



UNIVERSIDAD  
NACIONAL  
DE LA PLATA

## FACULTAD DE INFORMÁTICA

# TESINA DE LICENCIATURA

**TÍTULO:** Gestión de la atención en sistemas de información en vehículos (IVIS) desde el diseño temprano  
**AUTORES:** Leonel Franco Mandarinó  
**DIRECTOR:** Andrés Rodríguez  
**CODIRECTOR:**  
**ASESOR PROFESIONAL:**  
**CARRERA:** Licenciatura en Sistemas

### Resumen

*Se observa un aumento en el número de aplicaciones para dispositivos móviles que están pensadas como sistemas de información para su uso en vehículos. Al mismo tiempo, se percibe un aumento en la cantidad de accidentes viales causados por distracciones como consecuencia del uso del celular. En esta Tesina se propone un entorno metodológico y tecnológico que brinde soporte a desarrolladores de aplicaciones móviles destinadas a ejecutarse en el interior de los vehículos. Dicho entorno, toma la forma de un ambiente de simulación y recopilación de datos, orientado al prototipado rápido de sistemas de información en vehículos.*

### Palabras Clave

*Simulación; Conducción; Aplicaciones Móviles; DRT; Carga cognitiva*

### Conclusiones

*En esta tesina se ha hecho un análisis del estado del arte con respecto a los accidentes y costos viales causados por aplicaciones móviles, se analizaron las dificultades que existen a la hora del desarrollo de aplicaciones dirigidas a conductores. Finalmente se propuso una suite económica de prueba y simulación sobre la atención humana utilizando IVIS. Así se demuestra que este tipo de prototipos son una opción interesante para todo tipo de programadores que busquen mejorar la usabilidad de sus aplicaciones, protegiendo a sus usuarios.*

### Trabajos Realizados

*Se desarrolló una suite que busca simular la conducción de un auto mientras se resuelven diferentes desafíos, los cuales intentan determinar la carga cognitiva de una persona mientras utiliza una aplicación dada. Para esto, se logró implementar un mecanismo de DRT y un detector de rostros con los que se puede monitorear el comportamiento del conductor. Luego, la información producida por los componentes se almacena de forma práctica, para facilitar su posterior consumo por otras aplicaciones y generar estadísticas sobre los datos.*

### Trabajos Futuros

*Se propone la implementación de nuevas o mejores aplicaciones consumidoras de la información recopilada durante la simulación. Se podrían generar mejores gráficos y diferentes estadísticas en muchos tipos de interfaces y dispositivos. Otra forma de extender esta tesina, podría ser mediante el agregado de nuevos eventos que permitan contextualizar con mayor facilidad el estado del vehículo y la simulación en todo momento.*

Fecha de la presentación: Marzo 2022

Gestión de la atención en sistemas de información  
en vehículos (IVIS) desde el diseño temprano

Mandarino, Leonel Franco

25 de febrero de 2022

## Resumen

En los últimos años se observa un aumento en el número de aplicaciones para dispositivos móviles que están pensadas como sistemas de información para su uso en vehículos, por ejemplo Waze<sup>1</sup>, Spotify<sup>2</sup> o GasBuddy<sup>3</sup>. Aunque el diseño y desarrollo de este software puede ser abordado como un caso más de aplicaciones móviles, en la industria automotriz se utilizan conocimientos y técnicas específicas para la interacción de los conductores con el conjunto del software incluido en los vehículos. Al mismo tiempo, junto con un incremento masivo en el número de aplicaciones móviles se percibe un aumento en la cantidad de accidentes viales causados por distracciones como consecuencia del uso del celular.

Como contribución en este contexto, en esta tesina se propone un entorno metodológico y tecnológico que brinde soporte a desarrolladores de aplicaciones móviles destinadas a ejecutarse en el interior de los vehículos. Para esto, se contemplan los requisitos de seguridad y se mantiene un umbral tecnológico y económico bajo, de forma tal de proveer un fácil acceso a los desarrolladores frente a los costosos entornos físicos con vehículos reales. Dicho entorno, toma la forma de un ambiente de simulación y recopilación de datos, orientado al prototipado rápido de sistemas de información en vehículos.

---

<sup>1</sup><https://www.waze.com/>

<sup>2</sup><https://www.spotify.com/>

<sup>3</sup><https://www.gasbuddy.com/>

# Agradecimientos

A mis padres, por acompañarme en cada paso del camino. Por estar siempre presentes, en las buenas y en las malas.

