2.2.1	Parámetros Ángulo	El servicio REST recibirá los parámetros “time” y “angle”.
RSER 2.2.2	Parámetro time	El parámetro time tendrá formato de *“Unix time”.*

Requisitos: Clientes

Tabla 14 Requisitos de clientes

CÓDIGO	*NOMBRE REQUISITO*	*DESCRIPCIÓN DEL REQUISITO*
RCLI 1	Cliente Velocidad	Se implementará una aplicación Android que envíe peticiones al servidor *[36]*.
RCLI 1.1	Comenzar/Parar	La aplicación contará con un botón para comenzar y parar el envío de información.
RCLI 1.2	Comenzar/Parar	La aplicación **nunca** deberá enviar información sin haberle solicitado el comienzo del envío.
RCLI 1.3	Visualización	La aplicación mostrará la velocidad actual en la pantalla.
RCLI 1.4	Permisos	La aplicación solicitará los permisos mínimos para poder funcionar: Localización e Internet.
RCLI 1.5	Frecuencia	Se enviarán los datos cada segundo siempre que se haya activado la opción de enviar información.
RCLI 2	Cliente Angulo	Se implementará una aplicación Arduino que envíe peticiones al servidor.
RCLI 2.1	Posición	Este cliente debe cumplir los requisitos de seguridad establecidos en RS1.
RCLI 2.2	Frecuencia	Se enviará la información del ángulo del volante cada segundo.

ASI 2.2: dentificación de Actores del Sistema

Para la instalación, configuración y correcto funcionamiento del sistema se han detectado diferentes perfiles que participan en la puesta a punto y mantenimiento del sistema.

Estos perfiles representan una distribución lógica de tareas. Durante la ejecución del proyecto todos estos perfiles serán cubiertos por una sola persona, pero en el caso de que el proyecto fuese llevado a cabo por múltiples personas, dichos perfiles podrían haber estado cubiertos por equipos especializados.

Los perfiles detectados son los siguientes:

- Administrador: Encargado de iniciar, mantener y configurar los dispositivos y servidores, asegurando que todas las peticiones se respondan de manera correcta.
- Científico de datos: Encargado de obtener y analizar datos de conducción, así como de generar los modelos predictivos utilizados en los servidores.
- Conductor: Este perfil representa al usuario final del sistema. Se beneficia pasivamente del sistema, conduciendo.

ASI 2.3: Especificación de Casos de Uso

A partir de los perfiles previamente identificados, se han detectado los casos de uso que cada uno de ellos deberá cumplir.

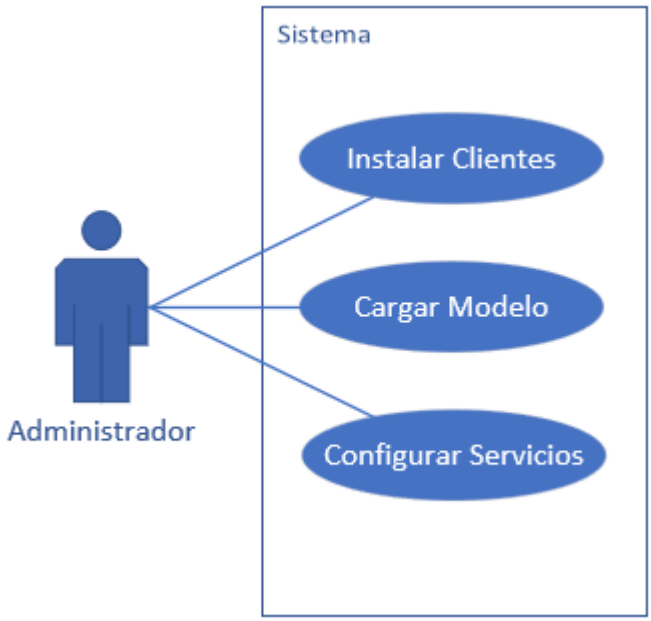


Ilustración 14 Casos de uso: Administrador

Nombre del Caso de Uso
Instalar Clientes
Descripción
El administrador instalará los dispositivos de captura de datos en el vehículo. Esto supondrá colocar y configurar los sensores y placas Arduino necesarias para que se envíen peticiones de manera correcta al servidor. Esta instalación deberá cumplir los requisitos de seguridad para una conducción segura.

Nombre del Caso de Uso
Cargar Modelo
Descripción
El administrador cargará un modelo predictivo en el servidor que proporciona respuestas a las peticiones que realizan los clientes. Dicho modelo predictivo será proporcionado por el perfil de científico de datos.

Nombre del Caso de Uso
Configurar Servicios
Descripción

El administrador configurará el servidor y los servicios REST de manera que puedan ser utilizados por los clientes, asegurándose de que hay una correcta conectividad.

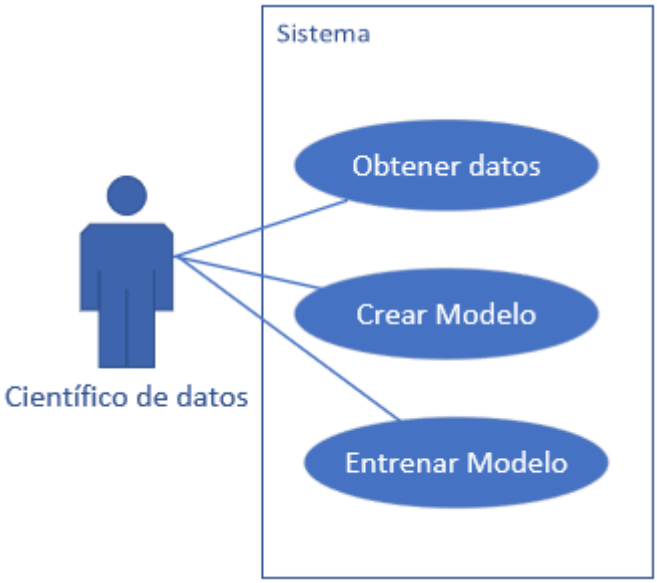


Ilustración 15 Casos de uso: Científico de datos

Nombre del Caso de Uso
Obtener Datos
Descripción
El tener datos de calidad es un paso clave para poder analizar y entrenar modelos predictivos que tengan un alto porcentaje de acierto inicial. Para poder capturar datos estructurados en nuestro sistema hay que realizar ciertas modificaciones a los clientes (Arduino y Android) para que, en lugar de realizar peticiones vía internet, se almacenen localmente para su posterior análisis, procesado y explotación.

Nombre del Caso de Uso
Crear Modelo
Descripción
La iteración en modelos predictivos incrementando la precisión con ciertos ajustes es algo habitual. Con este caso de uso el perfil de científico de datos deberá poder crear nuevos modelos para cargarlos posteriormente en el sistema.

Nombre del Caso de Uso
Entrenar Modelo
Descripción
Se deberán poder entrenar los modelos generados.

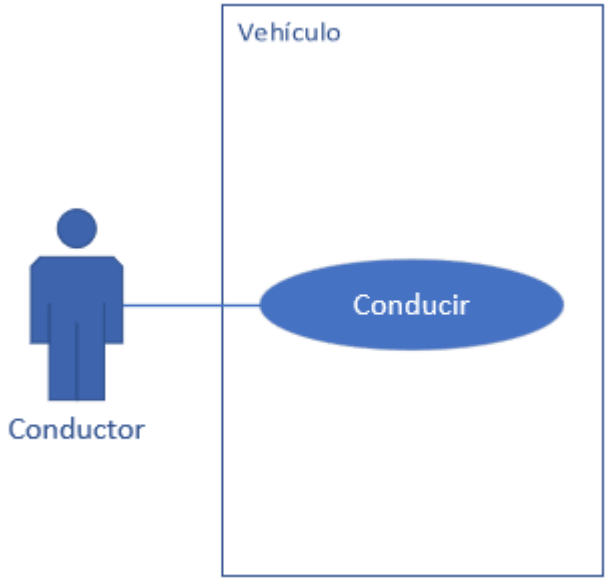


Ilustración 16 Casos de uso: Conductor

Nombre del Caso de Uso
Conducir
Descripción
Para predecir el uso de intermitentes un conductor solamente tendrá que conducir como hace siempre, y sin ninguna otra intervención adicional, los sistemas instalados en el vehículo se comunicarán con el servidor enviando los datos necesarios.

ASI 3: IDENTIFICACIÓN DE SUBSISTEMAS

A partir de los casos de uso anteriores podemos identificar claramente 3 subsistemas.

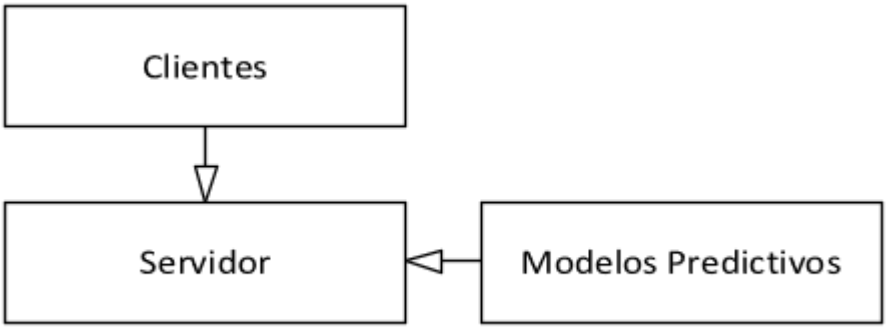


Ilustración 17 Subsistemas

ASI 3.1: Subsistema de clientes

Este subsistema contendrá todos los elementos hardware, software y sus configuraciones necesarias para capturar y enviar datos.

ASI 3.2: Servidor

Este subsistema contendrá todo lo necesario para servir peticiones a clientes y poder cargar modelos predictivos, estos serán generados por el subsistema generador de modelos predictivos, y serán cargados manualmente por el administrador.

ASI 3.3:Generador de modelos Predictivos

Este subsistema contendrá todo lo necesario para cargar datos de entrenamiento, así como generación y entrenamiento de modelos Keras.

ASI 3.4: Descripción de los Interfaces entre Subsistemas

Los clientes enviarán datos al servidor vía peticiones REST. Existirán dos interfaces: "/speed" y "/angle", que aceptarán los siguientes parámetros:

- Time: *timestamp* de la fecha y hora cuando se realizó la petición
- Speed: velocidad en kilómetros por hora.
- Angle: Ángulo del volante en grados.

Se proporciona el siguiente ejemplo de los datos de entrada:

Petición Arduino

time=1618087748

angle=90

Petición Android

time=1618087748

speed=60.5

ASI 4: ANÁLISIS DE LOS CASOS DE USO

INSTALAR CLIENTES	
PRECONDICIONES	El administrador tiene acceso a un vehículo.
POSCONDICIONES	El vehículo tiene instalados los clientes que obtienen velocidad y ángulo del volante.
ACTORES	Administrador
DESCRIPCIÓN	El administrador coloca el detector de ángulo de volante con el giroscopio en el volante. El administrador instala la aplicación Android de captura de velocidad. El administrador comprueba que la conducción no se vea afectada.
EXCEPCIONES	Algunos de los dispositivos no tienen el comportamiento esperado. El sistema instalado impide una conducción segura. El administrador notifica la incidencia.

CARGAR MODELO	
PRECONDICIONES	El administrador tiene un modelo a cargar.
POSCONDICIONES	El sistema utiliza el modelo cargado para dar respuesta.
ACTORES	Administrador
DESCRIPCIÓN	El administrador almacena el modelo generado en el servidor. El administrador indica al servidor que utilice el nuevo modelo.

CONFIGURAR SERVIDOR	
PRECONDICIONES	El modelo predictivo está cargado en el servidor.
POSCONDICIONES	El sistema está operativo y puede responder a peticiones entrantes.
ACTORES	Administrador
DESCRIPCIÓN	El administrador configura la red. El administrador configura el firewall. El administrador configura otros parámetros. El administrador comprueba el correcto funcionamiento del servidor.
EXCEPCIONES	El servidor no presenta el comportamiento esperado. El administrador notifica la incidencia.

OBTENER DATOS INICIALES	
PRECONDICIONES	Los clientes están instalados.
POSCONDICIONES	Los ficheros han sido generados correctamente.
ACTORES	Científico de datos.
DESCRIPCIÓN	El científico de datos conduce durante un periodo de tiempo que permita obtener una cantidad necesaria de datos para el entrenamiento inicial del modelo.
EXCEPCIONES	No se han generado los ficheros o estos tienen los datos almacenados en un formato incorrecto.

CREAR MODELO	
PRECONDICIONES	-
POSCONDICIONES	Un modelo ha sido implementado.
ACTORES	Científico de datos.
DESCRIPCIÓN	El científico de datos crea un modelo predictivo, pudiendo cambiar diversos parámetros de configuración o arquitectura principal del mismo.

ENTRENAR MODELO	
PRECONDICIONES	El científico de datos tendrá los ficheros de datos necesarios.
POSCONDICIONES	El científico de datos habrá generado un modelo predictivo.
ACTORES	Científico de datos.
DESCRIPCIÓN	El científico de datos realiza la clasificación inicial de datos manualmente. El científico de datos entrena el modelo con todos los datos. El científico de datos almacena en un fichero el modelo resultante.

CONDUCIR	
PRECONDICIONES	Los clientes están instalados.
POSCONDICIONES	El servidor ha recibido las peticiones.
ACTORES	Conductor.
DESCRIPCIÓN	El conductor conduce de forma habitual. Los clientes envían peticiones al servidor de forma continua durante la conducción.
EXCEPCIONES	El servidor no ha recibido el número de peticiones esperadas.

ASI 5: ANÁLISIS DE CLASES

ASI 5.1: Diagrama de Clases

Se han dividido los diagramas de clases de acuerdo con los subsistemas encontrados:

Modelo Predictivo

Para crear un nuevo modelo predictivo hay que implementar: la interfaz NeuralNetwork y el modelo. Este modelo predictivo es el que se entrenará a partir de los tres ficheros de información:

- Arduino: Fichero indicando la marca de tiempo y el ángulo del volante.
- GPXParser: Fichero GPX con la ruta seguida para calcular la velocidad en cada punto del tiempo.
- Training: fichero indicando cuando hay que poner el intermitente en cada marca de tiempo, o *timestamp*.

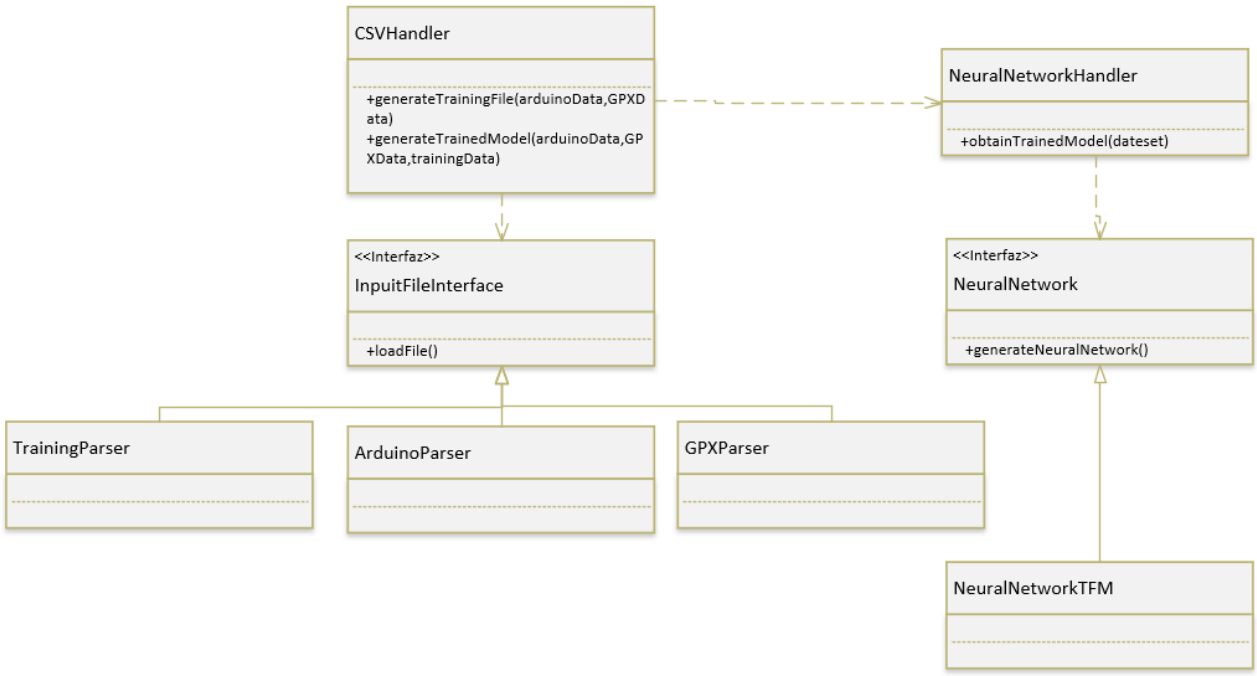


Ilustración 18 Diagrama de clases: modelo predictivo

- Interfaz NeuralNetwork: Define un único método para devolver modelos predictivos.
- NeuralNetworkTFM: Implementa NeuralNetwork, devolviendo una red neuronal de tipo CNN totalmente conectada con un porcentaje de *dropout*. El porcentaje de dropout, en castellano

"dilución" permite ignorar ciertas conexiones de la red neuronal, dependiendo del porcentaje indicado, alterando así los resultados y precisión de la red neuronal.

- NeuralNetworkHandler: Clase con una instancia NeuralNetwork obtenida por patrón Strategy.
    - obtainTrainedModel(Dataset): Entrena el modelo predictivo elegido con el conjunto de datos (*Dataset*) que se le pasa como parámetro.
- TrainingParser: Define como se tiene que cargar los datos de entrenamiento suministrados por el científico de datos. Para ello separará el CSV proporcionado para retornar un diccionario de hora-intermitente.
- ArduinoParser: Define como se tiene que cargar los datos de ángulo de volante. Separar el fichero líneo a línea para cargar retornar un diccionario de hora-ángulo.
- ArduinoParser: Define como se tiene que cargar un fichero GPX. Si los puntos del trayecto GPX tienen velocidad, la utilizará, si no, la calculará basándose en distancia recorrida/tiempo. Retornará un diccionario de hora-velocidad.
- CSVHandler: El encargado de interconectar los ficheros proporcionados, unificarlos en un solo conjunto de datos y entrenar el modelo predictivo implementado.

Clientes

Ilustración 19 Diagrama de clases: Clientes

Android

El objetivo de la aplicación Android es enviar la velocidad actual del vehículo al servidor. Para ello se calculará utilizando la ubicación del dispositivo. Se hará un clonado de la aplicación GPSLogger [36] y se incluirán los métodos definidos:

- createLocationRequest: Solicitar obtener la ubicación actual. En el caso de que la aplicación contenga un método similar, se reutilizará.
- locationCallback: Respuesta a la solicitud, con datos de posición.
- startLocationUpdates: Llamado para comenzar a capturar datos cada segundo.
- StopLocationUpdates: Llamado para parar la captura de datos.
- sendHTTPRequest: Método que envía al servidor la información de velocidad.

Arduino

El objetivo de la placa Arduino es obtener el ángulo del volante y enviarlo al servidor.

- ObtainAngle: Cada segundo, se obtendrán datos del sensor giroscopio incluido en el volante.
- makeHTTPRequest: Envía al servidor el dato del ángulo.

Server

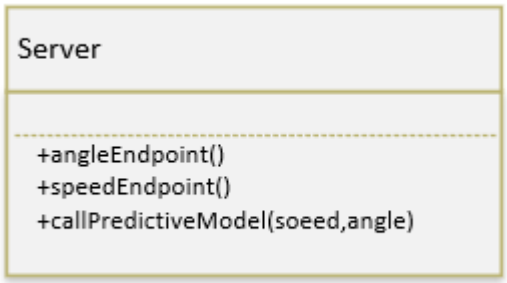


Ilustración 20 Diagrama de clases: Servidor

El propósito del servidor es recibir las peticiones de los clientes y detectar cuándo hay que poner un intermitente llamando al modelo predictivo.

- angleEndpoint: Servicio que gestiona las llamadas hechas desde Arduino.
- speedEndpoint: Servicio que gestiona las llamadas hechas desde Android.
- callPredictiveModel: cuando se han recibido dos peticiones para una misma hora, se llama al modelo predictivo.

Escuela de
Ingeniería Informática
Universidad de Oviedo

Capítulo 3 Diseño del Sistema de Información

Fase de Desarrollo

DSI

DSI 1: Diseño de Casos de Uso Reales

Caso de Uso: Captura de Datos

Para capturar datos se asume que los clientes de Arduino y Android están instalados.

Arduino

Se proporcionan dos "proyectos" para Arduino, uno servirá para almacenar datos en tarjetas SD de manera local, y el otro para enviar peticiones vía internet a nuestro servidor. En ambos casos es necesario obtener la hora, esto puede hacerse vía internet o bien con un reloj instalado y configurado. Se proponen diferentes configuraciones de pines y sensores para que se elija la que mejor se adapte a la situación.

Arduino UNO con sensor de Reloj:

Tabla 15 Pines de instalación en Arduino Uno

	PIN	**Conector**	
Arduino UNO	Digital 4	CS	SD
	Digital 13	SCK	
	Digital 11	MOSI	
	Digital 12	MISO	
	5V	VCC	
	GND	GND	
	DIGITAL 9	SCL	RELOJ
	DIGITAL 8	SCA	
	3.3V	VCC	
	GND	GND	
	3.3V	VCC	Giroscopio
	GND	GND	
	SCL	SCL	
	SCA	SCA	
	Digital 2	int	

Arduino UNO WiFi REV2

Tabla 16 Pines de instalación en una Arduino UNO WiFI REV2

	PIN	**Conector**	
	3.3V	VCC	Giroscopio
	GND	GND	

	SCL	SCL	
	SCA	SCA	
	Digital 2	int	

No se ha incluido la configuración de una Arduino con los pines de un sensor WiFi externo.

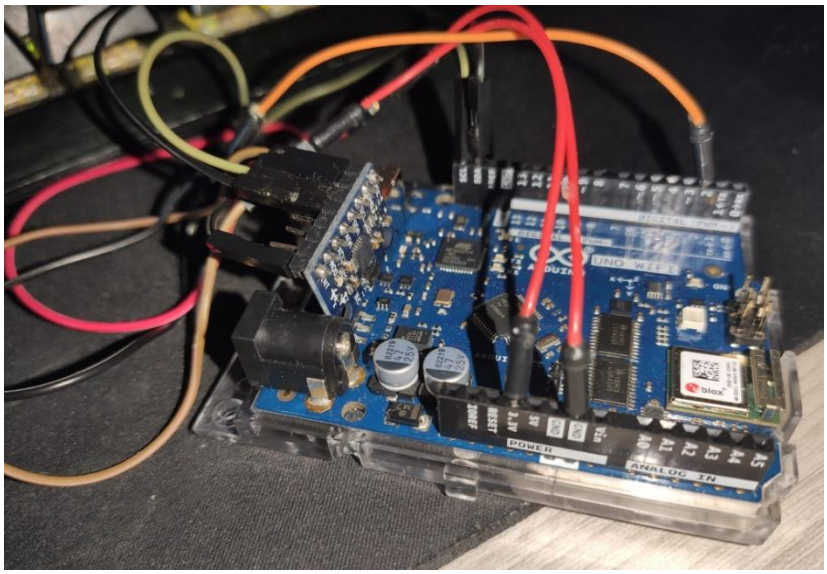


Ilustración 21 Montaje Arduino WIFI para envío de datos al servidor

Para hacer una adaptación de código para utilizar otro tipo de sensor, bien Wifi o GPS será necesario seguir la documentación del fabricante o de una librería compatible. Habrá que indicar en método setup en qué pines se han conectado los sensores. Como ejemplo, en caso de querer utilizar un sensor Wifi independiente, habrá que seguir la configuración para obtener un elemento del tipo definido por Arduino “WifiClient”.

Caso de Uso: Obtención de datos

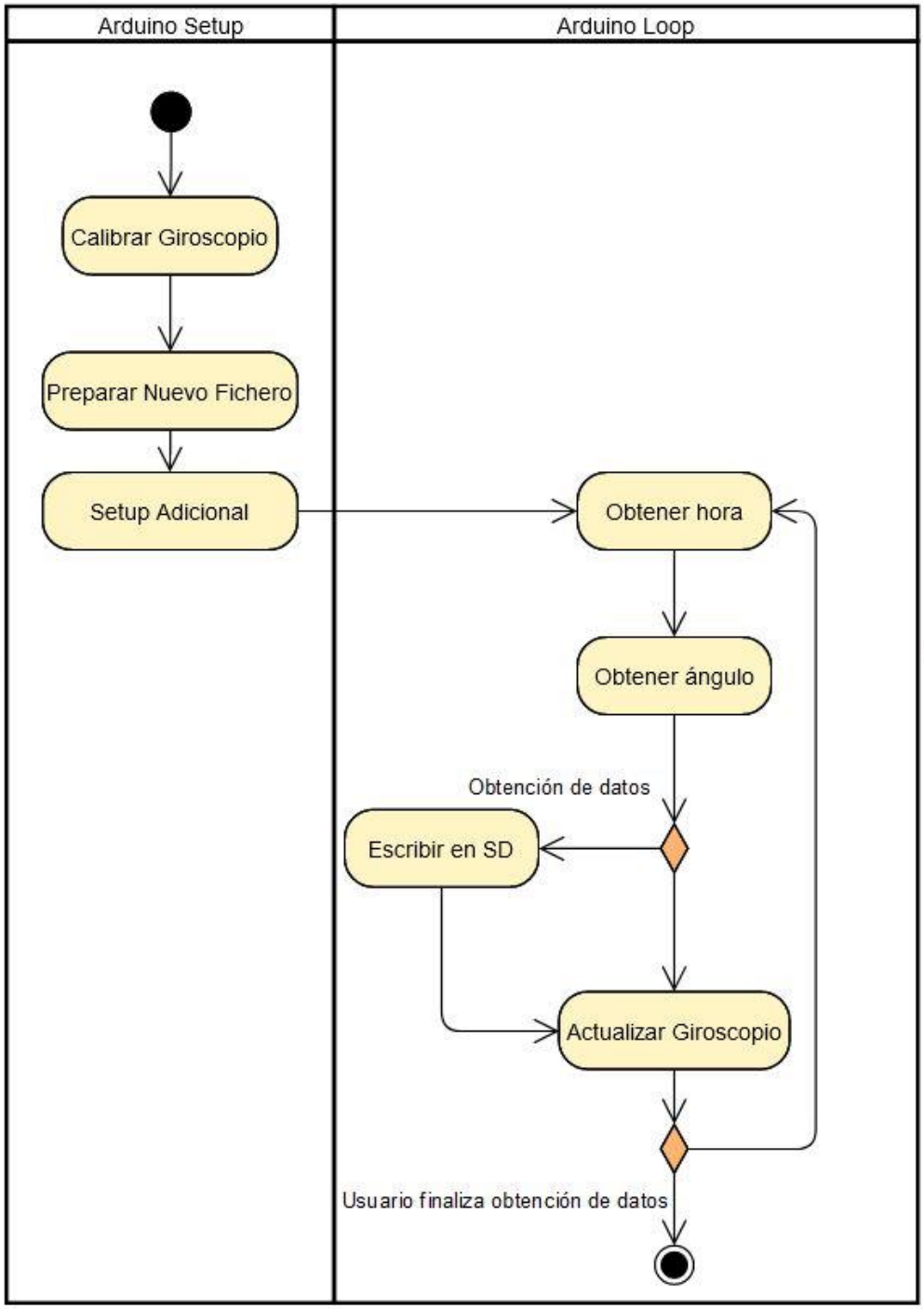


Ilustración 22 Flujo de obtención de datos

Se incluye "Setup Adicional" en el caso de que se quieran utilizar diversas tarjetas. Una Arduino UNO necesitará un reloj adicional que configurar, sin embargo, una Arduino con sensor WIFI necesitará configuración de red para poder obtener la hora actual.

El flujo de envío de datos es similar, sustituyendo "Escribir en SD" por "Realizar petición REST".

Android

Para la captura de datos y generación de GPX se utilizará una aplicación ya existente, open source y con licencia libre "GPS Logger" que generará un GPX del trayecto recorrido.

Tras haber activado en ambos clientes la captura de datos, los datos del trayecto quedarán registrados en ficheros físicos.

Caso de Uso: Entrenamiento de modelo predictivo

Se asume que se han obtenido los ficheros de datos mediante captura de datos.

A partir de ambos ficheros hay que generar un fichero de entrenamiento en el que se indique, bien de manera automática o manual, en qué momentos del trayecto se tenían que haber activado los intermitentes. Este fichero deberá indicar la hora y sentido de intermitentes.

En una primera instancia, cuando no existe ningún modelo predictivo, este entrenamiento tendrá que ser manual. Para evitar la doble carga de tener que indicar cuándo hay que poner intermitente y además preocuparse del formato del fichero, se proporcionará un proceso en el que dados los ficheros de Arduino y de GPX genere un CSV en el que se tendrá que rellenar marcando las horas.

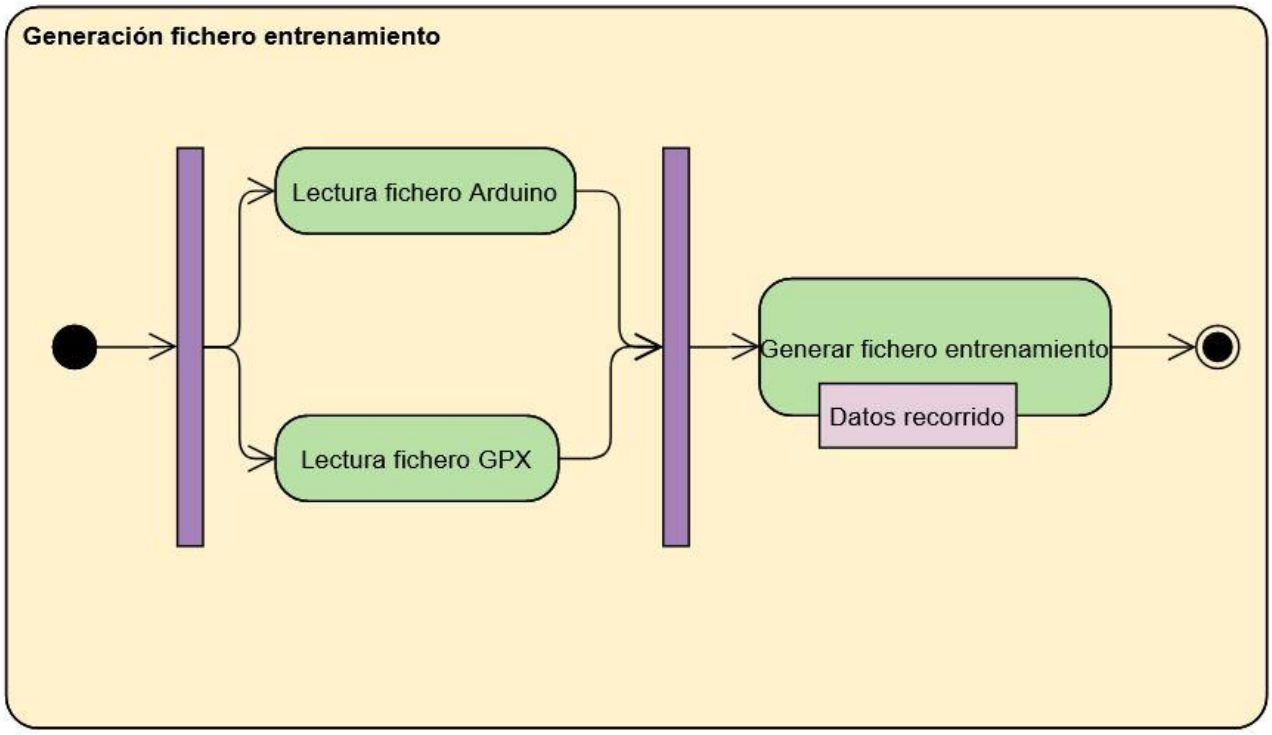


Ilustración 23 Generación de ficheros de entrenamiento

A la hora de generar un fichero de entrenamiento y tras haber leído ambos ficheros, generará una línea de fichero de entrenamiento por cada timestamp que exista en los dos ficheros.

Una vez obtenido y rellenado el fichero de entrenamiento, se llamará a un proceso muy similar incluyendo como nuevo parámetro el fichero rellenado. Este proceso unificará de nuevo por marca de tiempo todos los ficheros, obtendrá la implementación del modelo predictivo a utilizar y lo entrenará pasándole el conjunto de datos de velocidad, ángulo e intermitente.

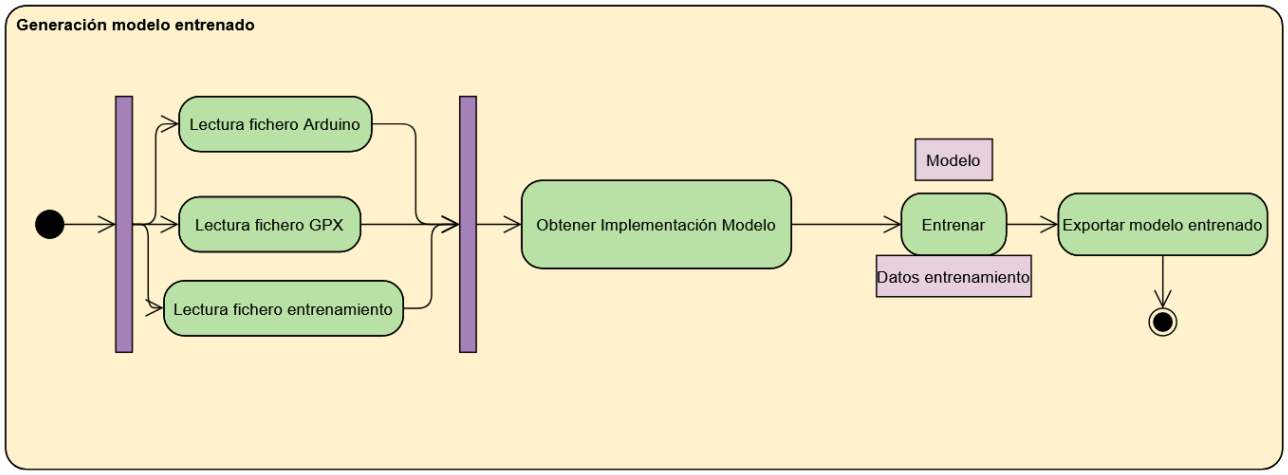


Ilustración 24 Generación de un modelo entrenado

Al final del proceso, se tendrá un fichero que represente al modelo predictivo que pueda ser cargado de nuevo sin pérdida de información.

Caso de Uso: Conducir

Se asume que el usuario final tiene los dos clientes configurados. Ambos enviarán información al servidor en paralelo.

La manera de procesar las peticiones será muy similar en los dos servicios, pues ambos son parámetros necesarios para poder realizar la llamada al modelo predictivo.

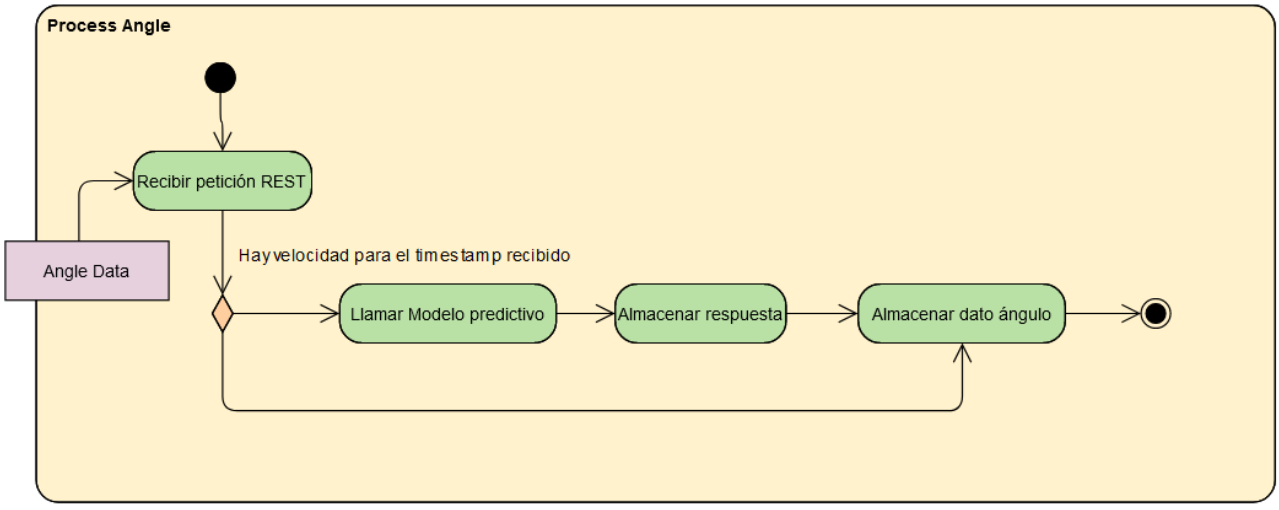


Ilustración 25 Endpoint REST de ángulo

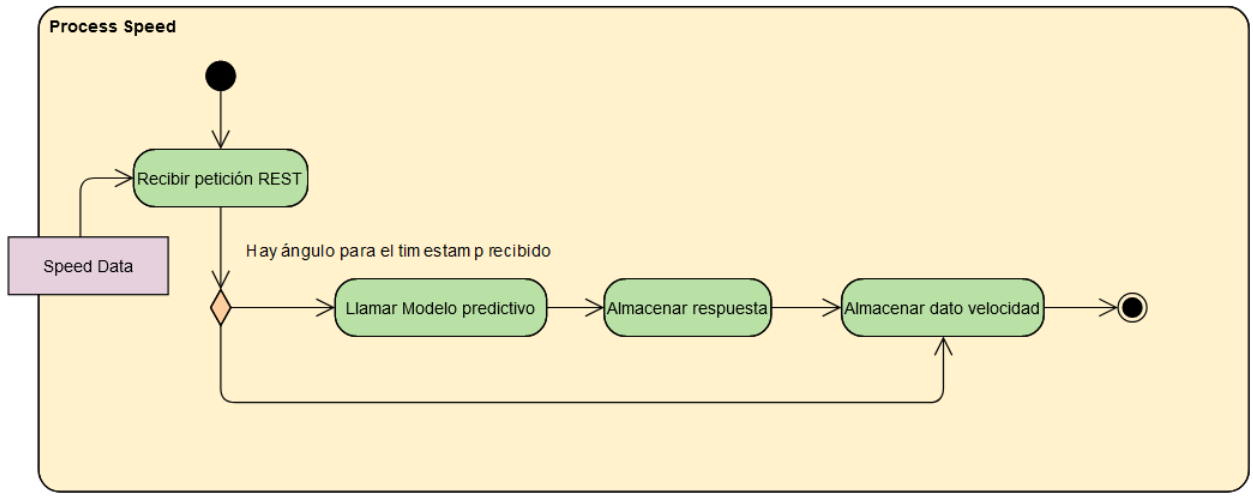


Ilustración 26 Endpoint REST de velocidad

Sin embargo, en estos puntos hay dos elementos importantes con ciertas peculiaridades en su funcionamiento:

- **Almacenaje de respuesta y comprobación de la existencia de datos:** Este servidor es una prueba de concepto, por lo que no se tienen muy en cuenta conceptos como condiciones de carrera ni semáforos. Esta solución no es escalable de cara a miles de clientes enviando peticiones, pues tampoco es capaz de identificar los clientes. Se asume que habrá algunos casos en los que nunca se realicen llamadas al modelo predictivo.
- **Respuesta:** Como se ve en las figuras, nunca se dará respuesta al cliente para informar de si se debería activar o no un intermitente. Aun así, este dato será almacenado en el servidor para su posterior estudio.

DSI 2: DISEÑO DE CLASES

En esta sección mostraremos la diferencia entre el diseño de clases propuesto inicialmente y las modificaciones que han surgido.

Modelo Predictivo

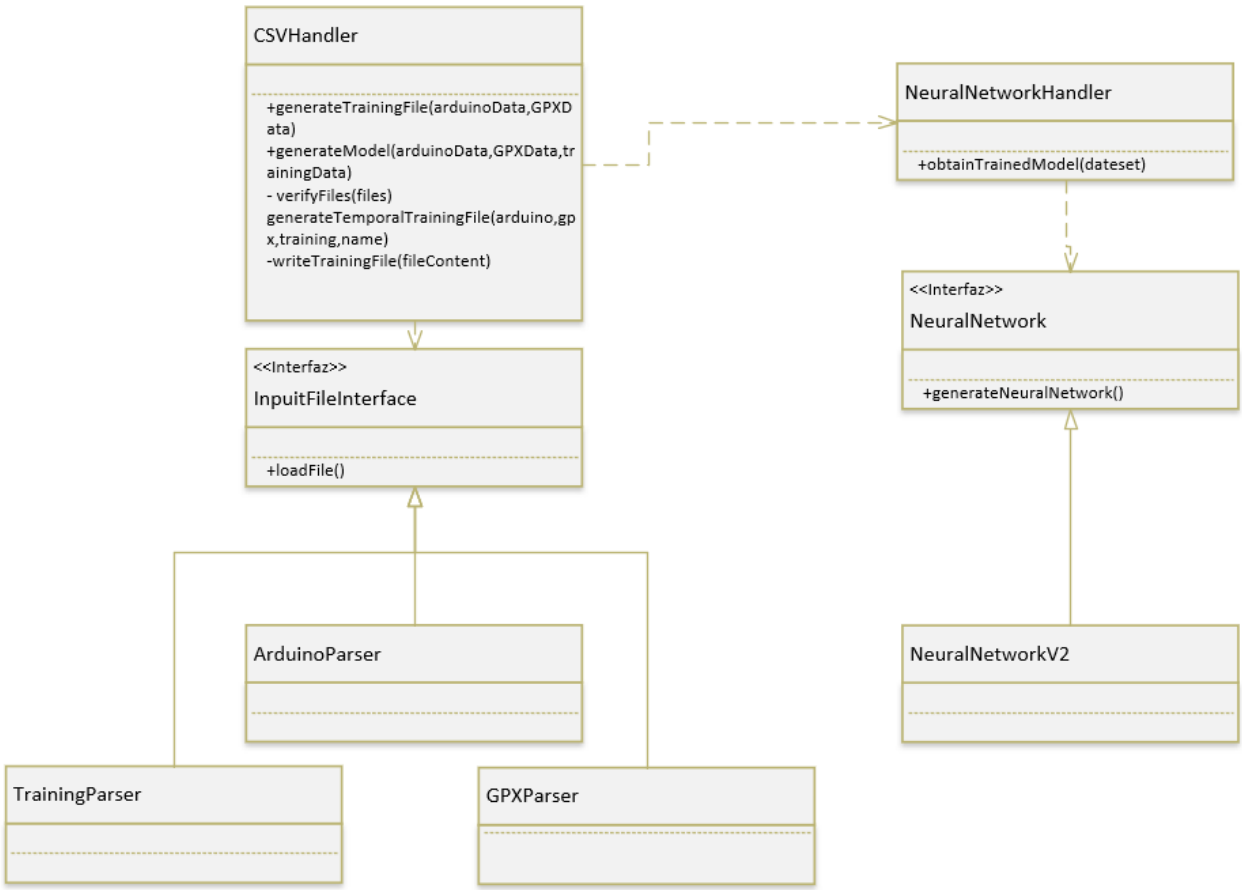


Ilustración 27 Diagrama de Clases Modelo Predictivo

CSVHandler:

Se han incluido diversos métodos privados para realizar validaciones sobre las entradas, así como métodos para generar los contenidos de ficheros y escribirlos.

Para lectura de algunos ficheros, como el de entrenamiento se utilizará la librería Python Pandas. Esta librería se creó a partir de NumPy para cargar, manipular y analizar datos, por lo que el manejo de contenido de ficheros en memoria, una vez leídos del disco es bastante rápida.

TrainingParser:

Cargará el CSV de entrenamiento. Este estará formado por dos elementos. “hour” y “blinker”.

ArduinoParser:

Los ficheros generados por Arduino son bastante similares a un CSV, como se explica en DSI 4: Diseño Físico de Datos, por ello se ha implementado un lector ad-hoc para cargarlo.

GPXParser:

El fichero GPX será generado por la aplicación Android modificada GPS Logger. Este fichero GPX cumple la especificación, por lo que para cargar el fichero GPX y extraer velocidades utilizaremos la librería Python GPXPy.

NeuralNetworkHandler:

Esta clase configurará los parámetros de Keras y TensorFlow necesarios para utilizar la tarjeta gráfica del PC antes de generar y entrenar ningún modelo.

También contendrá la configuración y parametrización de entrenamiento como número de veces a entrenar o cómo dividir el conjunto de datos de entrenamiento.

NeuralNetworkV2:

Contendrá la implementación de una red neuronal de tipo CNN con un dropout del 10%.

Clientes

Arduino

Ilustración 28 Diagrama de clases: Arduino

El proyecto de Arduino utilizara las siguientes librerías:

- WiFiNINA: Librería utilizada para configurar el WiFi y enviar peticiones HTTP a nuestro servidor.
- MPU6050_tockn: Librería utilizada para calibrar y obtener los ángulos del giroscopio sin tener que realizar ninguna operación matemática compleja.
- SD: Librería que actúa de intermediaria para escribir ficheros en tarjetas SD.
- DS3231: Librería que configura el reloj, en caso de utilizar una placa sin WiFi.

Toda la configuración de sensores y librerías se realizará en el setup, como la configuración de red para tener internet o preparar un fichero de escritura para almacenar datos.

Android

Para implementar la funcionalidad de Android se ha hecho un clonado de la aplicación abierta y con libertad para uso comercial llamada GPS Logger [36].