A mi novia y amigos, por los recuerdos inolvidables, las risas, la infinidad de mates y pebetes de salame, pero principalmente por ser una parte tan importante en mi vida.

A Andrés y Alejandro por guiarme y aconsejarme durante el desarrollo de esta tesina. Por su enorme paciencia y dedicación en todo momento y lugar.

# Índice general

<b>1. Introducción</b>	<b>4</b>
1.1. Enfoque . . . . .	5
1.2. Organización . . . . .	6
<b>2. Motivación y trabajo relacionado</b>	<b>7</b>
2.1. Accidentes viales en Argentina y en el mundo . . . . .	7
2.2. SHRP 2 y NDS . . . . .	12
2.2.1. NDS: análisis sobre distracciones . . . . .	13
2.2.2. Análisis y resultados obtenidos . . . . .	14
2.3. Crecimiento y desafíos en la industria de IVIS . . . . .	15
2.3.1. Mecanismos de evaluación de IVIS . . . . .	16
2.4. Detection Response Tasks . . . . .	17
2.5. Creación de ambientes de simulación de bajo costo . . . . .	19
2.6. Análisis del comportamiento humano y uso de información en tiempo real . . . . .	21
2.6.1. Experimentos en simuladores de conducción . . . . .	22
2.6.2. Factores psicológicos de la información en tiempo real para la toma de decisiones . . . . .	23
2.6.3. Resultado de la experimentación . . . . .	24
2.7. Desarrollo de simuladores de conducción . . . . .	24
2.7.1. Arquitectura de simuladores de conducción . . . . .	25
2.8. Simulación del tráfico . . . . .	27
2.9. Utilización de cabinas virtuales en simuladores de bajo costo . . . . .	29
2.10. Skyline . . . . .	31
<b>3. Propuesta</b>	<b>35</b>
3.1. Estrategia general . . . . .	35
3.1.1. Objetivos . . . . .	36
3.1.2. Requerimientos a satisfacer . . . . .	36

3.2.	Diseño y desarrollo de propuesta . . . . .	39
3.2.1.	Ejecución de componentes . . . . .	40
3.3.	Componentes desarrollados . . . . .	42
3.3.1.	Plataforma de simulación . . . . .	42
3.3.2.	Detection Response Tasks . . . . .	46
3.3.3.	Reconocimiento facial . . . . .	47
3.3.4.	Servidor REST de eventos . . . . .	49
3.3.5.	Timeline chart . . . . .	49
3.3.6.	Launcher de componentes . . . . .	50
<b>4.</b>	<b>Validación</b>	<b>52</b>
4.1.	Objetivo . . . . .	52
4.2.	Definición de complejidad en interfaces . . . . .	53
4.3.	Participantes y equipamiento . . . . .	55
4.4.	Diseño del experimento . . . . .	56
4.5.	Recolección de datos y resultados . . . . .	58
4.5.1.	Encuesta posterior . . . . .	60
4.5.2.	Conclusión . . . . .	61
<b>5.</b>	<b>Conclusiones y trabajo futuro</b>	<b>62</b>
5.1.	Trabajo futuro . . . . .	63

# Capítulo 1

## Introducción

El proyecto Strategic Highway Research Program (SHRP 2)[1] obtuvo datos de video y audio de más de 3500 conductores en un período de 3 años utilizando la información recopilada por el Naturalistic Driving Study[2]. Este proyecto, logró capturar información sobre más de 35 millones de millas recorridas, que comprende 905 accidentes automovilísticos con daños a la propiedad y a los pasajeros.

Con esta información, se obtuvieron reportes que indican una prevalencia del 3.53 % de accidentes causados por el uso de aplicaciones nativas para el auto, y un 6.4 % de accidentes causados por el uso de aplicaciones para celulares. Por lo tanto, se estima que el 10 % de las veces, los accidentes se producen por conductores que están operando dispositivos electrónicos mientras manejan. También se puede notar que los accidentes son más frecuentes en el uso de aplicaciones móviles. Este porcentaje aumenta continuamente con el correr de los años, siendo que cada vez hay más autos circulando las calles con dispositivos electrónicos y pantallas que desvían la atención [3].