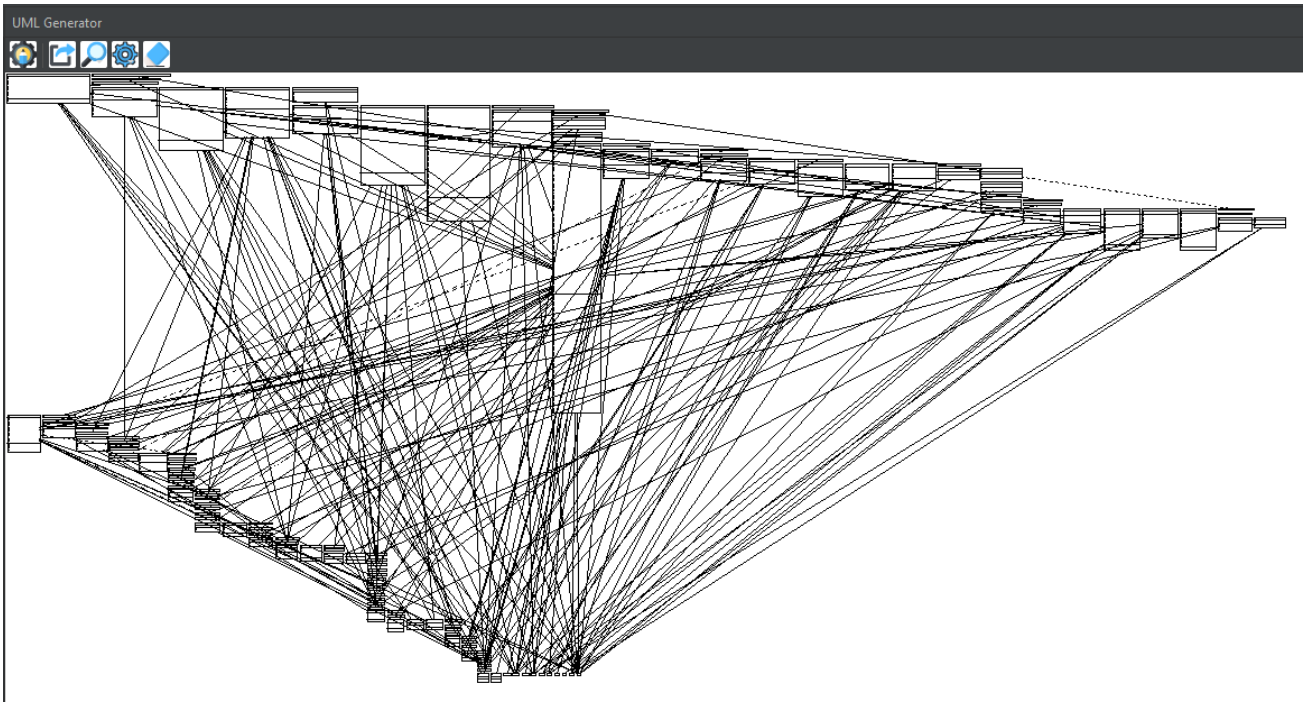


Ilustración 29 Diagrama UML Autogenerado

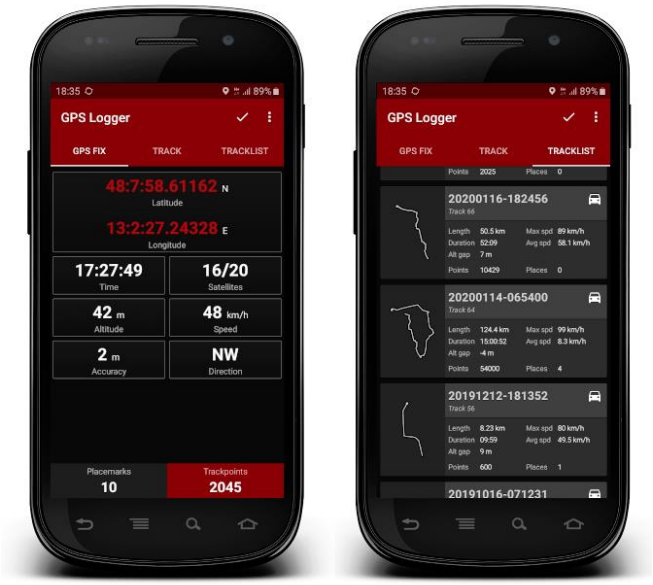


Ilustración 30 GPS Logger en funcionamiento

Está aplicación tiene un diagrama de clases muy complejo pues contiene casi todas las funcionalidades que necesitamos como capturar en tiempo real nuestra ruta vía satélites GPS, almacenarlos en rutas, configurar marcadores en el itinerario, obtener estadísticas de viajes, exportar datos a diferentes formatos y con diferentes estándares. La única funcionalidad que le falta es el enviar en tiempo real la velocidad al servidor.

Por ello, y para simplificar el diagrama, solamente explicaremos las modificaciones que hay que realizar al proyecto ya existente.

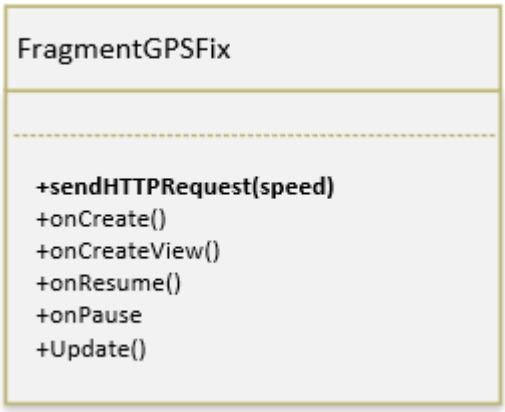


Ilustración 31 Diagrama de Clases modificadas GPSLogger

La clase "FragmentGPSFix" es la encargada de obtener la posición actual GPS y calcular altura, velocidad y notificar a las clases necesarias para mostrar toda esta información por pantalla.

Hemos incluido una función que envía una petición HTTP con la velocidad calculada por la aplicación y el momento en el que se envía la petición. Este envío será realizado con la librería Volley [37] .

Esta librería se encargará de gestionar y programar las conexiones de manera asíncrona, soportando múltiples conexiones simultaneas y teniendo una cola de peticiones pendientes. Esto nos permite realizar la petición al estilo "dispara y olvida" [38] sin tener que implementar ninguna gestión.

Server

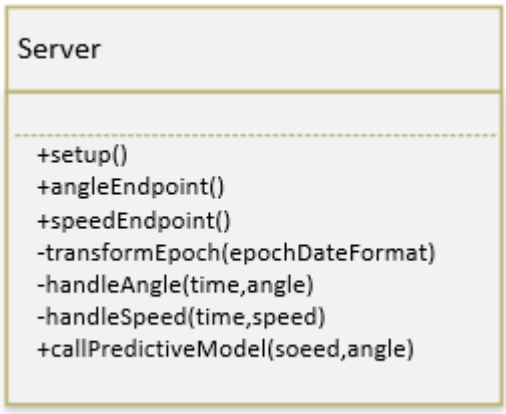


Ilustración 32 Diagrama de clases Servidor

En la inicialización se inicializará la configuración de TensorFlow y Keras para poder llamar al modelo predictivo. El modelo predictivo será cargado en memoria mediante un fichero indicado al arrancar el servidor mediante parámetro.

Ambos endpoints transforman el epoch Linux en un formato más humano, y transforman los Strings recibidos al formato que el modelo predictivo espera, entero y double para el ángulo y velocidad respectivamente.

La sincronización de ambos clientes se realizará almacenando en memoria las peticiones en un mapa, cuyas claves son el datetime de petición.

DSI 3: DISEÑO DE LA ARQUITECTURA DE MÓDULOS DEL SISTEMA

DSI 3.1: Diagramas de Paquetes

En esta sección se explicará la división lógica que se ha seguido para estructurar y paquetizar el proyecto.

Paquete 1: Modelo predictivo

Contiene todas las clases responsables de generar y entrenar modelos predictivos. Una clase se encargará de orquestar las llamadas a los siguientes subpaquetes para utilizar los datos provenientes de ficheros en entrenamientos de modelos predictivos.

Subpaquete 1: File Handlers

Contiene al interfaz y las implementaciones para cargar ficheros en memoria, procesarlos y devolverlos para su posterior utilización.

Subpaquete 2: Predictors

Contiene la interfaz y las implementaciones para crear modelos predictivos.

Paquete 2: Android

La arquitectura definida para GPS Logger no sigue ningún tipo de estructura de paquetes interna. No se ha alterado su estructura para no rehacer toda la arquitectura y tener que arreglar todos los problemas derivados de ello.

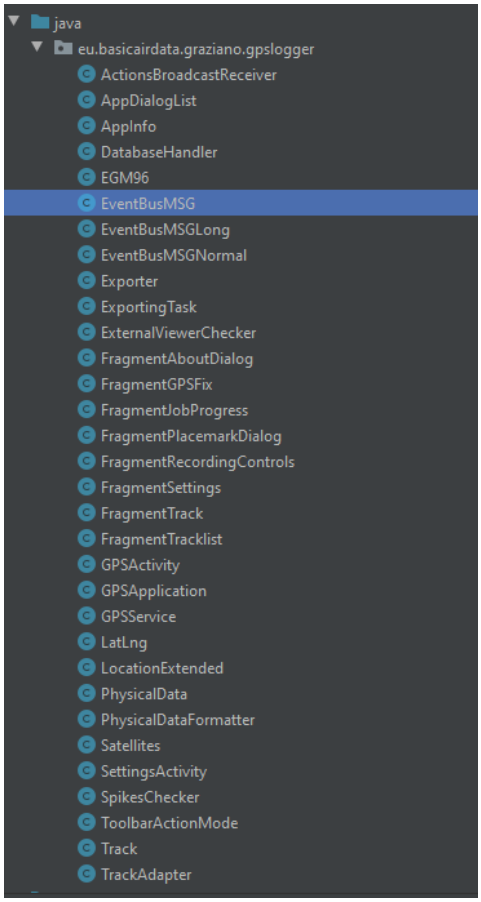


Ilustración 33 Clases de GPSLogger sin paquetizar

Paquete 3: Arduino

La única separación lógica que haremos en Arduino será extraer la configuración de red a un fichero externo que cargar. El resto estará en un solo fichero.

Paquete 4: Servidor

Contendrá las clases necesarias para poder cargar un modelo predictivo y responder a peticiones rest.

DSI 3.2: Diagrama de Despliegue

En esta sección explicaremos todos los pasos necesarios para desplegar el sistema en un entorno productivo.

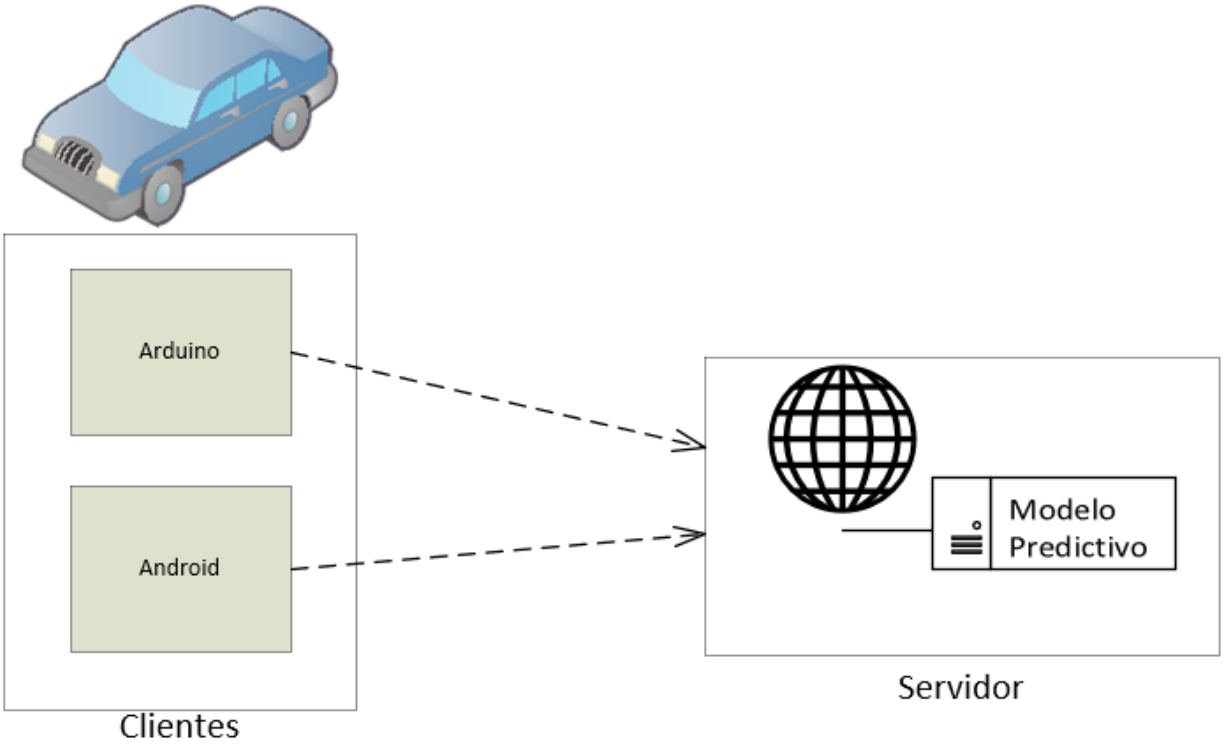


Ilustración 34 Diagrama de despliegue

Vehículo:

Será necesario instalar los clientes Android y Arduino en el vehículo.

Android:

Antes de instalar ninguna aplicación, hay que actualizar la IP a la que se envían los datos e incluir dicha IP en network_security_config.xml para que Android permita la salida de datos al servidor indicado vía internet.

Una vez hayamos hecho dicho cambio, se podrá instalar la aplicación GPSLogger modificada que hemos proporcionado en un dispositivo Android que pueda acceder de alguna manera a la red de GPS. Bien un móvil con tarjeta SIM y conexión o un dispositivo Android emparejado con un receptor GPS.

Una vez instalado, habrá que arrancar la aplicación y aceptar los permisos de internet y posición para un correcto funcionamiento.

Para este proyecto se optará por utilizar un dispositivo móvil.

Arduino:

Antes de instalar ninguna aplicación, hay que actualizar la IP a la que se envían los datos en la aplicación.

Una vez tengamos este cambio listo habrá que encontrar un sitio en la posición del conductor del vehículo que no moleste durante la conducción y colocar la Arduino con una batería de manera que el giroscopio recopile el giro del volante durante la conducción y cargar el programa proporcionado.

Para nuestro proyecto se utilizará una Arduino WiFi Rev2 con un giroscopio MPU6050.

Ilustración 35 Posición estimada de Placa Arduino, figura grande y azul, junto al sensor en Amarillo

En nuestro caso se colocará la placa tras el volante para no afectar a su manejo.

Ilustración 36 Posición a colocar el giroscopio

Es importante saber que en función de cómo coloquemos el giroscopio tendremos que recoger los datos de un eje concreto.

Colocando el volante en posición de 0º grados, y colocando el sensor en la posición de la siguiente figura, el ángulo a recoger sería el Z.

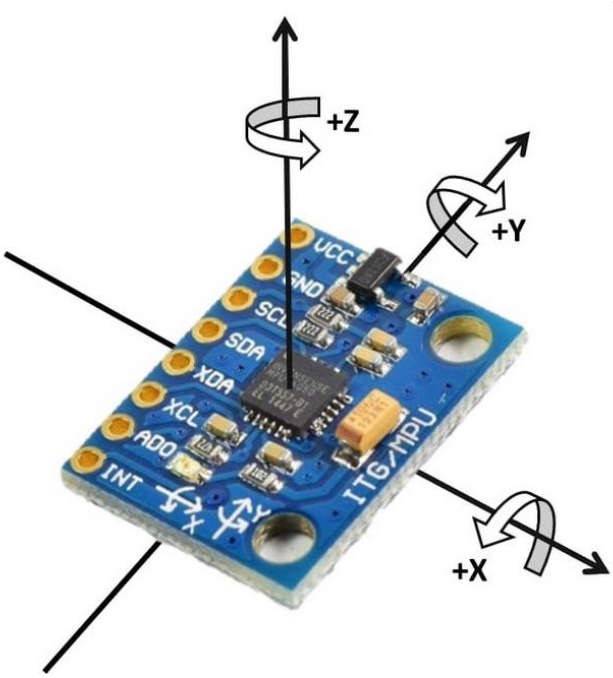


Ilustración 37 Referencia del sensor MPU6050

Servidor y red:

La arquitectura de red de nuestro proyecto será muy simple. Crearemos una red WiFi privada en el vehículo, a la cual conectaremos los anteriores clientes, y tendremos un ordenador portátil con el servidor arrancado, para tener visibilidad de manera sencilla.

Otra alternativa sería desplegar la aplicación Flask en cualquier proveedor como Google Cloud.

Lo importante de este despliegue es asegurarse de que la URI, puerto y endpoints estén disponibles para realizar peticiones.

DSI 4: Diseño Físico de Datos

En el proyecto no tenemos una base de datos en la que almacenar información y todas las estructuras de datos en ficheros utilizadas son CSV salvo los ficheros generados por Arduino.

Su formato, por cada línea es el siguiente. En negrita incluimos las variables.

epoch|angleX:**anguloX**\tangleY:**anguloY**\t angleZ:**anguloZ**\n

Un ejemplo de entrada sería:
1620561289|angleX:59.16 angleY:6.38 angleZ:0.25

En el ejemplo anterior se incluyen los tres ángulos obtenidos por el giroscopio, pero realmente solo uno de ellos es el que es útil, al representar el ángulo de giro del volante.

DSI 5: Especificación Técnica del Plan de Pruebas

Se han definido tres tipos de pruebas a ejecutar. Una de ellas servirá para asegurar que la información llega realmente al sistema, definidas como pruebas de integración. La otra servirá para analizar el porcentaje de precisión de diversos modelos predictivos, denominadas pruebas de rendimiento. También se cuentan con pruebas unitarias que comprueba el correcto funcionamiento a nivel atómico de los métodos y clases implementadas.

DSI 5.1: Pruebas unitarias

Se han realizado pruebas unitarias para Arduino, para los generadores de modelos predictivos y en el servidor.

Arduino

Para realizar pruebas unitarias en Arduino se ha decidido utilizar AUnit [39], un framework de testing para Arduino.

La idea de utilizar este framework es verificar y validar que el código cargado en una Arduino funciona correctamente. Se implementarán pruebas que validen el funcionamiento del giroscopio, así como la conectividad de red.

Modelo predictivo

Se probarán todos los elementos de lectura de ficheros, la generación de modelos predictivos, así como un flujo completo de lectura y exportación de ficheros, simulando a un usuario final utilizando nuestro programa.

Servidor

Se probarán los endpoints y lógica interna, así como una llamada al modelo predictivo cargado.

DSI 5.2: Pruebas de Integración

Prueba Integración Android/Arduino: se encenderá la aplicación y se confirmará que están llegando peticiones al servidor. En caso de fallo habrá que hacer un estudio de la red para ver qué elemento está fallando.

DSI 5.3: Pruebas de Carga de Servidor

El proyecto se ha planteado de manera que actualmente no somos capaces de diferenciar quién realiza las peticiones, pues solo tendríamos un vehículo con dos clientes realizando peticiones. A modo de estudio de escalabilidad se plantea esta sección de pruebas de carga. En ellas simularemos la existencia de múltiples vehículos realizando peticiones.

Para ello desplegaremos el servidor Flask en un entorno productivo, ya que hasta ahora estaba en modo desarrollo, el cual no está planteado para eficiencia. Se utilizará Waitress [40], así como, su tutorial para desplegar a producción nuestro servidor. Se utilizará la configuración por defecto de Waitress.

La máquina en la que colocaremos el servidor tiene 32GB de memoria RAM, un procesador Intel i7 8700K, y el servidor se encuentra instalado en un disco duro SSD NVMe de 500GB.

Para realizar las pruebas de carga utilizaremos Locust [41], un framework en Python que nos permite definir cómo actúan diversos clientes, utilizando Python y diversos decoradores que ofrecen.

Las pruebas que realizaremos consistirán en definir un cliente que en un intervalo de uno a dos segundos realice dos peticiones REST, una cada uno de los endpoints. Además, para simular problemas técnicos, una de cada cinco peticiones solo llamará a uno de los dos endpoints. Comenzaremos con un solo vehículo simulado, e incrementaremos de cinco en cinco el número de vehículos, hasta un total de 500. Viendo cómo reacciona nuestro servidor al creciente número de peticiones por segundo (RPS, *Requests per second*).

DSI 5.4: Pruebas de Rendimiento

Dado un conjunto de datos de entrenamiento, se ejecutarán diferentes pruebas para analizar la precisión de diferentes redes neuronales con diferentes parametrizaciones, así como diferentes métodos estadísticos como Análisis Discriminante Lineal, KNN, redes bayesianas, arboles de clasificación y regresión (CART) o máquinas de vector soporte (SVM). El objetivo de estas pruebas es analizar qué métodos predictivos pueden ser más útiles, así como tener una comparativa sobre diferentes configuraciones de redes neuronales.

Capítulo 4 Construcción del Sistema de Información

Fase de Desarrollo

CSI

CSI 1: PREPARACIÓN DEL ENTORNO DE GENERACIÓN Y CONSTRUCCIÓN

En esta sección incluiremos un análisis de los aspectos éticos y profesionales de nuestro proyecto, así como una descripción de tecnologías y herramientas utilizadas, junto a unos manuales para que en caso de querer continuar este proyecto se tenga una línea base de la que partir.

Finalmente se mostrarán los resultados de las pruebas de integración y rendimiento del proyecto.

CSI 1.1: Aspectos sociales, legales, éticos y profesionales

Como ya explicamos previamente, y al tratarse de una versión inicial, estamos cubiertos legalmente por la RGPD pues no almacenamos ningún tipo de información personal. Evidentemente se podría detectar de qué vehículo nos llega la información, en el caso de que este proyecto fuese a expandirse a múltiples vehículos o conductores habría que tomar medidas o bien para almacenar la información de manera segura, o bien para anonimizar el origen de los datos.

A nivel comercial y enlazando con el punto anterior, los datos de conducción son un bien muy valorado por empresas aseguradoras. En muchos casos, ofrecen precios de seguros más económicos incluyendo un sensor GPS para localizar ubicación, velocidad y hábitos de conducción, haciendo así más difícil un fraude respecto a las preguntas de cuánto se utiliza el vehículo, o en caso de accidente, detectar si se estaba superando el límite de velocidad. No es de extrañar que quizá el hecho de saber si se debería haber utilizado un intermitente enfrentado a saber si el usuario lo utilizó realmente puede ser de bastante valor para estas empresas en caso de un siniestro. Evidentemente este tipo de datos y tecnología también es de gran valor para empresas que están invirtiendo en la obtención de vehículos autónomos.

Cualquier modificación en cómo funcionan los vehículos tienen un impacto en la sociedad. La popularización de vehículos híbridos y eléctricos trajo consigo el problema de que en pocas velocidades eran totalmente silenciosos, causando accidentes en personas con problemas visuales o qué no prestaban atención a su entorno, para solucionarlo se decidió incluir un Sistema de Aviso Acústico de Vehículos (AVAS). Un problema similar tiene vehículos autónomos, que están sufriendo ciertos siniestros, en algunos casos con víctimas mortales, cada uno con orígenes y casuísticas propias.

La tecnología para detectar cuándo hay que activar un intermitente, y el hecho de activarlo automáticamente se puede utilizar como asistencia a la conducción humana, así como en vehículos totalmente autónomos, en la gran mayoría de casos ayudará al humano cuando se le olvide que tenía que haberlo que activado. Sin embargo, no hay que olvidar, que al igual que puede ayudar, en falsos positivos puede ser el causante de incidentes, al reaccionar otros viandantes o vehículos a nuestra señalización incorrecta. Por ello, si este sistema que activa intermitentes se llega a implementar realmente, es prioritario que sea algo que el conductor sea capaz de sobrescribir en caso de falso positivo. Este sistema no debería eliminar los controles de activación, luces de intermitentes en el

control de mando del vehículo ni su característico sonido de activación. De esta manera funcionaría de manera similar a un control de crucero o sistema que te mantiene en el carril, algo que funciona pero que en cualquier momento un conductor puede anular si considera necesario. Con el paso del tiempo, y cuando los vehículos autónomos pasen a sustituir totalmente a conductores humanos, esta tecnología tendrá el mismo impacto, sin la posibilidad de anulación por parte de un conductor. En este caso se habrán tenido que recopilar una cantidad de datos para tener una alta precisión, o bien un uso de futuras tecnologías que incrementen la precisión de uso.

CSI 1.2: Lenguajes de programación

Los lenguajes de programación utilizados para implementar el proyecto han sido:

- Lenguaje de programación Arduino (variante de C++) para programar la captura y envío de datos en la placa Arduino. Se han utilizado diversas librerías de sensores que ofrecen interfaces que simplifican el desarrollo. Estas librerías son:
    - DS3231: Librería que gestiona el sensor de reloj.
    - I2CDev: Librería que unifica interfaces para trabajar de manera sencilla con dispositivos I2C.
    - MPU6050_tockn: Librería que permite calibrar giroscopios y obtener sus valores directamente en ángulos.
    - WiFiNINA: Librería adaptada para la placa Arduino UNO WiFi Rev.2, permite gestionar conexiones Wifi y realizar peticiones.

Python para servidor, así como generación de modelos predictivos, análisis de ficheros y entrenamiento de modelos predictivos. Para el servidor hemos utilizado Flask como punto de partida para generar un servidor en pocas líneas de código.

CSI 1.3: Herramientas y programas usados para el desarrollo

Mencionaremos las herramientas utilizadas que han facilitado el desarrollo de este proyecto.

Arduino IDE [42]

Entorno de desarrollo para las placas Arduino y todas las placas compatibles. Permite compilar y cargar programas a placas conectadas por USB. Tiene un sistema de gestión de librerías, con un buscador integrado para encontrar librerías. Además, incluye ejemplos de código funcionales.

Ilustración 38 Placa Arduino Uno WiFi Rev2 Conectada a un puerto USB

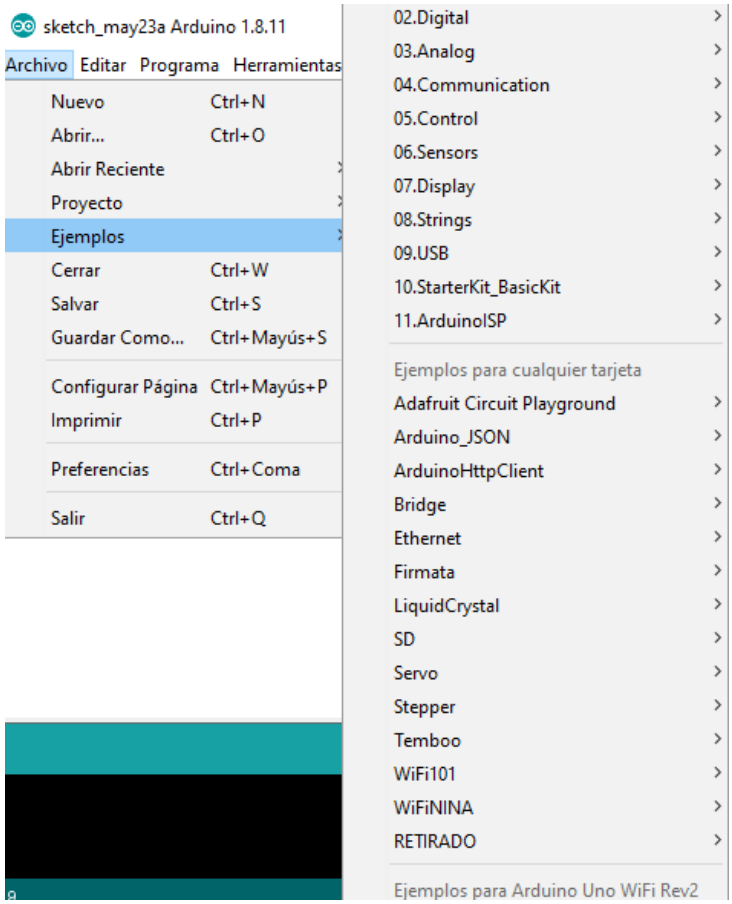


Ilustración 39 Ejemplos proporcionados por el IDE, por las placas detectadas y por las librerías instaladas

PyCharm [43]

Entorno de desarrollo para Python. Permite crear entornos Python independientes, gestionar paquetes sin haber instalado la herramienta pip, realizar un análisis de código, remarcándote posibles errores de ejecución o warnings.

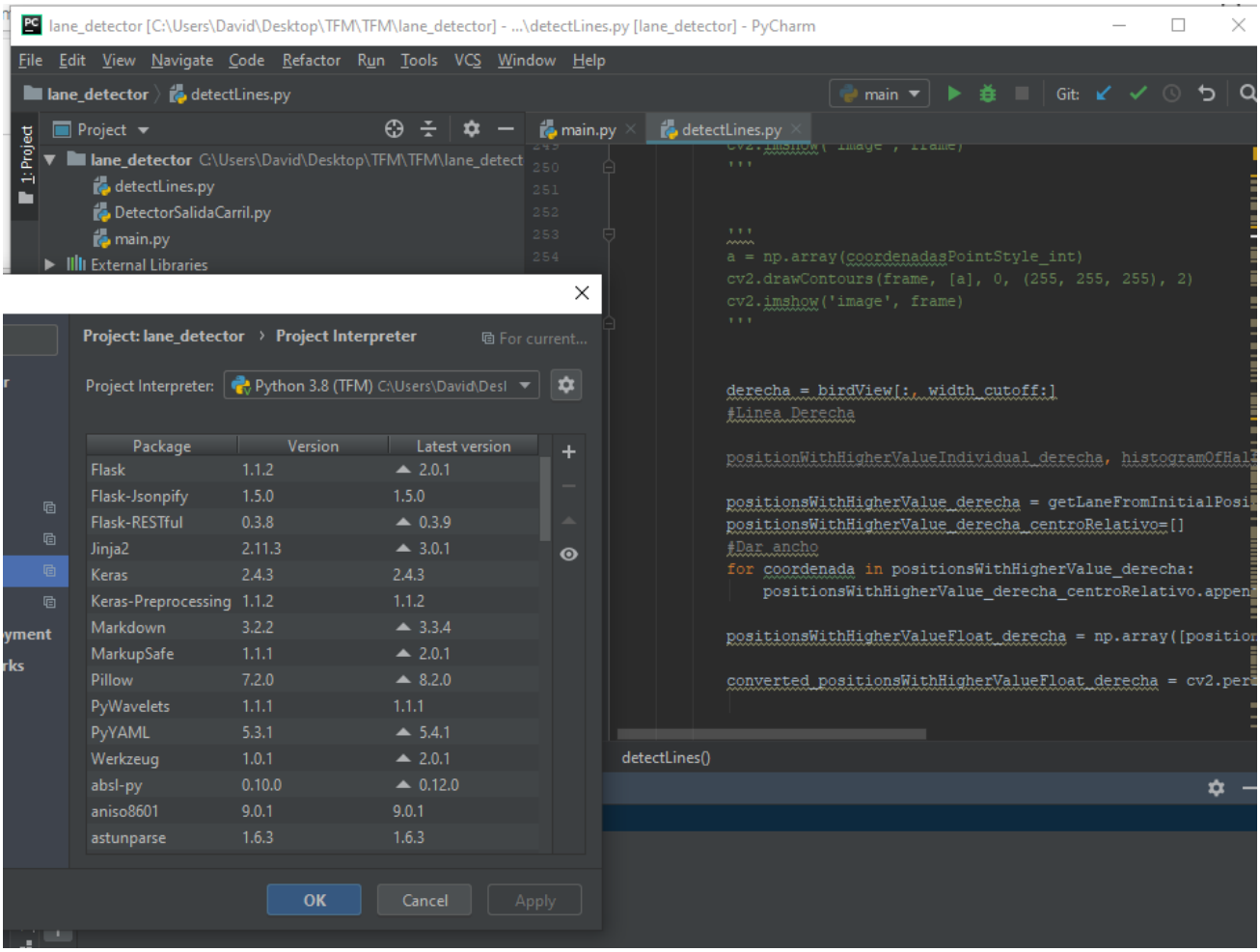


Ilustración 40 Pycharm con la gestión de paquetes abierta

IDE para Python con gestión de librerías y entornos por proyecto, permitiendo crear configuraciones de ejecución. Su función de auto importado con Control+Espacio también ha sido de utilidad cuando hay múltiples métodos en diferentes librerías.

Notepad++ [44]

Editor de ficheros con soporte para múltiples lenguajes. No se ha programado nada con él en el proyecto, pero ha sido una herramienta muy útil a la hora de abrir ficheros CSV, GPX, o txt sin que se abran con editores por defecto como Excel, que tardan en cargar.

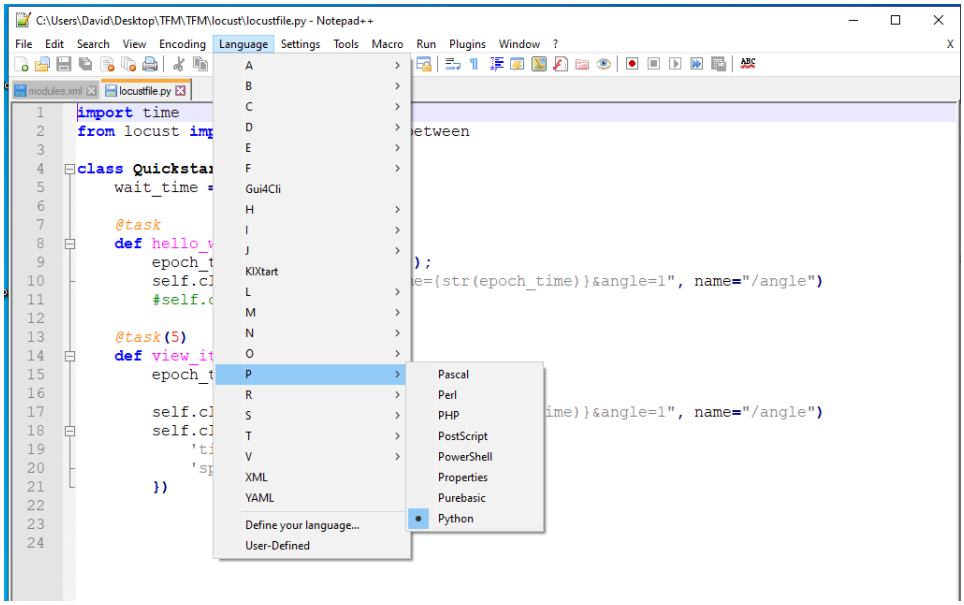


Ilustración 41 Notepad++ abriendo un fichero Python, auto detectando el lenguaje

Insomnia [45]

Herramienta para definir, documentar y realizar peticiones REST a servicios. Permite ver las respuestas del servicio, crear pipelines, test y gestionar entornos con variables.

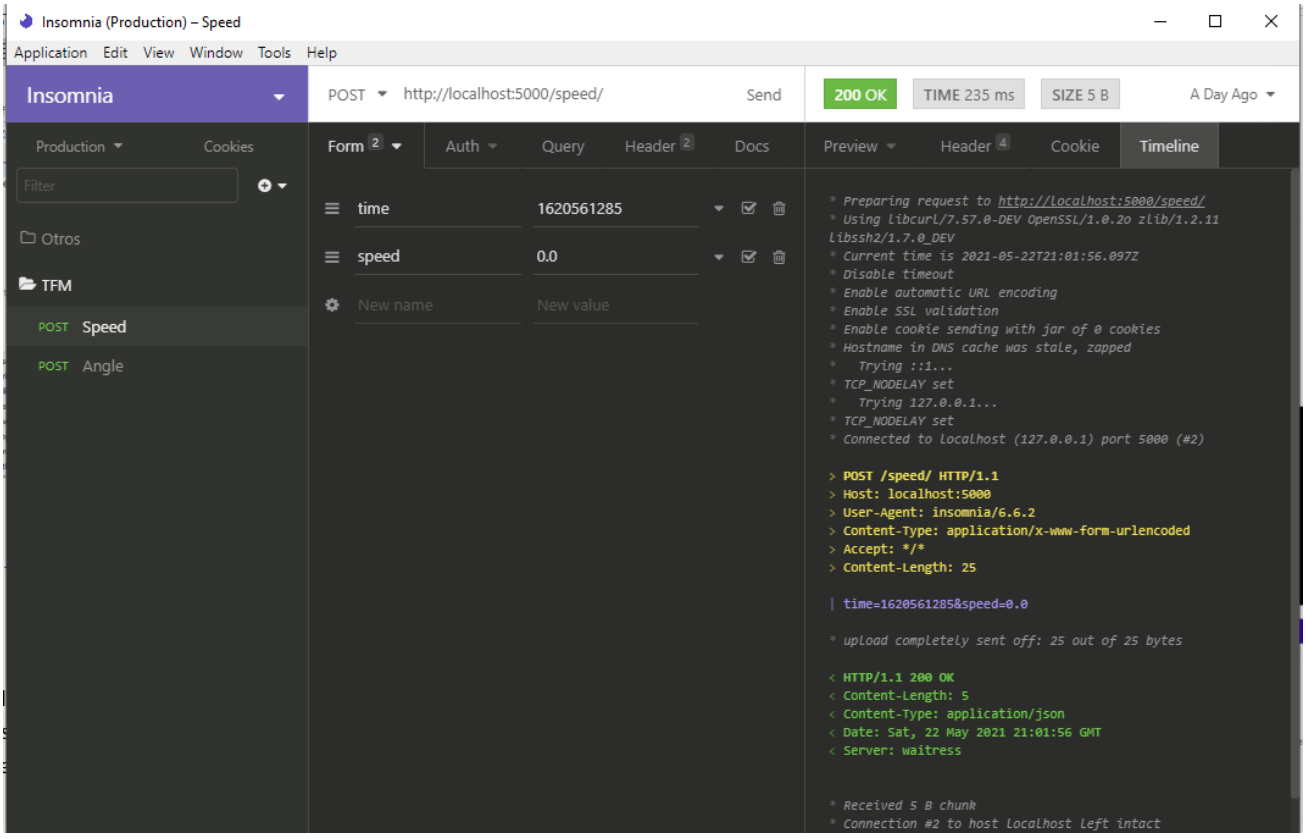


Ilustración 42 Petición y respuesta al entorno productivo de nuestro servicio. Se puede apreciar en la pestaña de timeline que estamos utilizando "waitress", tal y como recomienda Flask en su documentación.

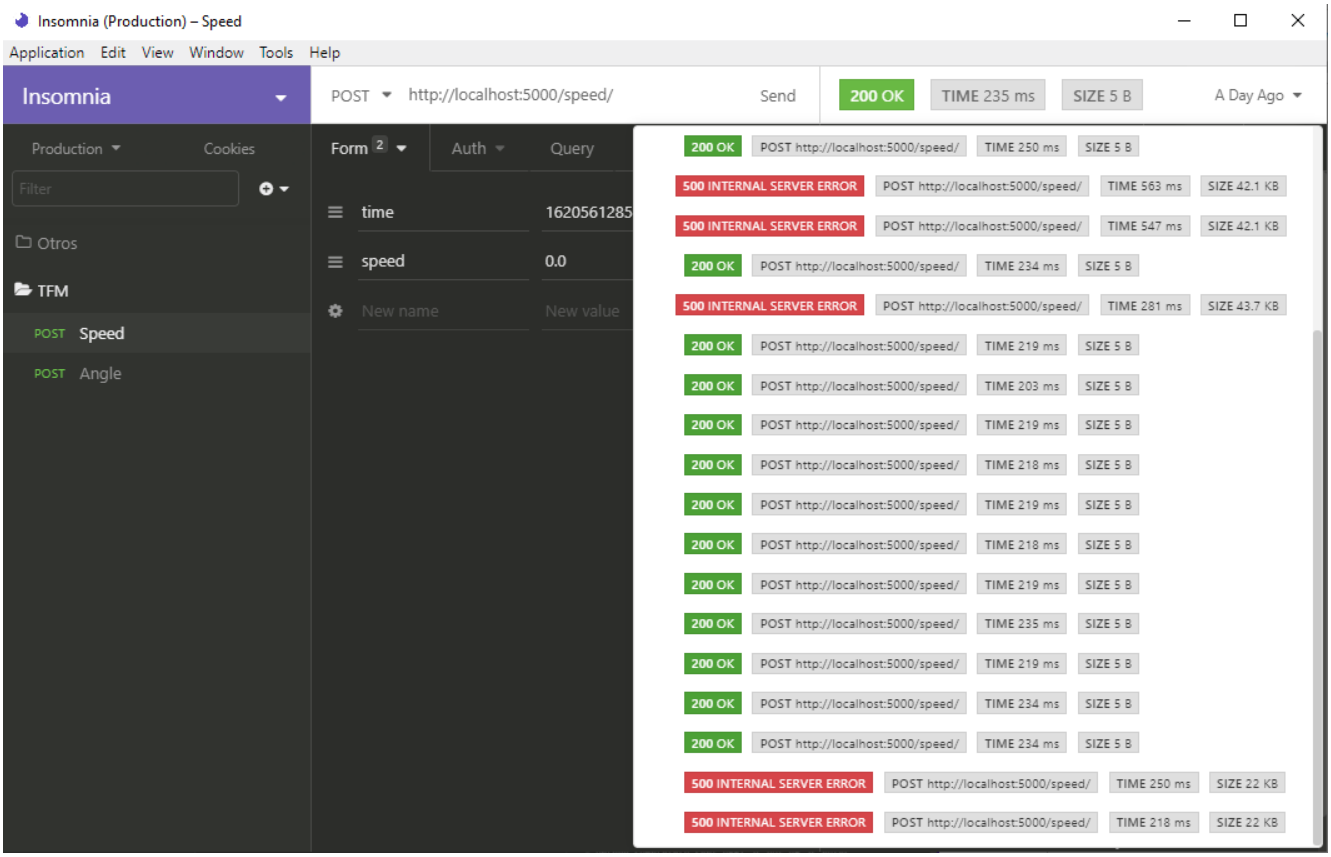


Ilustración 43 Histórico de peticiones realizadas durante el desarrollo.

GPS Logger [46]

La aplicación Android desarrollada se hizo en base a esta aplicación open source [36].

Antes de plantear la creación de un cliente que envié datos a un servidor, se utilizó la aplicación de la appstore de Android para recoger datos en GPX y poder generar un modelo predictivo inicial.

Se pueden encontrar las capturas y funcionamiento de esta aplicación en el apartado CSI 6.3: Manual de Usuario.

Android Studio [47]

Entorno de desarrollo para construir aplicaciones Android. Permite crear dispositivos virtuales en caso de no disponer de un dispositivo físico, así como instalar la aplicación en dispositivos físicos.

Incluye herramientas de depuración, log, analizador de código, así como aplicaciones hechas por la comunidad. Permite simular acciones avanzadas, en nuestro caso, estábamos interesados en simular coordenadas GPS o un trayecto GPX.

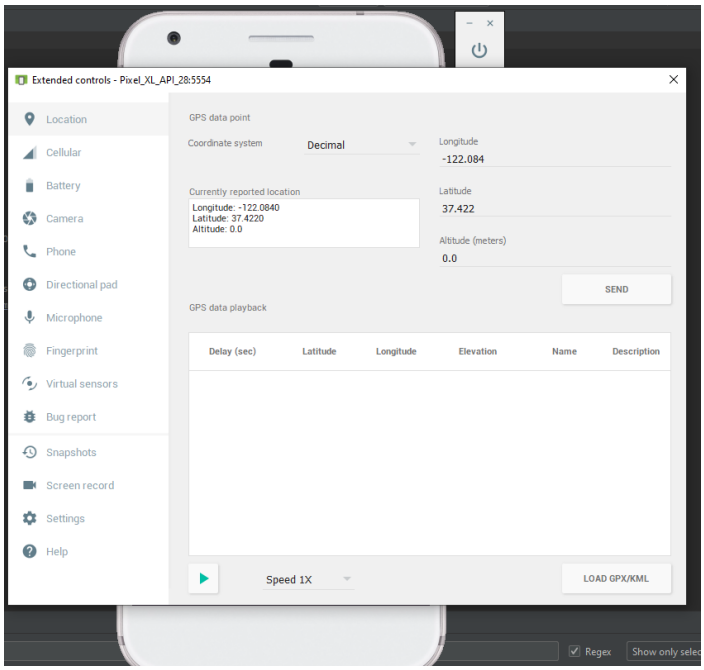


Ilustración 44 Simulador de posición

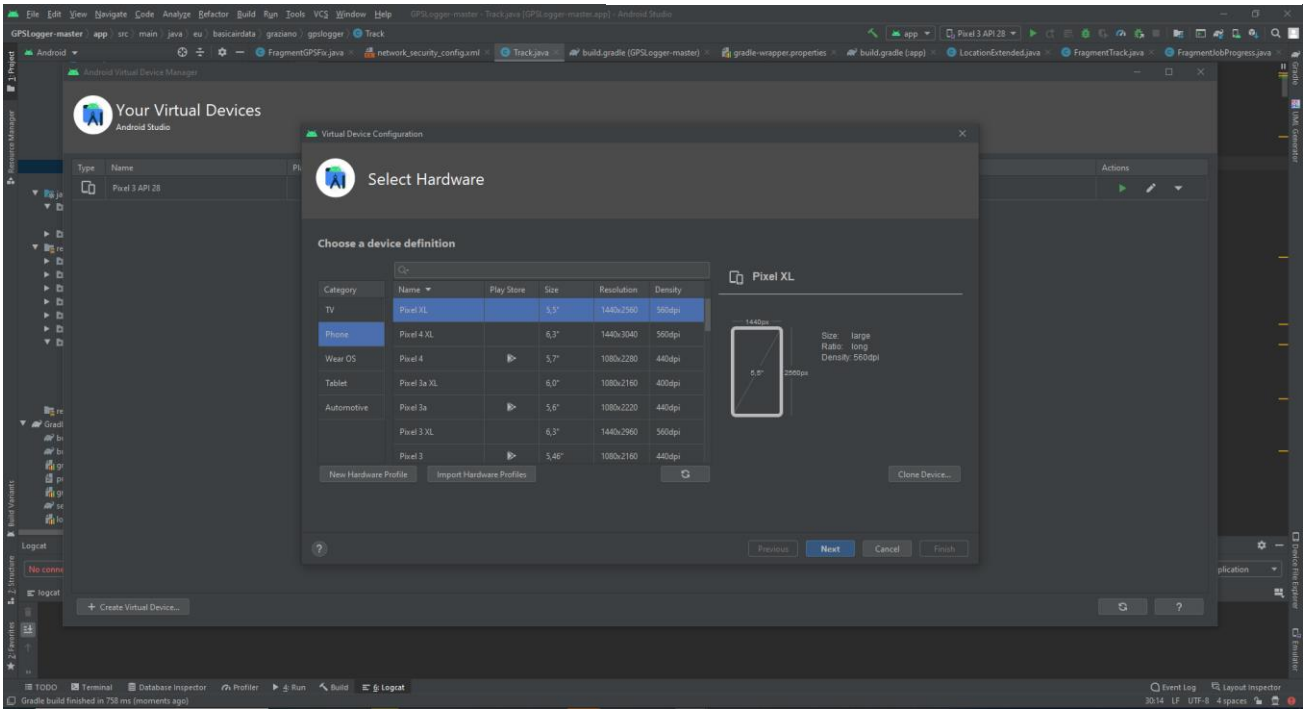


Ilustración 45 Creando un dispositivo móvil virtual

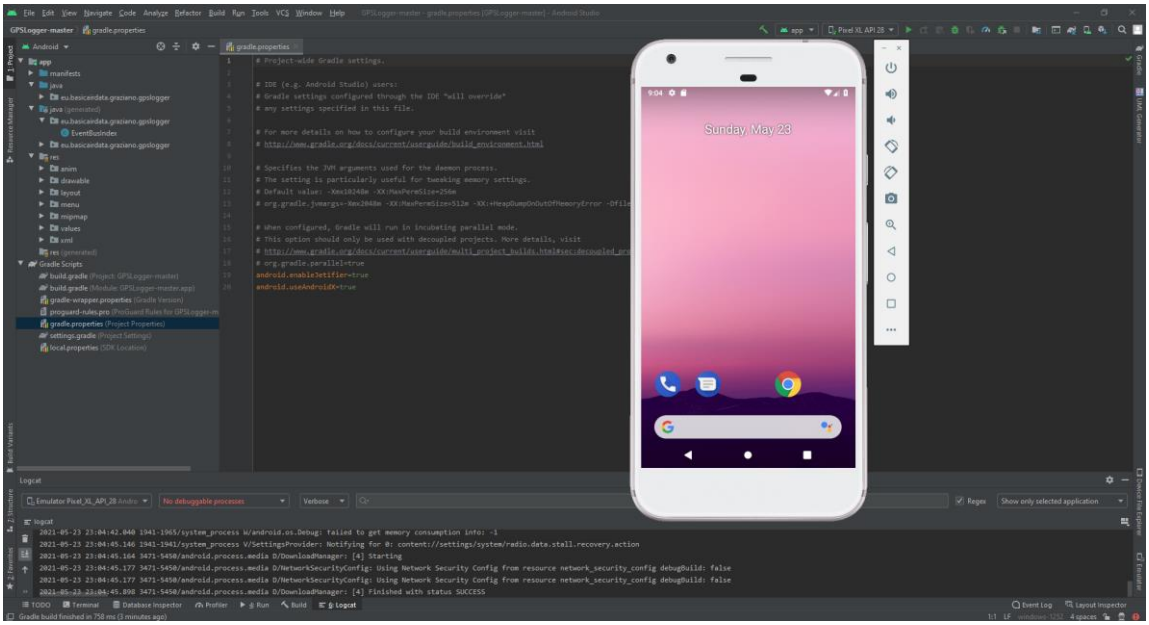


Ilustración 46 Dispositivo Android virtual listo para ser utilizado

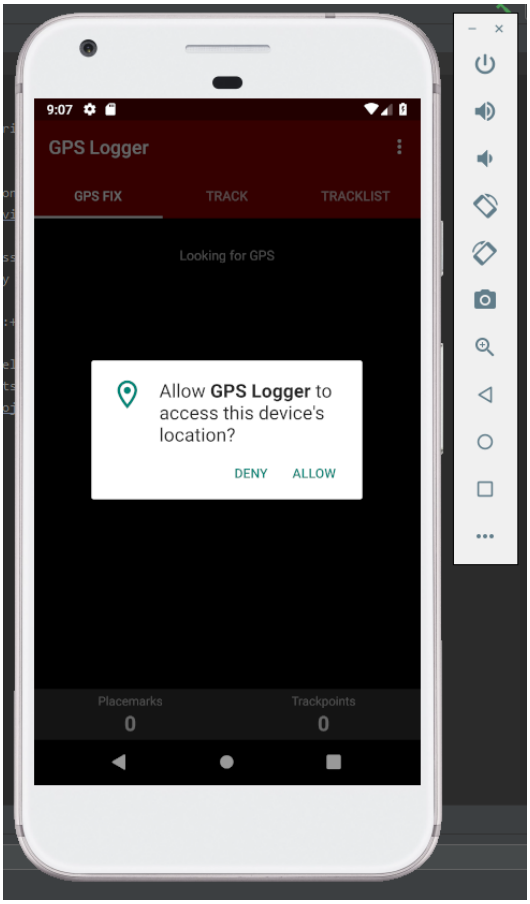


Ilustración 47 Abriendo GPS Logger en un entorno virtual

CSI 2: EJECUCIÓN DE LAS PRUEBAS UNITARIAS

A continuación, se exponen los resultados y problemas encontrados durante la ejecución de las pruebas unitarias.

Arduino

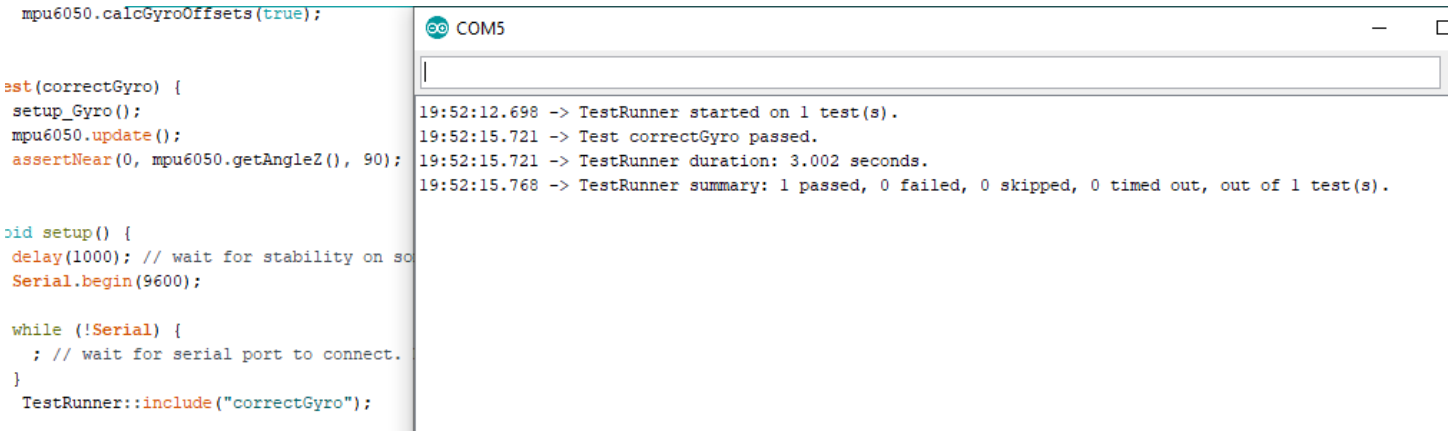


Ilustración 48 Prueba Unitaria con AUnit

Tras realizar la instalación del framwork, se descubre que AUnit no es compatible con la placa Arduino UNO WIFI Rev2, por esta razón se descarta implementar la prueba unitaria de conectividad, delegando dicha prueba a la sección de pruebas de integración.

Se cambia de placa, utilizando una Arduino UNO, solamente con el giroscopio conectado y se ejecutan las pruebas. Se confirma que se ha captado un ángulo tras la inicialización.

Modelo predictivo

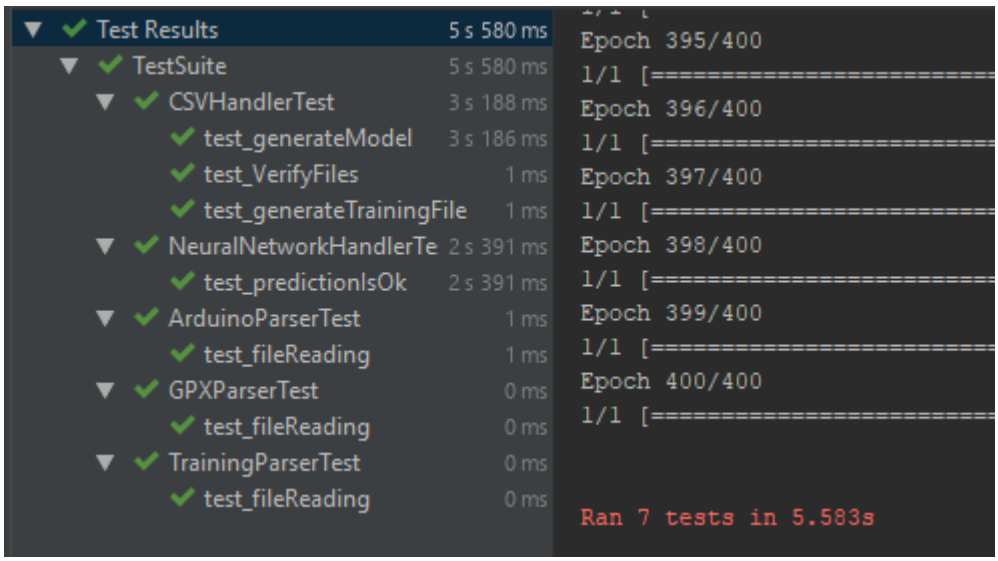


Ilustración 49 Ejecución de pruebas unitarias para lectura, entrenamiento, y generación de modelos predictivos

Tras ejecutar las pruebas se detecta que durante la lectura de ficheros nunca se llegaban a cerrar los ficheros, gracias a un mensaje de alerta ofrecido al ejecutar los test. Se realizan acciones correctivas para cerrar los ficheros tras su utilización.

Debido a que se ha mantenido un Split de datos al 70-30, durante las pruebas se han tenido que introducir 2 entradas, en lugar de una sola, ya que si no daba problemas de ejecución. Otra posible solución habría sido comprobar la cantidad de datos que llegan en el conjunto de datos.

Para probar la lectura de ficheros se han creado ficheros con un solo registro en cada uno. Otra opción habría sido generar los ficheros internamente en la prueba y eliminarlos a posteriori.

Servidor

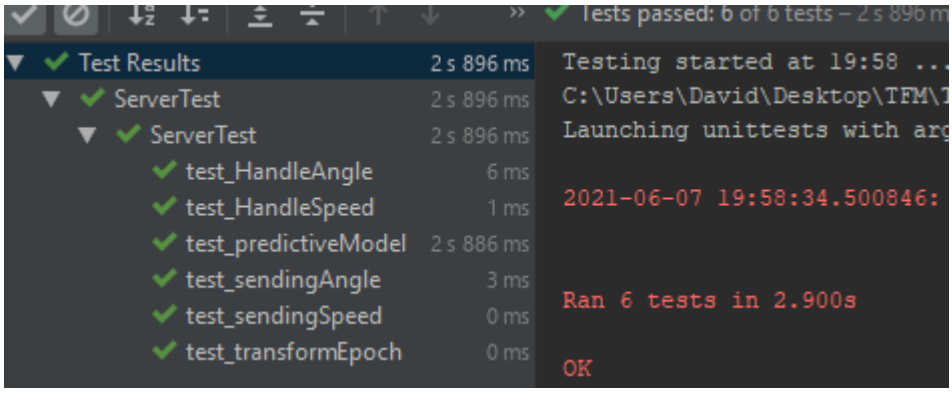


Ilustración 50 Ejecución de las pruebas del Servidor

En este caso todos los resultados tras la ejecución de pruebas unitarias son correctos.

CSI 3: EJECUCIÓN DE LAS PRUEBAS DE INTEGRACIÓN

Las pruebas de integración que se habían planteado servían para comprobar la correcta conexión de red entre Arduino/Android y el servidor.

Tabla 17 Resultado de las pruebas de conexión

Pruebas de conexión	
Prueba	**Resultado Esperado**
Petición desde aplicación Android	La petición llega al servidor
	Resultado Obtenido
	La petición **no** llega al servidor
Prueba	**Resultado Esperado**
Petición desde aplicación Arduino	La petición llega al servidor
	Resultado Obtenido
	La petición **no** llega al servidor

Se ha podido comprobar que hay algún tipo de conexión entre los clientes y el servidor. En el proyecto nuestro servidor se encuentra desplegado en un ordenador portátil en la misma red que los otros clientes.

Tras un análisis se descubre que el portátil tiene el firewall activado y rechaza todas las conexiones entrantes. Se cambia la configuración para permitir peticiones entrantes, abriendo el puerto 5000, que es donde teníamos desplegado nuestro servidor. Tras realizar este cambio se comprueba que las peticiones llegan correctamente.

CSI 4: EJECUCIÓN DE LAS PRUEBAS DE CARGA DEL SERVIDOR

Tras haber ejecutado las pruebas definidas, los resultados obtenidos han sido los siguientes:

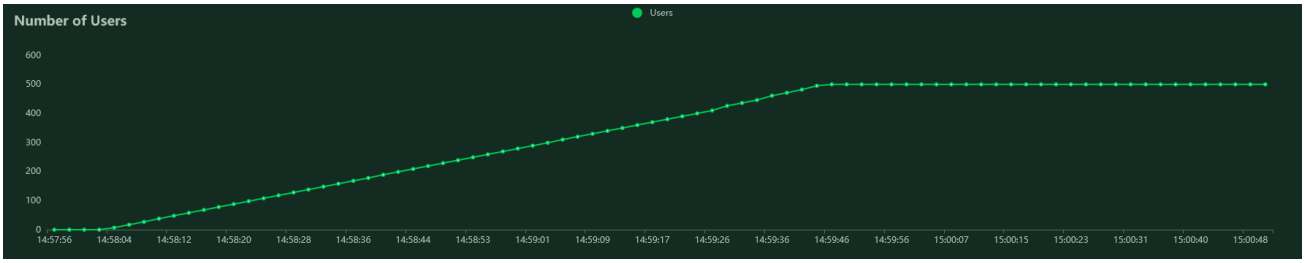


Ilustración 51 Incremento de vehículos (usuarios) a lo largo del tiempo

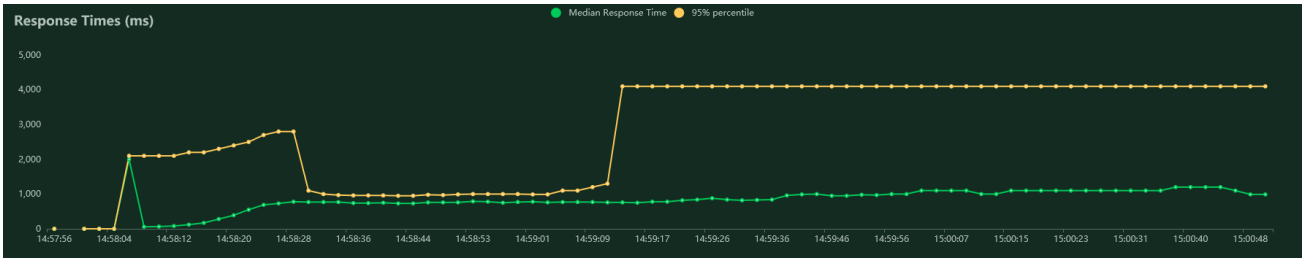


Ilustración 52Tiempo de respuestas a lo largo del tiempo

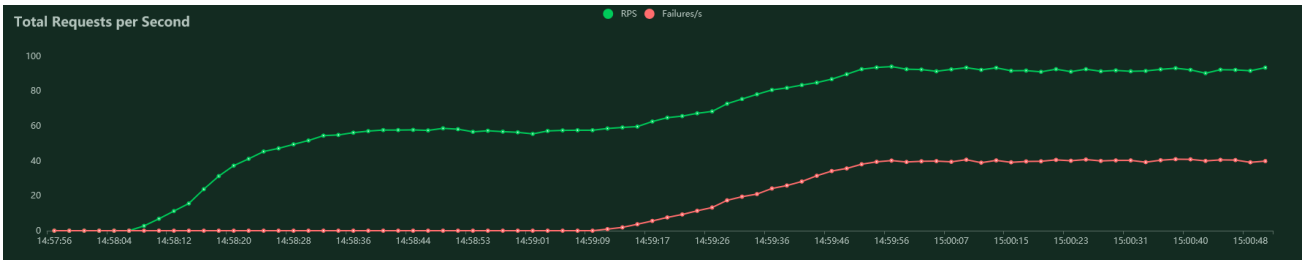


Ilustración 53 Número de peticiones por segundo versus numero de peticiones fallidas

Podemos comprobar que la primera petición realizada tuvo un elevado tiempo de respuesta, y a partir de esta petición el sistema comienza a funcionar de manera "optima". El incremento de vehículos, y con ello el número de peticiones se mantiene estable, sin ningún tipo de fallo hasta alcanzar los 340 vehículos, con un número aproximado de 58 peticiones por segundo. Es justo este momento en el que también comienza a incrementarse.

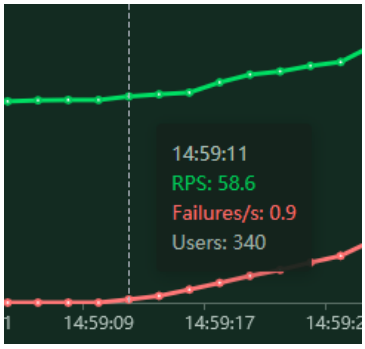


Ilustración 54 Momento de aparición de fallos en las respuestas

El incremento de peticiones y el número de respuesta de errores es constante, analizando los errores que vemos en Locust y los logs de Waitress podemos comprobar que los errores mostrados son debido a que las peticiones se han encolado y todavía no se han respondido, o bien se ha llegado al límite de peticiones abiertas "*no longer accepting new conexions, total open connections reached the conection limit*". Este último error es el que aparece registrado en Locust.

```
HTTPConnectionPool(host='localhost', port=5000): Max retries exceeded with url: /angle?time=1621774790&angle=1 (Caused by NewConnectionError(': Failed to establish a new connection: [Errno 10061] [WinError 10061] No se puede establecer una conexión ya que el equipo de destino denegó expresamente dicha conexión.'))
```

Ilustración 55 Error observado en Locust

El número de vehículos soportados, por lo tanto, con un solo servidor sin cambiar ningún tipo de configuración se encuentra entre los 330 y 340. Seguramente este número pueda ser incrementado considerablemente jugando con las configuraciones proporcionadas por Waitress, o mediante la utilización de diversos servidores y balanceadores de carga. Hay que mencionar que quizá mereciese

la pena trasladar este servidor a otra tecnología, o con otro planteamiento con mayor escalabilidad como serverless.

CSI 5: EJECUCIÓN DE LAS PRUEBAS DE RENDIMIENTO

Se ha realizado un estudio con diferentes modelos predictivos y parámetros para analizar cuál de todos tiene mejor precisión. Todos los modelos han sido entrenados con 4 horas de datos de conducción. Este set de datos se ha dividido al 70% para realizar entrenamiento y al 30% para comprobar la precisión obtenida.

Ilustración 56 Equipo instalado en funcionamiento para la obtención de datos de entrenamiento

Ilustración 57 Datos capturados durante la conducción

Redes neuronales:

Todas las redes neuronales probadas se han basado en la arquitectura CNN.

Los parámetros alterados durante las diferentes pruebas han sido:

- Número de capas: Numero de convoluciones antes de llegar al output, como ejemplo, la red neuronal mostrada en la Ilustración 59 cuenta con 2 capas. Ignoramos la capa inicial y final pues esas son constantes.
- Número de neuronas por capa: Cantidad de neuronas que posee una capa. La Ilustración 59 cuenta con 4 neuronas por capa.
- Porcentaje de conexión entre diferentes capas: Cuantas conexiones hay entre las capas. Se altera mediante el parámetro dropout. La Ilustración 59 cuenta con un 100% de conexiones, es decir, sin dropout, pues todas las neuronas de una capa apuntan a todas las neuronas de la siguiente.
- Algoritmo optimizador: La utilización de algoritmos optimizadores es utilizado habitualmente para reducir la ratio de error en la predicción. Las métricas internas utilizadas son la velocidad de convergencia y la generalización ante nuevas entradas.
    - Velocidad de convergencia: Cuántas veces se entrena el modelo hasta alcanzar una alta precisión. Tras un número determinado de veces el entrenamiento no incrementará la precisión por muchas iteraciones que añadamos.
    - Generalización de nuevas entradas: Utilizando esta métrica se valora la precisión que hay en el conjunto de datos contra la precisión frente a nuevas entradas. Tras un número de entrenamientos, el porcentaje de errores devuelto será mínimo, y si seguimos entrenando, alguno de los porcentajes de error se incrementará.

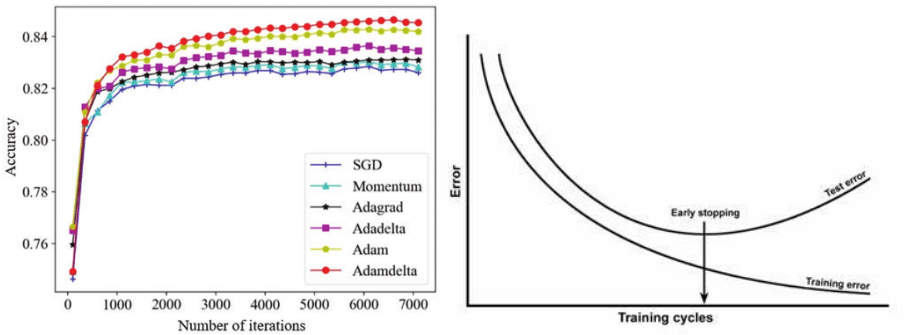


Ilustración 58 A la izquierda, ejemplo gráfico de la velocidad de convergencia frente al número de iteraciones [54]. A la derecha el momento óptimo de parar entrenamiento minimizando los errores en la generalización [55].

En la figura anterior se puede comprobar que, tras un número de entrenamientos, no merece la pena seguir entrenando.

Los algoritmos que se han decidido utilizar han sido Adam [50] y SGD [51]. Dos implementaciones populares que cubren cada uno, uno de los casos anteriores. Existen muchos otros optimizadores que se podrían haber utilizado [52] [53].

La única variable que se ha mantenido constante ha sido la cantidad de datos a procesar en cada batch. Como sabemos la frecuencia de los datos, se han agrupado en bloques de 15 segundos.

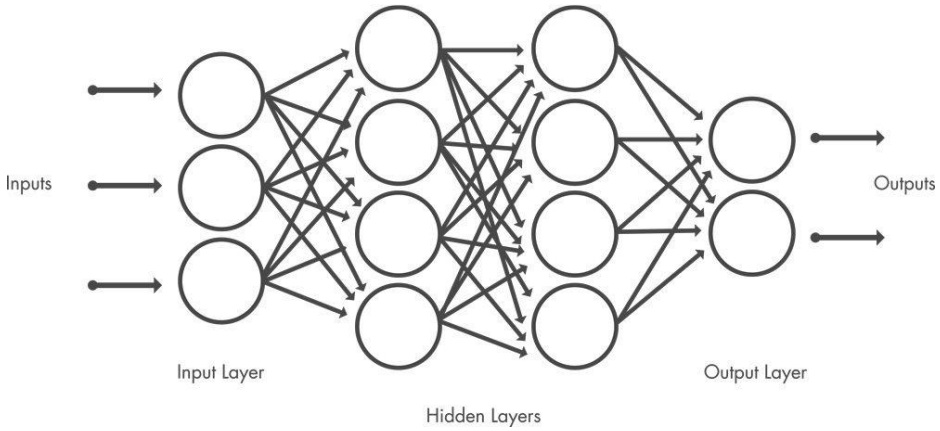


Ilustración 59 Red neuronal CNN con tres entradas y dos salidas

Tabla 18 Resultado de estudio con redes neuronales

Epoch	Nº Capas	Nº Neuronas	Dropout	Optimizador	Tiempo de entrenamiento	Training Data: 70%				Splitted Data 30%			
						Accuracy	Loss	Precision	Recall	Accuracy	Loss	Precision	Recall
200	3	8	0%	ADAM	58s	0.8728	0.3278	0.6674	0.2871	0.8681	0.3242	0.6450	0.2774
400	3	8	0%	ADAM	113s	0.8730	0.3331	0.6922	0.3508	0.8709	0.3392	0.6759	0.3223
4000	3	8	10%	ADAM	1181s	0.8768	0.3258	0.8148	0.2157	0.8778	0.3268	0.7333	0.1917
4000	3	8	0%	ADAM	1163s	0.8835	0.3067	0.8624	0.2634	0.8815	0.3180	0.7680	0.2238
400	3	8	10%	ADAM	122s	0.8678	0.3410	0.8687	0.1001	0.8626	0.3441	0.9032	0.0801
400	3	8	25%	ADAM	116s	0.8589	0.3581	0.8152	0.1092	0.8579	0.3688	0.7400	0.0774
400	**5**	**8**	**0%**	**ADAM**	**119s**	**0.8855**	**0.3089**	**0.7027**	**0.3751**	**0.8790**	**0.3089**	**0.7440**	**0.3472**
400	8	8	0%	ADAM	134s	0.8856	0.3068	0.7616	0.3458	0.8902	0.3103	0.7615	0.3304
400	3	8	0%	SGD	108s	0.8554	0.3510	0.6479	0.3192	0.8725	0.3236	0.5243	0.2405
400	3	8	10%	SGD	115s	0.8677	0.3557	0.7520	0.1568	0.8663	0.3630	0.7475	0.1786
4000	3	8	0%	SGD	1157s	0.8802	0.3187	0.6711	0.3427	0.8781	0.3288	0.7108	0.3227
4000	3	8	10%	SGD	1103s	0.8768	0.3196	0.8566	0.2071	0.8657	0.3501	0.8571	0.2059

También se ha estudiado cómo de bien responden los modelos estadísticos, en este caso, no se ha tocado ninguna parametrización, dejando la configuración por defecto proporcionada por Sci-Kit

Modelo	**Trained Data 70%**			**Splitted Data 30%**		
	Precision	**Recall**	**F-Score**	**Precision**	**Recall**	**F-Score**
KNN	**0.845**	**0.666**	**0.745**	**0.713**	**0.541**	**0.615**
CART	1.000	0.997	0.999	0.683	0.636	0.659
Redes Bayesianas	0.533	0.242	0.333	0.493	0.216	0.300
SVM	**0.963**	**0.769**	**0.855**	**0.840**	**0.389**	**0.532**

Tabla 19 Tabla comparativa de modelos estadísticos

Como se había explicado anteriormente, se ha dividido el conjunto de datos en dos, el conjunto de datos de entrenamiento con un 70% y el conjunto de datos de validación con el 30% restante. En ambos conjuntos sabemos cuál debería ser la respuesta correcta. Por ello entrenamos los modelos con un 70% de datos y analizamos su precisión. Y después intentamos predecir un 30% de datos (que los modelos nunca habían tenido como entrada), y vemos cómo de bien se adaptan a información que no se encontraban presente en el conjunto de datos inicial.

***Accuracy*:** Representa cuantas veces categorizamos correctamente una entrada. Como ejemplo si en 100 entradas de datos, categorizamos correctamente 95 de ellas, tendríamos un *accuracy* de 0.95.

Loss: Representa la diferencia entre el resultado real y el valor que se ha predicho. Puede ser visualizado como la distancia al valor real. Es un dato que hay que analizar subjetivamente, no va a tener el mismo impacto humano un *loss* de 0.1 en una red neuronal que clasifica imágenes por categoría (edificios, paisajes, personas…) que una red neuronal que intenta predecir cáncer en etapas tempranas.

Precision: Identifica el ratio de casos predichos sobre el número total de que tener esa precisión.

Recall: Exhaustividad, identifica el ratio de positivos identificados correctamente.

F-Score: Valor-F, combina los valores de precision y recall.

Es importante entender que tener un alto accuracy no siempre es correcto, en casos en los que las clases están desbalanceadas, pues podríamos generar un modelo con alta accuracy pero que no sea granular. Por ejemplo, si el 0.01% de las personas tiene tumores, y creamos un modelo que siempre diga que no tienes un tumor, su accuracy será del 99.99%, un valor que es muy alto, pero que puede llevarnos a confusión, pues realmente no está detectando en qué casos sí que hay tumores, es ahí donde precisión y recall pueden ayudarnos a diferenciar.

En el caso de este proyecto, no podemos afirmar con seguridad en que porcentaje durante una conducción habitual habría que poner intermitentes, pues depende del trayecto y situación concretas.

Analizando los resultados en redes neuronales vemos que incluyendo dropout afectamos al recall de los resultados, reduciendo nuestro resultado de predicción, además, ADAM parece tener mejores resultados que SGD.

En cuanto a los modelos estadísticos, a simple vista destaca CART, pero una precisión y recall tan elevados pueden ser debido a un sobreajuste. Las redes bayesianas no tienen tan buen resultado ante el subconjunto de datos de validación. KNN y SVM parecen dar buenos resultados. Una de las hipótesis que se había valorado es que con los datos que tenemos (velocidad y ángulo de volante) se podría inferir una función gráfica que representase que, a cierta velocidad, si se supera X ángulo de giro en valor absoluto haya que activar el intermitente, es decir, que los valores recabados estuviesen relacionados, ya que no es lo mismo girar el volante una vuelta yendo a 120km por hora que a 5km/h.

Se han indicado en negrita los posibles modelos a utilizar con los datos recogidos. Habrá que tener en cuenta que, si en el futuro se añaden otros datos, habrá que repetir este estudio.

Si comparamos los resultados con otros estudios similares, como puede ser "T*urn Signal Prediction: A Federated Learning Case Study*" [6] se han obtenido unos resultados bastante similares. En ese estudio se utilizaron conjuntos de datos en los que se había recogido la velocidad, la aceleración, si se pulsó el freno o no, así como el ángulo del volante.

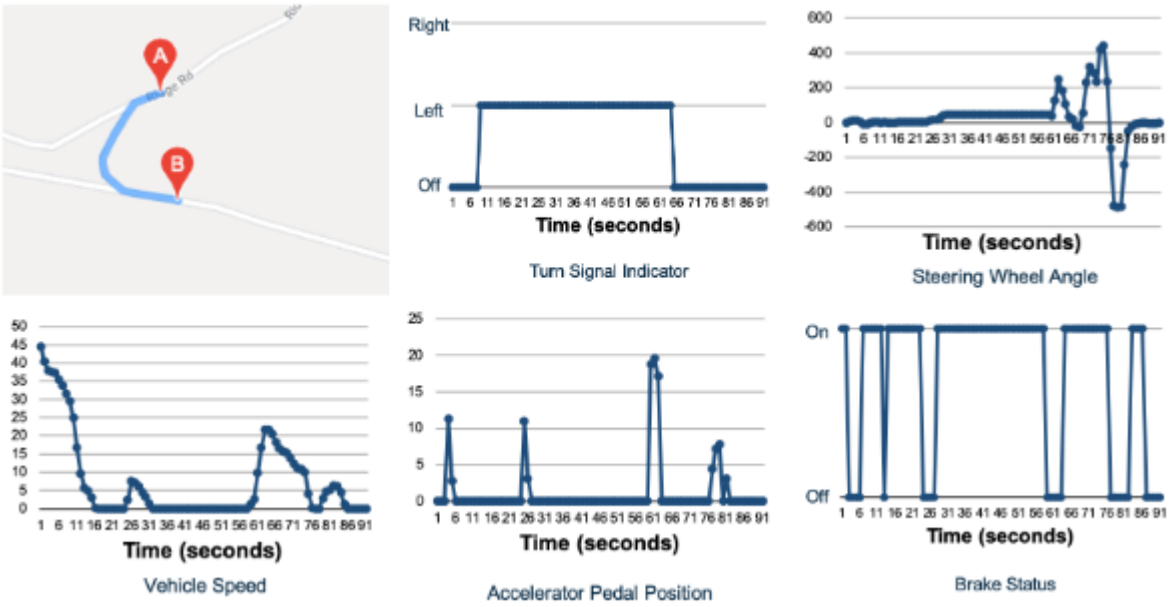


Ilustración 60 Ejemplo de trayecto del estudio con el que nos comparamos

En este estudio se planteó una red neuronal que sigue una arquitectura CNN, con una variante denominada Long Short-Term Memory (LSTM) [54]. A continuación, se muestra una comparativa de precisiones.

Approach	Accuracy(%)			
	Overall Weighted	Off	Left	Right
Central	90.4	92.3	85.6	86.5
Federated	91.1	93.9	85.4	82.3

Ilustración 61 Resultados del paper

Estudio	Accuracy (media)
Turn Signal Prediction: A Federated Learning Case Study	87.2%
Nuestra red neuronal	88.2%

Para calcular la media, en el caso del artículo de investigación con el que nos comparamos hemos utilizado los resultados de la fila "Federated". En nuestro caso hemos utilizado los números tanto del conjunto de datos de entrenamiento como el conjunto que habíamos apartado.

Podemos comprobar que tenemos un Accuracy bastante similar, con un incremento no demasiado resañable. Donde vemos diferencia con este estudio es en el recall. Esto es debido a la ingente cantidad de datos (8 millones de entradas de datos) que se utilizaron en el estudio.

También podemos comparar el F-Score, que como explicamos previamente, combina diversos valores, entre ellos el recall. Es por esto último que en este caso no hemos sido capaces de alcanzar esos porcentajes.

Estudio	F-Score (Media)
Turn Signal Prediction: A Federated Learning Case Study	87.4%
Nuestro modelo SVM	74.5%

A pesar de ello, hemos tenido unos resultados muy similares, con un conjunto de datos recopilado individualmente y con menos cantidad de datos.

En las conclusiones y ampliaciones también hablaremos de ello, pero uno de los aspectos clave que puede ser incorporado para incrementar tanto la predicción como el margen de tiempo de predicción es incorporar la posición de los ojos y estudiar cuanto tiempo pasamos mirando ciertas zonas durante la conducción. Un estudio realizado en 2011 en "Lane Change Intent Prediction for Driver Assistance:On-Road Design and Evaluation" [14] obtuvo resultados concluyentes con un margen de anticipación de hasta 3 segundos, utilizando dicho dato.

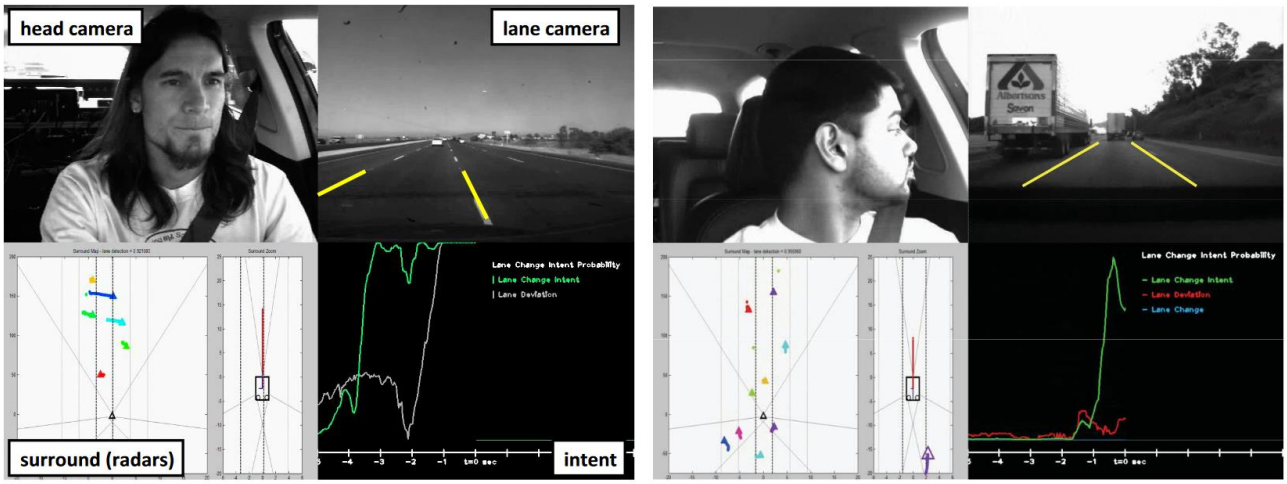


Ilustración 62 Estudio de Lane Change Intent Prediction for Driver Assistance:On-Road Design and Evaluation

CSI 6: ELABORACIÓN DE LOS MANUALES DE USUARIO

En esta sección se mostrarán los aspectos más relevantes para instalar, mantener y modificar el proyecto en diferentes manuales.

CSI 6.1: Manual de Instalación

Existen tres elementos a instalar: la aplicación Android, la placa Arduino y el servidor. En esa sección se tratarán cada uno de estos elementos.

Instalación de la aplicación Android

Para instalar la aplicación desde Android Studio hay que seguir los siguientes pasos

1. Activar el modo desarrollador en el dispositivo en el que queremos instalar la aplicación
2. Conectar el dispositivo al ordenador.
    a. Si hiciese falta, instalar los controladores necesarios para que el dispositivo sea reconocido.
3. Arrancar Android Studio y cargar el proyecto.
4. Seleccionar nuestro dispositivo en la lista de desplegables.
5. Pulsar el botón de ejecución.
6. Aceptar la solicitud de instalación en nuestro dispositivo.

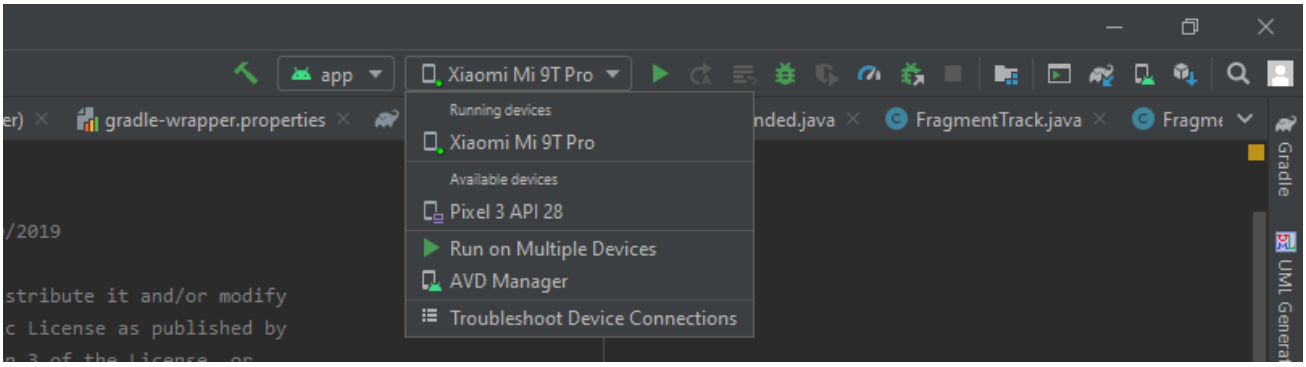


Ilustración 63 Instalación desde Android Studio

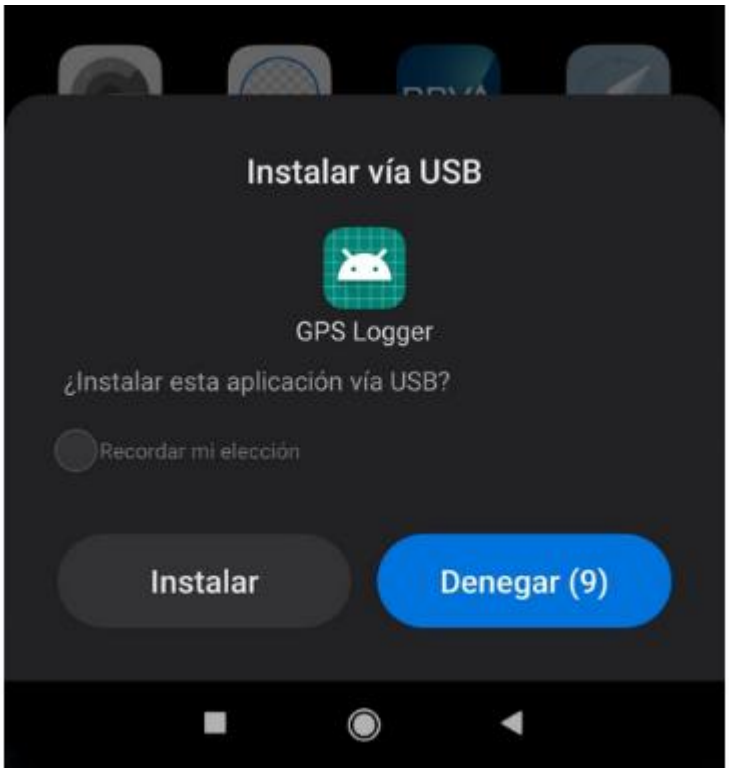


Ilustración 64 Permisos de instalación vía USB

Una alternativa a este método de instalación es la generación de un fichero de instalación APK firmado, que se pueda instalar directamente desde el sistema de ficheros del dispositivo, o incluso subir la aplicación a un repositorio de aplicaciones.

Ilustración 65 Dando permisos de GPS a la aplicación

Instalación de la placa Arduino

Una vez se hayan decidido los sensores a utilizar, se cargará el programa en la placa utilizando Arduino IDE [42]. Además, durante el cargado de datos, habrá que configurar en el fichero "arduino_secrets.h" el nombre y contraseña de red a la que conectarse, así como la IP/URI del servidor al que enviar información.

Tras ello se colocará la placa en el vehículo de manera que el giroscopio este acoplado en el volante, captando el ángulo de giro. El método de sujeción dependerá del vehículo y herramientas a utilizar. En este proyecto no se ha establecido un estándar a seguir. En este caso nosotros hemos colocado la Arduino detrás del volante con cinta, sin embargo, también se podrá atornillar dentro del chasis del vehículo, extendiendo cable hasta el sensor. Otro de los puntos clave es la alimentación de la placa. En nuestro proyecto hemos utilizado una pila de 9V, en función del tipo de instalación se podrá valorar utilizar la batería del vehículo.

Durante la elección del lugar, hay que recordar que la placa no debe afectar a una conducción segura.

Instalación del Servidor

Para instalar el servidor será necesario que la maquina disponga de la versión 3.8 de Python.

Las librerías instaladas en nuestro entorno, tanto para el servidor como para la generación y entrenado de modelos predictivos son las siguientes. Este listado ha sido obtenido mediante PyCharm y algunas de los paquetes mostrados son paquetes necesarios para el correcto funcionamiento del IDE.

Paquete	Versión
Flask	1.1.2
Flask-Jsonpify	1.5.0
Flask-RESTful	0.3.8
Jinja2	2.11.3
Keras	2.4.3
Keras-Preprocessing	1.1.2
Markdown	3.2.2
MarkupSafe	1.1.1
Pillow	7.2.0
PyWavelets	1.1.1
PyYAML	5.3.1
Werkzeug	1.0.1
absl-py	0.10.0
aniso8601	9.0.1
astunparse	1.6.3
cachetools	4.1.1
certifi	2020.6.20
chardet	3.0.4
click	7.1.2
cycler	0.10.0
decorator	4.4.2
gast	0.3.3
google-auth	1.21.0
google-auth-oauthlib	0.4.1
google-pasta	0.2.0
gpxpy	1.4.2
graphviz	0.16
grpcio	1.31.0
h5py	2.10.0
idna	2.10
imageio	2.9.0
itsdangerous	1.1.0
joblib	0.16.0
kiwisolver	1.2.0
matplotlib	3.3.0
networkx	2.4
numpy	1.18.5
oauthlib	3.1.0
opencv-python	4.4.0.40
opt-einsum	3.3.0

pandas	1.1.1
pip	20.2.2
protobuf	3.13.0
pyasn1	0.4.8
pyasn1-modules	0.2.8
pydot	1.4.2
pyparsing	2.4.7
python-dateutil	2.8.1
pytz	2020.1
requests	2.24.0
requests-oauthlib	1.3.0
rsa	4.6
scikit-image	0.17.2
scikit-learn	0.23.2
scipy	1.4.1
setuptools	49.4.0
six	1.15.0
sklearn	0.0
tensorboard	2.3.0
tensorboard-plugin-wit	1.7.0
tensorflow-gpu	2.3.0
tensorflow-gpu-estimator	2.3.0
termcolor	1.1.0
threadpoolctl	2.1.0
tifffile	2020.7.24
urllib3	1.25.10
wheel	0.34.2
wrapt	1.12.1

CSI 6.2: Manual de Ejecución

En esta sección se explicará que acciones de configuración iniciales hay que realizar, tras haber instalado todos los elementos, para comenzar a utilizar nuestras aplicaciones.

Ejecución de la aplicación Android

Para que la aplicación Android envíe peticiones al servidor simplemente será necesario abrirla, teniendo acceso a internet en el dispositivo y una señal GPS de la cual recoger los datos.

Si no se dispone de señal GPS no se enviará ningún dato.

Para parar el envío de datos habrá que quitar la conexión GPS, internet o cerrar la aplicación.

Ilustración 66 Solicitud de permisos por parte de la aplicación

Ejecución de la placa Arduino

En cuanto la placa obtenga energía, realiza el procedimiento de inicialización (setup), calibrando el volante, y obteniendo acceso a la red wifi. Una vez dicha inicialización finalice, la placa comenzará a enviar información al servidor.

Para parar el envío de datos habrá que quitarle el suministro de energía a la placa.

CSI 6.3: Manual de Usuario

Tras haber seguido los pasos indicados en Instalación de la aplicación Android, tendremos una aplicación funcional.

El menú principal es el siguiente:

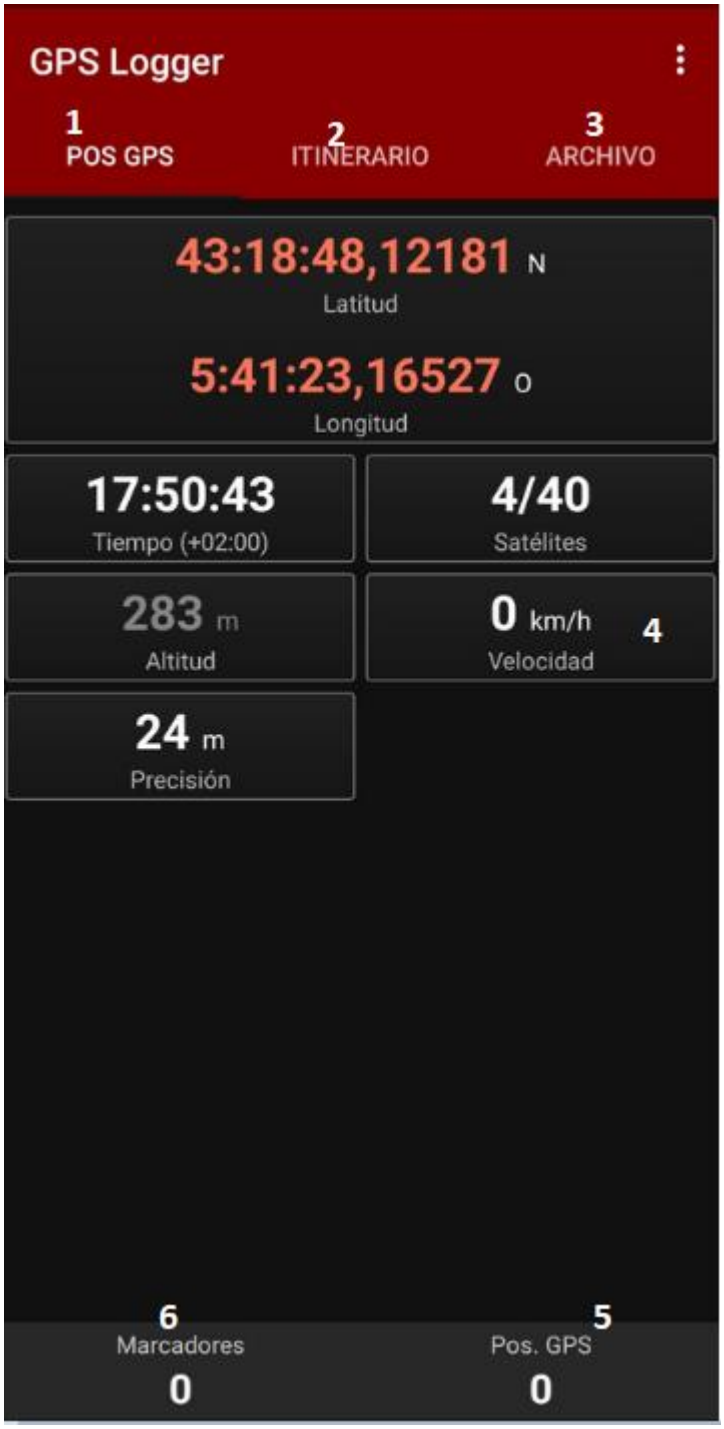


Ilustración 67 Elementos encontrados nada más abrir la aplicación

Los elementos indicados como 1,2 y 3 permiten alternar entre: 1:la información actual, 2: un resumen del trayecto que estemos haciendo, con información como velocidad media o distancia recorrida, y 3: el conjunto de todos los trayectos realizados.

El elemento 4 es el dato enviado al servidor.

El elemento 5 permite comenzar un trayecto.

El elemento 6 permite añadir un marcador al trayecto, esto no tiene ningún tipo de uso en nuestro proyecto.

Además, los tres puntos de la zona superior derecha permiten acceder al menú de configuración.

Al pulsar en el elemento 5, este cambiará de color para indicar que estamos realizando un trayecto.

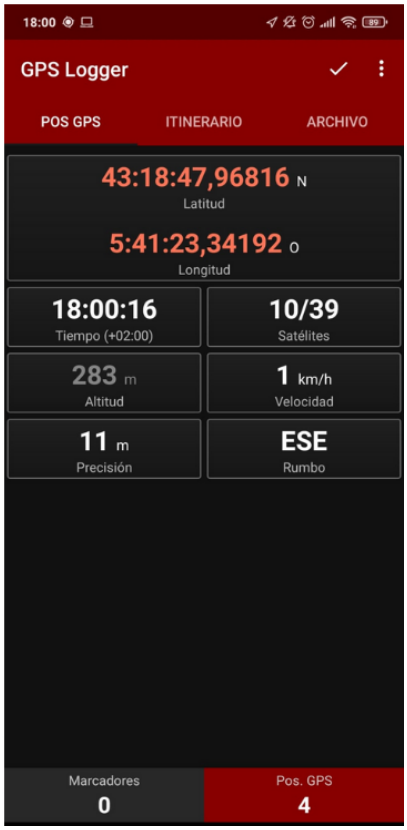


Ilustración 68 Trayecto siendo capturado

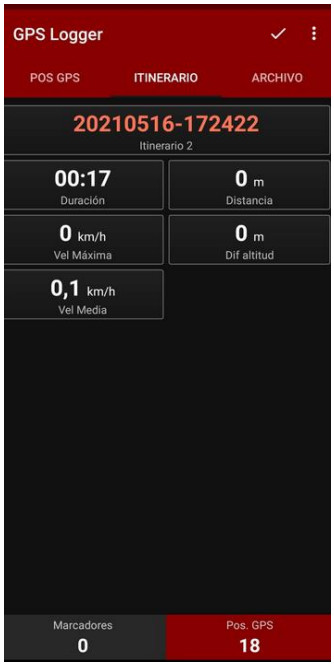


Ilustración 69 Información de un itinerario

Para finalizar la captura de datos de un itinerario, será necesario pulsar dos veces en el check que aparece junto a los tres puntos de las opciones. Este solo será visible si estamos capturando un trayecto.

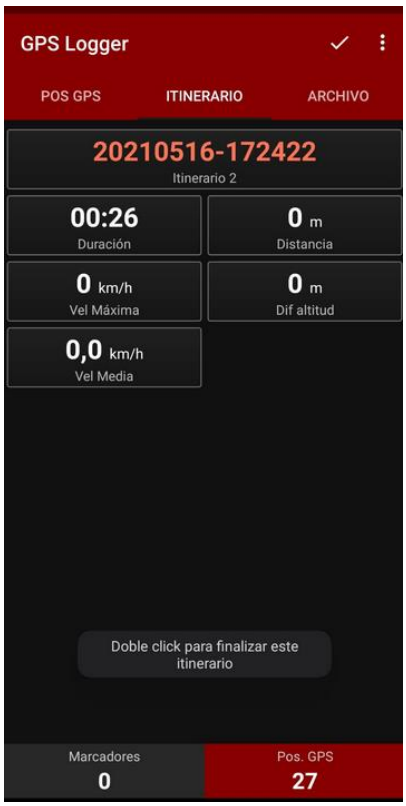


Ilustración 70 Mensaje que aparece al hacer un solo click en el check

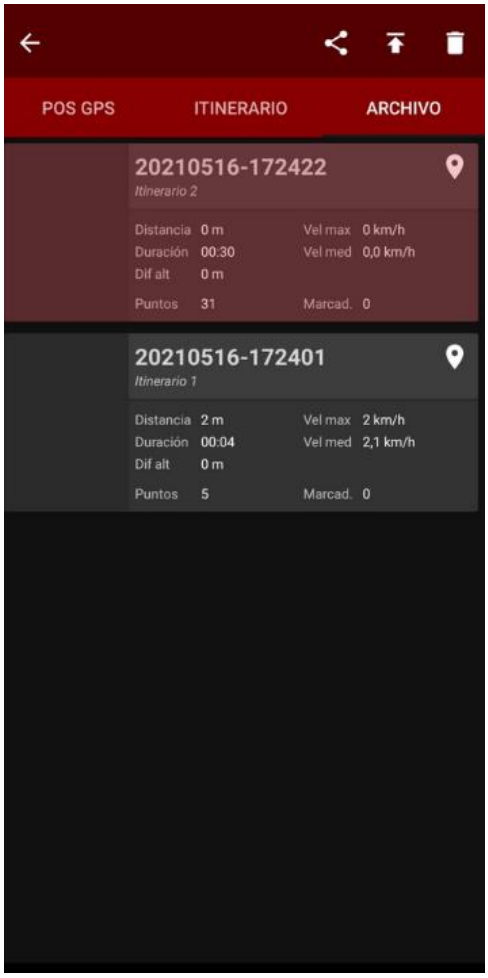


Ilustración 71 Listado de trayectos

Para exportar un trayecto finalizado, hay que mantener pulsado el trayecto, y tras ello pulsar en el icono de compartir.

Ilustración 72 Exportación de ficheros

Es importante tener marcada siempre la opción de exportar GPX, así como la opción de GPS de alta precisión si fuese posible.

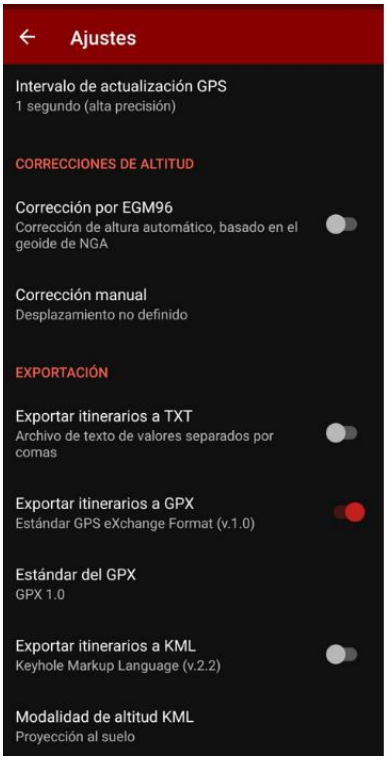


Ilustración 73 Ajustes de la aplicación

CSI 6.4: Manual del Programador

En esta sección se mencionarán los aspectos que se han considerado más relevantes en el aspecto técnico. No se pretende explicar toda la implementación y se asume un conocimiento de programación.

Arduino

Se explicarán las diferentes posibilidades de Arduino, para no cerrar el proyecto solo a ciertos sensores o placas.

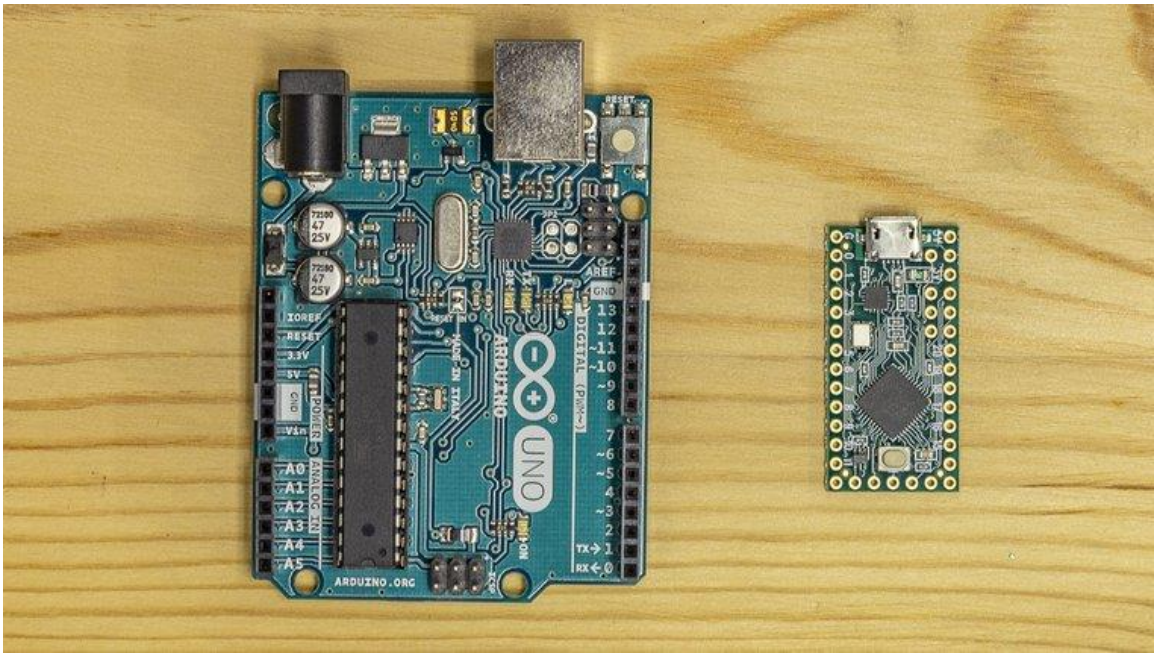


Ilustración 74 Las razones para utilizar diferentes placas pueden ser debido a diversas razones. Una teensy es mucho mas pequeña y puede colocarse en el vehículo de manera mucho más optima.

Usos de diferentes placas compatibles:

En el proyecto se ha utilizado una placa original Arduino, pero cualquier clónico o alternativa (como por ejemplo Teensy [55]) que soporte el número mínimo de conexiones de sensores y sea compatible con el programa proporcionado serviría. Esto puede permitir reducir presupuesto o colocar el proyecto mucho más cómodamente.

En caso de querer utilizar una placa diferente habrá que seguir la configuración indicada por el fabricante para poder compilar el programa con el IDE de Arduino. Esto suele involucrar instalar controladores y/o utilizar el Gestor de Tarjetas de Arduino IDE.

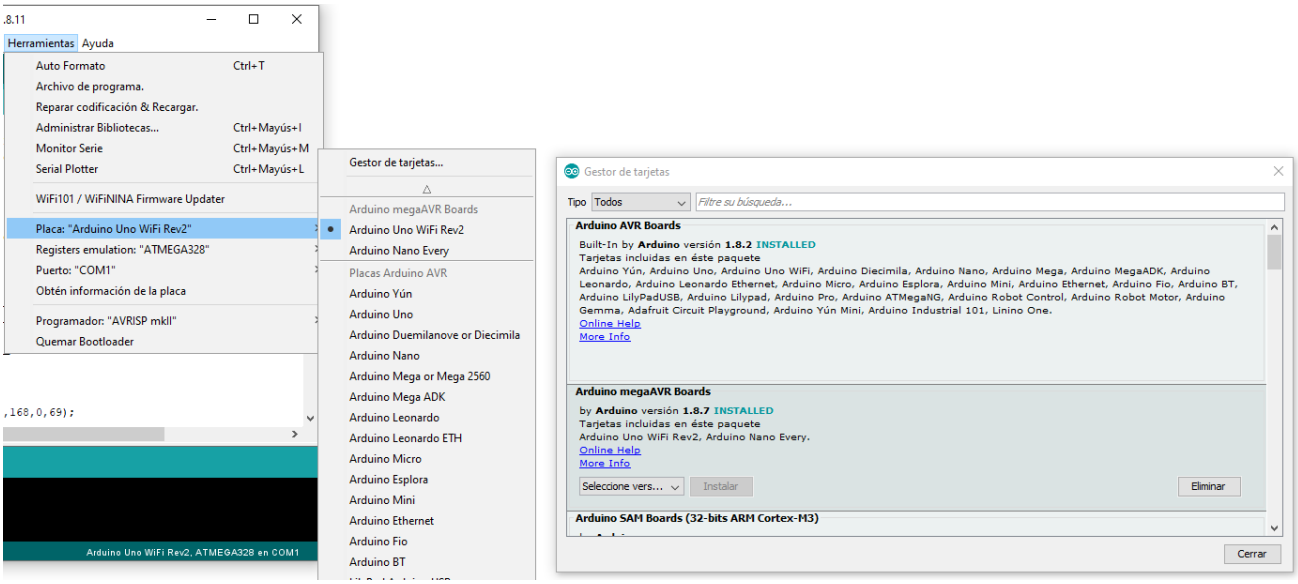


Ilustración 75 Gestor de tarjetas en Arduino IDE

Existen numerosas placas de diferente origen que Arduino [56] como las de Adafruit [57]. las Teensy [58], las Elegoo [59] así como un montón de marcas genéricas.

Utilización de sensores Wifi diferentes:

Todos los sensores Wifi ofrecen una interfaz común denominada "WiFiClient" [60]. Para utilizar cualquier sensor simplemente será necesario realizar las operaciones necesarias en el setup para proporcionar una implementación de WiFiClient en el atributo que se utiliza en el resto del programa.

```
WiFiClient client;

void setup() {
  Serial.begin(9600);

  while (!Serial) {
    ; // wait for serial
  }
  setup_Internet();
  setup_Gyro();
```

Ilustración 76 WifiClient proporcionado en metodo setup_Internet()

En el caso de lo entregado, dicha implementación la gestiona automáticamente la librería WiFiNINA una vez le proporcionamos las credenciales de la red.

Utilización de sensores de giroscopio diferentes:

De nuevo, será cuestión de utilizar las librerías correspondientes para configurarlo y adaptar el método que recoge los ángulos por los definidos en la documentación de la librería.

Se proporciona un método setup_Gyro que realiza todas las tareas de gestión sobre el giroscopio. Estamos utilizando una librería que se encarga de inicializar la conexión y transformar los datos proporcionados del giroscopio a un ángulo en grados.

```
void setup_Gyro(){
   Wire.begin();
   mpu6050.begin();
   mpu6050.calcGyroOffsets(true);
}
```

Ilustración 77 Inicialización de la librería MPU6050

En caso de querer utilizar otro sensor u otra librería sería recomendable utilizar una estructura similar, eliminando los contenidos de este método y proporcionando una implementación propia, las librerías de la comunidad suelen contener la documentación y ejemplos, y en muchos casos se puede modificar el contenido de este método por el código proporcionado de la documentación directamente. Habrá que tener muy en cuenta que pines espera utilizar por defecto dicha librería y estudiar como se pueden utilizar otros en el caso de que varios elementos quieran utilizar el mismo pin. Por ejemplo, previamente en la documentación, en DSI 1: Diseño de Casos de Uso Reales se estableció que el reloj debía estar instalado en los pines 8 y 9. La razón de este requisito es precisamente el caso comentado, en ese proyecto se estaban utilizando diferentes librerías, y varias coincidían en los mismos pines por defecto, por ello se cambiaron los pines del reloj a esos dos.

En la documentación se establecía que el primer parámetro se relacionaba con el SCA del reloj y el segundo con el SCL.

Se espera que un programador que quiera modificar o utilizar diferentes sensores tenga estos factores en cuenta.

```
//Variables Reloj
DS3231 rtc(8, 9);

double timeStep, time, time
double arx, ary, arz, grx,

//Variables acelerometro

#include <MPU6050_tockn.h>
MPU6050 mpu6050(Wire);
```

Ilustración 78 Estableciendo los pines 8 y 9 para el reloj DS3231

Utilización de placas diferentes a Arduino:

En el caso de querer utilizar otro tipo de placas por cualquier razón (tamaño, precio, experiencia…) no compatibles con nuestro proyecto, será necesario reimplementar el proyecto de 0. El único requisito a cumplir es ser capaz de enviar información en una petición REST al servidor definido.

Android

La aplicación ha sido forkeada de GPSLogger [36]. La única modificación realizada ha sido enviar la velocidad cuando se tiene recepción GPS y se ha pulsado en grabar.

```
public void Update() {
    //Log.w("myApp", "[#] FragmentGPSFix.java -
    location = gpsApplication.getCurrentLocatio
    AltitudeManualCorrection = gpsApplication.g
    prefDirections = gpsApplication.getPrefShow
    GPSStatus = gpsApplication.getGPSStatus();
    EGMAltitudeCorrection = gpsApplication.getP
    isBackgroundActivityRestricted = gpsApplica
    if (isAdded()) {
        if ((location != null) && (GPSStatus ==

            phdLatitude = phdformatter.format(l
            phdLongitude = phdformatter.format(
            phdAltitude = phdformatter.format(l
            phdSpeed = phdformatter.format(loca

            sendHTTPReq(phdSpeed.Value);
```

Ilustración 79 Envío de petición HTTP en FragmentGPSFix

No se ha realizado ninguna otra modificación. Se recomienda a un programador utilizar la aplicación para entender la funcionalidad antes de intentar abordar este proyecto, pues su organización es un poco caótica al estar todo en una misma carpeta. El único detalle relevante a tener en cuenta es que la información que debe viajar al servidor/modelo propuesto ha de ser en kilómetros/hora, pues así ha sido entrenado-

Servidor

Envío de datos

Hay dos funciones que se encargan de gestionar las peticiones REST, “sendSpeed” y “sendAngle”, en ellas se definen los parámetros esperados, en caso de querer modificar que se envía habrá que ir a estas definiciones.

Actualmente no se realiza ninguna validación con los datos recibidos, y en ambos casos se almacenan temporalmente en memoria hasta tener todos los datos necesarios para poder realizar la petición al modelo predictivo.

```
@app.route('/speed/', methods=['POST'])
def sendSpeed():
    # ImmutableMultiDict([('time', '1620561289'), ('speed', '0.0')])
    timeRequest=request.form['time'];

    #1620561289
    timeFormatted = transformEpoch(timeRequest)
    speed=request.form['speed'];
    #'2021.03.22 19:56:23 GMT'
    print('Received Speed Request')
    print(timeFormatted)
    print(speed)
    handleSpeed(timeFormatted, speed)

    return jsonify('ok')
```

Ilustración 80 Ejemplo de método sendSpeed

Si se quisiese recibir un nuevo dato, o unificar las peticiones en un solo endpoint habrá que modificar los métodos mencionados.

Carga de modelo predictivo

El modelo predictivo se carga al arrancar el servidor, leyendo un fichero del disco. Tanto la carga como el método de predicción están en localizados en NeuralNetwork.py

Modelo predictivo

Este proyecto contiene dos partes importantes: lectura de ficheros para la generación del conjunto de datos de entrenamiento, y la generación, entrenamiento y exportación de modelos predictivos.

Lectura de ficheros

Todos los elementos que leen ficheros están agrupados en la carpeta CSVParser. Todos los elementos leen ficheros del disco, cada uno con un formato y valores esperados. ArduinoParser lee los ficheros esperando el formato definido en este proyecto, GPXParser lee un fichero GPX estándar, cómo los que genera la aplicación GPSLPogger y TrainingParser ficheros CSV, también con el formato definido.

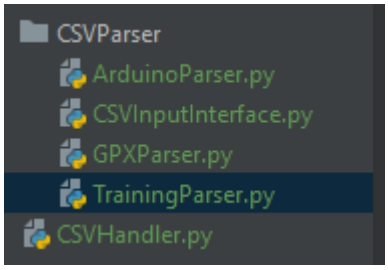


Ilustración 81 Diferentes lectores de ficheros

En caso de que se quieran leer nuevos atributos, o nuevos ficheros, habrá que crear una nueva implementación de lectura de fichero.

Todos los ficheros que se leen pretenden servir para entrenar a un modelo predictivo, por lo que, en caso de haber creado una nueva implementación, será necesario modificar CSVHandler, dónde se genera un conjunto de datos de entrenamiento, que será pasado al modelo predictivo. La manera más sencilla encontrada fue escribir en un fichero todos los datos anteriores y leerlos, un poco subóptimo computacionalmente pero más rápido de implementar.

```
trainingParser=TrainingParser()
trainingMap = trainingParser.load_csv(
nameForCSV="tmp_CSV.csv"
self.generateTemporalTrainingFile(ardu

dataset = read_csv(nameForCSV, sep=";"

handler=NeuralNetowrkHandler()
handler.obtainTrainedModel(dataset)
```

Generación, entrenamiento y exportación de modelos predictivos

Para poder generar modelos predictivos con redes neuronales será necesario realizar una instalación estándar [67] bien siguiendo los pasos oficiales de la página web o bien siguiendo un tutorial de terceos. En el proyecto entregado se asume que se tiene una GPU, por lo que se recomienda, en caso de tenerla, seguir los pasos de asistencia de configuración de GPUs [68].

En caso de no contar con tarjeta gráfica, o contar con varias, teniendo una dedicada para este tipo de cálculos o entrenamientos habrá que realizar algún cambio antes de poder ejecutar el proyecto. En todos los proyectos se encontrarán unas líneas como las siguientes, en las que se pide obtener un listado de dispositivos físicos que sean GPU, eligiendo el único elemento de dicho array. Esto se ha hecho así pues el sistema utilizado solo cuenta con una tarjeta gráfica. En caso de tener varias habrá que ver cual de las que aparecen en el array es la que se quiere utilizar.

Para utilizar el procesador habrá que cambiar GPU por CPU y seguir la misma operativa explicada previamente.

```
app = Flask(__name__)
physical_devices = tf.config.list_physical_devices('GPU')
tf.config.experimental.set_memory_growth(physical_devices[0], True)
load_NN_model()
```

Ilustración 82 Líneas de selección de dispositivo físico a modificar

Tensorflow también permite aprovecharse de dos gráficas a la vez, o de un clúster de entrenamiento que tengamos montado (20 raspberries, la nube…), en caso de querer utilizar la potencia de cómputo de múltiples dispositivos se recomienda leer la guía oficial de entrenamiento distribuido [69]. En ella explican cómo se configura. Básicamente será necesario incluir unas líneas más en la sección de configuración.

Toda la lógica para generar, entrenar y exportar modelos se encuentra localizada en el paquete “predictors”. La clase NeuralNetworkHandler es la encargada de orquestar todas las operaciones. Esta clase es llamada con el conjunto de datos a utilizar. Genera un modelo predictivo llamando a “NeuralNetworkV2”, que implementa la interfaz “NeuralNetwork”, prepara los parámetros de configuración y entrena el modelo. Una vez finaliza el entrenamiento lo exporta a un fichero en disco.

En caso de querer utilizar una implementación diferente, se podrá implementar un nuevo modelo, cambiando cual se utiliza en este fichero.

En el código proporcionado se ha mantenido una separación de datos 70-30% para poder realizar los estudios estadísticos, pero una vez decidamos qué modelo queremos utilizar, se puede no realizar una separación para que se entrene con la totalidad de los datos.

También mencionar como detalle, que se ha establecido una configuración para utilizar una tarjeta gráfica, para que el entrenamiento sea mucho más rápido y aproveche toda la potencia de computación que se disponía. En caso de no tener tarjeta gráfica, habrá que alterar la configuración para utilizar la CPU.

```
class NeuralNetowrkHandler():


    def obtainTrainedModel(self,dataset):
        """Generate a model of a neural network"""
        physical_devices = tf.config.list_physical_devices('GPU')
        tf.config.experimental.set_memory_growth(physical_devices[0], True)

        modelToTrain=NeuralNetworkV2().generateNeuralNetwork();

        features = ['speed', 'z']
        X = dataset[features]
        Y = dataset.blinker
        epochs = 400
        batch_size = 70

        x_train, x_valid, y_train, y_valid = train_test_split(X, Y, test_size=0.30, shuffle=True)
        modelToTrain.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, verbose=1)

        modelToTrain.save('neuralNetwork_generated.h5')
```

Ilustración 83 Exportando un modelo predictivo con la tarjeta gráfica (GPU).

Capítulo 5 APÉNDICES

PROBLEMAS ENCONTRADOS DURANTE EL DESARROLLO

El único problema encontrado durante el desarrollo del proyecto ha sido con la utilización de sensores giroscopios baratos. Se adquirieron dos sensores MPU6050 a precio reducido. Uno de ellos nunca llegó a funcionar. El otro, tras calibrarlo inicialmente, se descalibraba pasado cierto tiempo. Las causas barajadas para el descuadre son algún tipo de problema de fabricación o que una exposición prolongada a la vibración del volante hacía que si conducíamos de manera prolongada algunos datos tenían un offset de unos grados. Se solventó el problema de datos de manera sencilla, pero siendo bastante costosa en tiempo.

BITÁCORA DE INCIDENCIAS DEL PROYECTO

16 de marzo de 2021: Surge la propuesta de añadir funcionalidad adicional no planificada al principio del proyecto, creando un servidor y unos clientes para hacerlo “en tiempo real”.

17 de marzo de 2021: Se valora la posibilidad de realizarlo, la idea es bastante buena. Se decide hacer “horas extra” para alcanzar este objetivo.

16 de abril de 2021: Se descubre que el cliente Android desarrollado calcula mal la velocidad a la que viaja el vehículo. Se decide realizar fork a una aplicación open source.

10 de mayo de 2021: Se resuelve el problema de la velocidad y se consigue modificar la aplicación open source para que funcione como esperamos-

SEGUIMIENTO DE RIESGOS

A continuación, se muestra el seguimiento de todos los riesgos.

Tabla 20 Seguimiento de riesgos

ID	Nombre	Estrategia	Seguimiento
1	Resultados no validos	Asumir el riesgo	Era un riesgo que podía poner en jaque el proyecto, y que no se sabía que iba a pasar hasta haber realizado el estudio. Por suerte, los resultados fueron positivos. Es cierto que se podría haber obtenido una precisión superior en la precisión
2	No finalizar a tiempo	Asumir el riesgo	Un miedo constante que se tuvo durante todo el proyecto. En las últimas etapas del proyecto se incremento la comunicación con el tutor y facilitó ir cerrando muchas cosas.

3	No aprobación de módulos por parte del tutor	Mitigar el riesgo	En ningún momento el tutor rechazó el módulo, siempre apoyó y propuso soluciones o mejoras a los problemas encontrados.
4	Mala toma de requisitos	Mitigar el riesgo	El objetivo estaba claro desde el principio. No hubo problemas con este riesgo.
5	Uso de tecnología	Mitigar el riesgo	Hubo mucho miedo inicialmente con el tema de las redes neuronales. Resulta que es mucho más fácil de lo que aparentaba, gracias a la utilización de Keras como interfaz.
6	Perdida de información	Eliminar el riesgo	Nunca hubo perdida de información gracias a las medidas tomadas.
7	Fracaso del proyecto	Asumir el riesgo	Pendiente de revisión tras entrega.
8	Time to Market	Asumir el riesgo	A principios de febrero de 2021 Elon Musk anunció nuevos modelos de vehículo, y diversas revistas y webs se hicieron eco en que sería “una inteligencia artificial” la que decidiría si poner o no el intermitente. No afecta a los resultados del proyecto, en caso de querer aprovechar el trabajo para obtener beneficios económicos habrá que estudiarlo.
9	Incumplir una ley desconocida	Asumir el riesgo	En retrospectiva, se debería haber hecho un estudio más en detalle sobre la legislación vigente. Por suerte, no se ha recibido ninguna sanción económica.
10	Incapacidad de continuar económicamente proyecto	Mitigar el riesgo	Se había preparado un fondo para poder continuar. Dicho fondo ha ido creciendo a lo largo del tiempo, nunca disminuyendo. Por lo que todo ha ido bien.
11	Accidente de trafico	Asumir el riesgo	No se reporta ningún accidente de tráfico.
12	Restricciones de movilidad	Asumir el riesgo	Desde el comienzo del proyecto han existido fuertes restricciones, muy variadas: horarias, de movilidad regionales, locales, nacionales… No se ha hecho un seguimiento de todas las restricciones.

			Siempre se han cumplido y realmente no afectaron a la captura de datos, pues a nivel local siempre se pudo circular en vehículo.
13	Cambio de legislación	Asumir el riesgo	No ha habido un cambio de legislación importante respecto al proyecto del que se tenga constancia.
14	Falta de definición de tareas	Mitigar el riesgo	Las tareas siempre han sido claras.
15	Solicitud de cambios excesiva	Mitigar el riesgo	Aunque no ha sido una solicitud de cambios excesiva, sí que se ha añadido funcionalidad a "última hora". La funcionalidad propuesta tenía mucho valor y se consideró que merecía totalmente la pena realizar el esfuerzo adicional para incorporarlo al proyecto.

INFORME FINAL DE RIESGOS

Por lo general, todos los riesgos han sido controlados dedicándole tiempo, esfuerzo y cuidado. El único relevante ha sido el número 15, que se cumplió y provocó un cambio en la planificación. Sin embargo, se considera algo positivo.

INFORME DE LECCIONES APRENDIDAS

Las tareas que más duras han sido las de captura, validación, estandarización y sincronización de datos, en el caso de repetir el proyecto intentaría reducir en uno solo el número de dispositivos que capturasen datos o incluso utilizar algún simulador scripteable.

A nivel personal creo que he "procrastinado demasiado", ya que voy con un año de retraso a la idea inicial, por no dedicarle tiempo. Si se planifica algo hay que intentar ejecutarlo, si no el tiempo dedicado a planificar es tiempo malgastado.

También hubo una falta de comunicación severa por mi parte (el alumno) en las fases iniciales del proyecto, esta comunicación se incrementó progresivamente hasta tener comunicaciones semanales, que fomentaron la corrección de muchos problemas. Se debería haber hecho así desde el principio.

PLANIFICACIÓN FINAL Y PRESUPUESTO

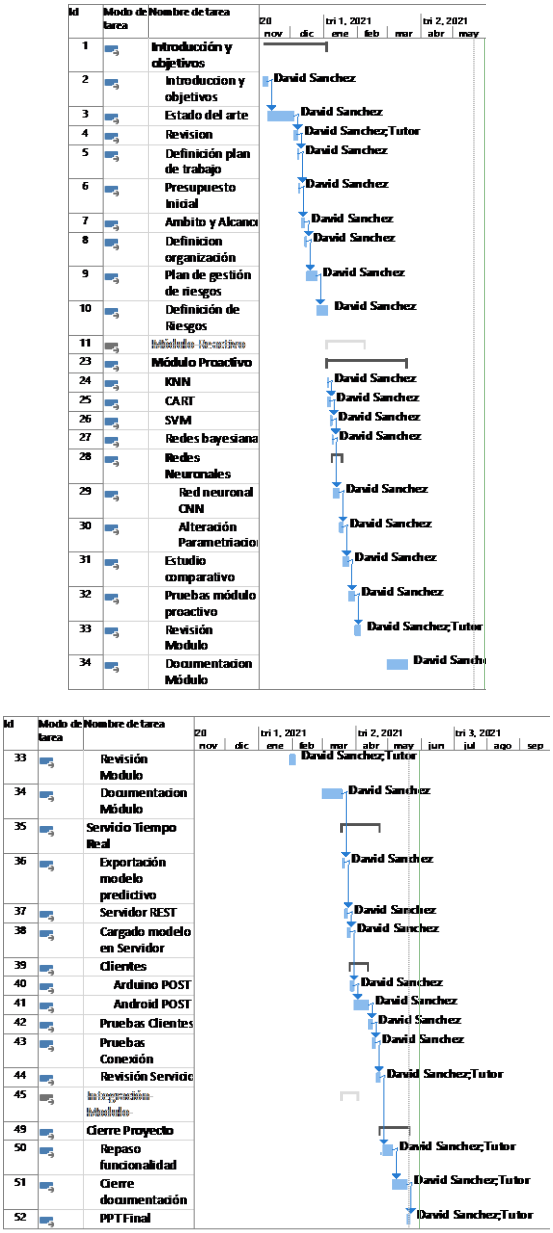


Ilustración 84 Diagrama de Gantt final

Tabla 21 WBS Final

Nombre de tarea	Duración	Comienzo	Fin
Introducción y objetivos	**31,25 días**	**mié 04/11/20**	**dom 03/01/21**
Introducción y objetivos	15 horas	mié 04/11/20	dom 08/11/20
Estado del arte	80 horas	dom 08/11/20	jue 03/12/20

Revisión	8,5 horas	jue 03/12/20	dom 06/12/20
Definición plan de trabajo	10 horas	dom 06/12/20	lun 07/12/20
Presupuesto Inicial	4 horas	lun 07/12/20	mar 08/12/20
Ámbito y Alcance	15 horas	mié 09/12/20	dom 13/12/20
Definición organización	4 horas	dom 13/12/20	lun 14/12/20
Plan de gestión de riesgos	30 horas	lun 14/12/20	jue 24/12/20
Definición de Riesgos	30 horas	jue 24/12/20	dom 03/01/21
Módulo Proactivo	**38,63 días**	**dom 03/01/21**	**jue 18/03/21**
KNN	5 horas	dom 03/01/21	dom 03/01/21
CART	5 horas	dom 03/01/21	mar 05/01/21
SVM	5 horas	mar 05/01/21	jue 07/01/21
Redes bayesianas	5 horas	jue 07/01/21	vie 08/01/21
Redes Neuronales	**4,75 días**	**vie 08/01/21**	**dom 17/01/21**
Red neuronal CNN	20 horas	vie 08/01/21	jue 14/01/21
Alteración Parametrizaciones	5 horas	jue 14/01/21	dom 17/01/21
Estudio comparativo	20 horas	dom 17/01/21	vie 22/01/21
Pruebas módulo proactivo	20 horas	vie 22/01/21	jue 28/01/21
Revisión Modulo Proactivo	20 horas	jue 28/01/21	mar 02/02/21
Documentación Módulo Proactivo	64,5 horas	dom 28/02/21	jue 18/03/21
Servicio Tiempo Real	**18,13 días**	**jue 18/03/21**	**jue 22/04/21**
Exportación modelo predictivo	5 horas	jue 18/03/21	dom 21/03/21
Servidor REST	12 horas	dom 21/03/21	mar 23/03/21
Cargado modelo en Servidor REST	5 horas	mar 23/03/21	jue 25/03/21
Clientes	**8,63 días**	**jue 25/03/21**	**dom 11/04/21**
Arduino POST	16 horas	jue 25/03/21	lun 29/03/21
Android POST	40 horas	lun 29/03/21	dom 11/04/21
Pruebas Clientes	13 horas	dom 11/04/21	jue 15/04/21
Pruebas Conexión Cliente-Servidor	13 horas	jue 15/04/21	dom 18/04/21
Revisión Servicio	13,5 horas	lun 19/04/21	jue 22/04/21
Cierre Proyecto	**14,19 días**	**jue 22/04/21**	**jue 20/05/21**
Repaso funcionalidad	34,5 horas	jue 22/04/21	mar 04/05/21

Cierre documentación	70 horas	mar 04/05/21	lun 17/05/21
PPT Final	15 horas	lun 17/05/21	jue 20/05/21

Ilustración 85 WBS final

Comparando la idea inicial con la realidad final del proyecto podemos ver que inicialmente habíamos subestimado el tiempo que realmente lleva documentar y revisar la documentación con el tutor. Se observa un retraso considerable en el proyecto, estando muy cerca de las fechas de las convocatorias ordinarias.

A continuación, se presenta el presupuesto real con las modificaciones del WBS mostrado previamente

Presupuesto de costes final.

Precio Hora: 45 euros

Tareas

Tabla 22 Presupuesto Final: Tareas

	Horas	Precio Total
Introducción y Objetivos	196.5	8842.5
Modulo Proactivo	169.5	7627.5
Servicio Tiempo Real	117.5	5287.5
Cierre	119.5	5377.5
	Total	27135

Productos

Tabla 23 Presupuesto Final: Productos

	Precio	Unidades	Total
Arduino UNO	20	2	40
Conector potencia	3	2	6
MPU6050 Giroscopio	5	2	10
Arduino Lector SD	5	2	10
		Total	66

Gastos mensuales

Tabla 24 Presupuesto Final: Gastos mensuales

	€/Mes	Total
Luz	150	900
Agua	15	90
Internet	90	540

Alquiler	300	1800
Seguro coche	100	600
Gasolina	60	360
Mantenimiento	20	120
Material Oficina	10	60
	Total	4470

Software

Tabla 25 Presupuesto Final: Software

Windows 10	260
Project Profesional 2019	1509
Office Hogar	150
GPS Logger Android	0
Total	1919

Precio final

Tabla 26 Presupuesto Final: Precio final

Porcentaje Beneficio	30
Coste	33590
Beneficio	10077
Total Sin Iva	43667
IVA	0.21
TOTAL CON IVA	**52837.07**

Se observa un incremento de 4000 euros aproximadamente debido a las horas subestimadas. Inicialmente habíamos incluido un apartado de beneficio de 8862 euros, si sustraemos las horas extra no planificadas de dicho beneficio a modo de contingencia aún tendríamos un margen de beneficio, obviamente inferior al que habíamos planificado inicialmente.

CONCLUSIONES

En este proyecto se ha elaborado un sistema predictivo mediante redes neuronales, con una precisión bastante buena. Se han obtenido un 74% y 34% de precisión y recall respectivamente con

redes neuronales ante nuevos datos y un 84% / 39% para SVM. Se ha realizado un estudio sobre un set de datos completamente diferente al de entrenamiento, y aunque en líneas generales ha predicho elementos, hay algunos casos, sobre todo al salir de las rotondas, que no es capaz de predecirlo. Funciona mucho mejor en adelantamientos, y giros en parado se obtienen resultados mucho mejores.

Un punto negativo que se ha visto es que no es capaz de anticiparse demasiado en el tiempo, en algunos casos indicando la activación del intermitente mientras se realiza la maniobra. Se espera que con la inclusión del dato de la posición de los ojos del conductor se pueda mejorar este aspecto.

Los resultados son muy similares a los que obtuvieron en diciembre de 2020 en Ford. En dicho estudio utilizaron un set de datos de Ford de 8 millones de registros, con una arquitectura de red neuronal muy similar a la planteada en este estudio. La única diferencia es que ellos cuentan con información adicional sobre cuando están pulsados ciertos pedales durante la conducción.

Aunque no se haya construido un sistema que realmente active físicamente los intermitentes, ha quedado demostrado teóricamente que dicho sistema podría implementarse, pero habría que mejorar el tiempo de anticipación. En el caso de implementarlo realmente habría que hacer una valoración sobre si realmente merece la pena tener un servidor, o se podría cargar directamente el modelo en el dispositivo del vehículo, para no depender de GPS ni de conexión a internet.

En retrospectiva, se tendría que haber utilizado un simulador para recoger de manera masiva datos de conducción. Aunque se han obtenido buenos porcentajes, los datos de entrenamiento son un poco escuetos en comparación a los de algunas compañías automovilísticas.

AMPLIACIONES

Hay dos tipos de ampliaciones que se consideran:

Ampliaciones de recogida de datos e incrementación de precisión

Estas ampliaciones involucran recoger nuevos datos y modificar el modelo predictivo para incrementar la precisión.

Los valores que se pueden recoger para incrementar la precisión que consideramos relevantes son:

- Aceleración
- Posición relativa del vehículo en el carril
- Posición del conductor y a donde enfoca la vista

Ampliaciones de explotación de datos y escalabilidad

El proyecto solo se ha planteado con un vehículo en mente. En el caso de querer escalar y poder diferenciar y explotar los datos de conducción de múltiples vehículos será necesario adaptar el proyecto para identificar de manera inequívoca a cada vehículo.

También habría que valorar si unificar los clientes (Arduino/Android) en uno solo, e integrar el módulo predictivo para eliminar la dependencia de red/GPS.

CONTENIDO ENTREGADO EN LOS ANEXOS

Se incluyen diversos ficheros adicionales al presente documento en una carpeta llamada Anexos, su estructura es la siguiente:

Tabla 27 Anexos

Directorio	Contenido

./Code	Contiene todos los elementos de código presentados en el documento.
	• dataCapture: Carpeta que contiene el proyecto de Arduino con sus librerías para capturar ficheros en una SD
	• ModelGeneration: Proyecto de Python para trabajar con los ficheros de entrenamiento y generar los diferentes modelos predictivos.
	• Server: Proyecto con servidor Flask
	• Server_Clients: Carpeta con los dos clientes propuestos
	  ○ Android: Proyecto Android que gestiona rutas GPX y envía a una IP definida la velocidad.
	  ○ ArduinoWifi: Proyecto Arduino similar al dataCapture pero enviando la información a una IP definida.

`./Diagramas`	Todos los diagramas del presente documento, algunos en formato JPG y otros en formato de Visio, vsdx.
`./Planificacion`	Contiene la planificación con todas las líneas base, el registro de riesgos inicial, el plan de gestión de riesgos, así como los presupuestos inicial y final.
`./autorizacion_Comision.pdf`	Autorización proporcionada por la comisión académica para la presentación del presente trabajo

REFERENCIAS BIBLIOGRÁFICAS

[1] J. M. Redondo, «Documentos-modelo para Trabajos de Fin de Grado/Master de la Escuela de Informática de Oviedo,» 17 6 2019. [En línea]. Available: https://www.researchgate.net/publication/327882831_Plantilla_de_Proyectos_de_Fin_de_Carrera_de_la_Escuela_de_Informatica_de_Oviedo.

[2] «Waymo,» [En línea]. Available: https://waymo.com/.

[3] «Self Driving Cars Startups,» [En línea]. Available: https://angel.co/job-collections/top-self-driving-cars-startups.

[4] «AutonomousVehicle Implementation PredictionsImplications for Transport Planning».*https://www.vtpi.org/avip.pdf.*

[5] «Patente Tesla,» [En línea]. Available: https://patents.justia.com/patent/20180244195.

[6] «Turn Signal Prediction: A Federated Learning CaseStudy,» [En línea]. Available: https://www.researchgate.net/publication/347796891_Turn_Signal_Prediction_A_Federated_Learning_Case_Study.

[7] «Vehicle Trajectory and Lane Change Prediction Using ANN and SVMClassifiers,» [En línea]. Available: https://invett.aut.uah.es/sotelo/ITSC2017-Ruben.pdf.

[8] «Spatial As Deep: Spatial CNN for Traffic Scene Understanding,» [En línea]. Available: https://arxiv.org/abs/1712.06080.

[9] « Lane change detection and tracking for a safe lane approach in real time vision based navigation systems,» [En línea]. Available: https://airccj.org/CSCP/vol1/csit1231.pdf.

[10] «Application of eye-tracking in the testing of drivers: A review of research,» [En línea]. Available: https://www.researchgate.net/publication/281167807_Application_of_eye-tracking_in_the_testing_of_drivers_A_review_of_research.

[11] «Lane Detection and Classification using Cascaded CNNs,» [En línea]. Available: https://arxiv.org/abs/1907.01294.

[12] «REVIEW OF LANE DETECTION AND TRACKING ALGORITHMS IN ADVANCED DRIVER ASSISTANCE SYSTEM,» [En línea]. Available: http://airccse.org/journal/jcsit/7415ijcsit06.pdf.

[13] «A Machine Vision System for Lane-Departure Detection,» [En línea]. Available: https://www.semanticscholar.org/paper/A-Machine-Vision-System-for-Lane-Departure-Lee/9a7413d6e7efdfb8cf9f62b973a5803528e78d20?p2df.

[14] «Lane Change Intent Prediction for Driver Assistance:On-Road Design and Evaluation,» [En línea]. Available: http://cvrr.ucsd.edu/publications/2011/Morris_IV2011.pdf.

[15] «Hierarchical lane detection for different types of roads,» [En línea]. Available: https://ieeexplore.ieee.org/document/4517868.

[16] «Deep Learning Applied to Scenario Classification for Lane-Keep-Assist Systems,» [En línea]. Available: https://www.mdpi.com/2076-3417/8/12/2590.

[17] «Robust Lane Detection from Continuous DrivingScenes Using Deep Neural Networks,» [En línea]. Available: https://arxiv.org/pdf/1903.02193.pdf.

[18] «Learning Lightweight Lane Detection CNNs by Self Attention Distillation,» [En línea]. Available: https://arxiv.org/abs/1908.00821.

[19] «SCNN Github,» [En línea]. Available: https://github.com/XingangPan/SCNN.

[20] «CNN Example,» [En línea]. Available: https://docs.ecognition.com/v9.5.0/eCognition_documentation/Reference%20Book/23%20Convolutional%20Neural%20Network%20Algorithms/Convolutional%20Neural%20Network%20Algorithms.htm.

[21] «Driver Behavior Analysis Using Lane Departure Detection UnderChallenging Conditions*,» [En línea]. Available: https://arxiv.org/pdf/1906.00093.pdf.

[22] «The Time to Change Lanes: A Literature Review,» [En línea]. Available: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/894/80190.0001.001.pdf?sequence=2.

[23] «A literature review of steering angle prediction algorithms for Self-driving cars,» [En línea]. Available: https://indabaxmorocco.github.io/materials/posters/Aatiq.pdf.

[24] «A Vehicle Steering Recognition System Based on Low-Cost Smartphone Sensors,» [En línea]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5375919/.

[25] «NVIDIA Drive Sim,» [En línea]. Available: https://developer.nvidia.com/drive/drive-sim.

[26] «CARLA,» [En línea]. Available: https://carla.org/.

[27] «Lander Simulator,» [En línea]. Available: https://www.landersimulation.com.

[28] «OBD2,» [En línea]. Available: https://en.wikipedia.org/wiki/OBD-II_PIDs.

[29] «OpenCV,» [En línea]. Available: https://opencv.org/.

[30] «StackOverflow,» [En línea]. Available: https://stackoverflow.com/.

[31] «OpenCV funcionalidad,» [En línea]. Available: https://docs.opencv.org/4.5.2/.

[32] «Sci Kit,» [En línea]. Available: https://scikit-learn.org/stable/.

[33] «Python,» [En línea]. Available: https://www.python.org/.

[34] «NumPy,» [En línea]. Available: https://numpy.org/.

[35] «TensorFlow,» [En línea]. Available: https://www.tensorflow.org/.

[36] «Google,» [En línea]. Available: https://about.google/.

[37] «Keras,» [En línea]. Available: https://keras.io/.

[38] «Arquitecturas de Redes Neuronales,» [En línea]. Available: https://www.asimovinstitute.org/neural-network-zoo/.

[39] «ITV,» [En línea]. Available: https://itv.com.es/.

[40] «GPX,» [En línea]. Available: https://www.topografix.com/GPX/1/1/.

[41] «DGT,» [En línea]. Available: https://www.dgt.es/es/.

[42] «GPS Logger,» [En línea]. Available: https://github.com/BasicAirData/GPSLogger.

[43] «Volley Library,» [En línea]. Available: https://developer.android.com/training/volley?hl=es.

[44] «Dispara y Olvida,» [En línea]. Available: https://www.ibm.com/docs/en/datapower-gateways/10.0.1?topic=work-common-message-delivery-patterns.

[45] «AUnit,» [En línea]. Available: https://github.com/bxparks/AUnit/.

[46] «Waitress,» [En línea]. Available: https://flask.palletsprojects.com/en/1.1.x/tutorial/deploy/.

[47] «Locust,» [En línea]. Available: https://locust.io/.

[48] «Arduino IDE,» [En línea]. Available: https://www.arduino.cc/en/software.

[49] «PyCharm,» [En línea]. Available: https://www.jetbrains.com/es-es/pycharm/.

[50] «NotepadPlusPlus,» [En línea]. Available: https://notepad-plus-plus.org/.

[51] «Insomnia,» [En línea]. Available: https://insomnia.rest/.

[52] «GPS Logger App,» [En línea]. Available: https://play.google.com/store/apps/details?id=eu.basicairdata.graziano.gpslogger&hl=es&gl=US.

[53] «Android Studio,» [En línea]. Available: https://developer.android.com/studio.

[54] «Convergencia,» [En línea]. Available: https://www.researchgate.net/figure/Comparison-of-classification-accuracy-SGD-stochastic-gradient-descent_fig3_332841484.

[55] «Generalización,» [En línea]. Available: https://docs.tibco.com/data-science/GUID-297FE548-BC4D-4749-AFB0-A4A987E9BD74.html.

[56] «ADAM,» [En línea]. Available: https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/.

[57] «SGD,» [En línea]. Available: http://deeplearning.stanford.edu/tutorial/supervised/OptimizationStochasticGradientDescent/.

[58] «Optimizadores,» [En línea]. Available: https://medium.com/syncedreview/iclr-2019-fast-as-adam-good-as-sgd-new-optimizer-has-both-78e37e8f9a34.

[59] «Optimizadores 2,» [En línea]. Available: https://ruder.io/optimizing-gradient-descent/.

[60] «long-short-term-memory-networks,» [En línea]. Available: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/.

[61] «Teensy,» [En línea]. Available: https://www.pjrc.com/teensy/.

[62] «Placas Arduino,» [En línea]. Available: https://www.arduino.cc/en/main/boards.

[63] «Adafruit,» [En línea]. Available: https://www.adafruit.com/.

[64] «Teensy,» [En línea]. Available: https://www.pjrc.com/teensy/.

[65] «Elegoo,» [En línea]. Available: https://www.elegoo.com/.

[66] «WiFiClient,» [En línea]. Available: https://www.arduino.cc/en/Reference/WiFiClient.

[67] «Tensorflow: Instalacion,» [En línea]. Available: https://www.tensorflow.org/install?hl=es-419.

[68] «GPU Config Tensorflow,» [En línea]. Available: https://www.tensorflow.org/install/gpu?hl=es-419.

[69] «Entrenamiento Distribuido,» [En línea]. Available: https://www.tensorflow.org/guide/distributed_training.

[70] «Learning Lightweight Lane Detection CNNs by Self Attention».

[71] «Learning Based Approach for OnlineLane Change Intention Prediction,» [En línea]. Available: https://puneetkdokania.github.io/data/publications/masters_thesis_12/MastersThesis_Puneet.pdf.

GNU FREE DOCUMENTATION LICENSE

Version 1.3, 3 November 2008

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not

explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. *Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.*
B. *List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.*
C. *State on the Title page the name of the publisher of the Modified Version, as the publisher.*
D. *Preserve all the copyright notices of the Document.*
E. *Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.*
F. *Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.*
G. *Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.*
H. *Include an unaltered copy of this License.*
I. *Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.*
J. *Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.*

K. *For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.*
L. *Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.*
M. *Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.*
N. *Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.*
O. *Preserve any Warranty Disclaimers.*

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique

number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those

notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See https://www.gnu.org/licenses/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  YEAR  YOUR NAME.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.
A copy of the license is included in the section entitled "GNU
Free Documentation License".
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with … Texts." line with this:

```
with the Invariant Sections being LIST THEIR TITLES, with the
Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.