Por lo general, los desarrolladores de aplicaciones nativas para automóviles trabajan para grandes empresas automotrices y son quienes están formados para tener en cuenta factores tales como la visión del conductor, el tiempo que debe prestar atención a la aplicación, los elementos y colores distractivos en la interfaz, entre otros.

Sin embargo, se observa un aumento en el número de aplicaciones para dispositivos inteligentes que están pensadas como sistemas de información para uso en el vehículo<sup>1</sup>. Para las empresas no especializadas en la industria automotriz, es difícil y costoso abordar el desarrollo de pruebas en pis-

---

<sup>1</sup><https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads/>

tas especializadas y vehículos físicos, sobre todo si se quiere trabajar sobre múltiples prototipos e ideas nuevas.

## 1.1. Enfoque

Para contribuir a la solución de esta problemática se propone una suite de simulación, prueba y recopilación de datos virtual enfocada en el desarrollo ágil y seguro de aplicaciones.

La suite deberá contemplar tres áreas de interés::

**El área de simulación,** la cual requiere un software capaz de representar de la manera más cercana a la realidad la conducción de un vehículo bajo distintos factores de riesgo como puede ser el tránsito, el clima o la velocidad. Se deberá proveer información acerca de esos eventos y permitir desplegar prototipos de sistemas de información en vehículos.

**El área de sensores,** complementa el área anterior y mediante el uso de distintos sensores permite medir eventos o datos de interés sobre el conductor y sobre la simulación. Por ejemplo, el tiempo que perdió el conductor atendiendo cierto acontecimiento, su tiempo de reacción, las partes de la interfaz que llamaron más su atención, etc.

**El área informativa,** permite generar estadísticas y obtener una vista general del nivel de esfuerzo requerido por el conductor para realizar las diferentes tareas propuestas, el desempeño que tuvo e información sobre el trayecto recorrido.

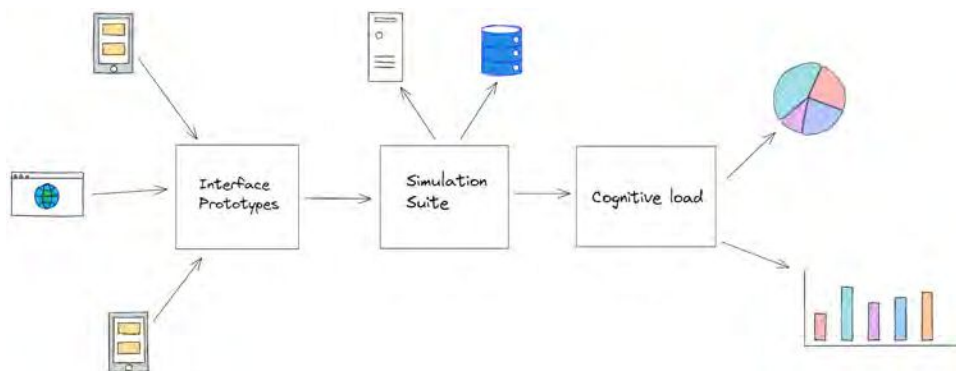


Figura 1.1: Proceso de diseño y prueba de interfaces en aplicaciones.



Como se muestra en la figura 1.1, se puede tener un análisis concreto y útil sobre la carga cognitiva de un conductor mientras utiliza una aplicación determinada.

Luego se pueden evaluar diferentes técnicas, interfaces y flujos de ejecución de forma tal que se minimice el nivel de distracción requerido por parte del conductor.

## 1.2. Organización

En los siguientes capítulos se presentan en profundidad los conceptos mencionados anteriormente. Se describe la implementación de la suite de pruebas y se abordan las problemáticas que se pueden presentar junto los resultados obtenidos luego de aplicar la solución.

Primero se dará una breve descripción de estudios, informes relacionados y sus contribuciones. Además definiremos algunos conceptos que servirán para contextualizar al lector en los diferentes mecanismos utilizados para determinar el nivel de carga cognitiva de un conductor.

Luego, en los siguientes capítulos, se discutirán las diferentes estrategias propuestas como diseño del ambiente de pruebas que utiliza al motor de simulación CARLA<sup>2</sup> como base. También se analizarán las diferentes partes que componen la suite de pruebas, evaluaremos sus diferentes funcionalidades, los posibles casos de uso y las oportunidades de extensión de las mismas.

Por último, se presentan las conclusiones del trabajo realizado y las contribuciones de la solución propuesta. La tesina cierra con posibles extensiones de trabajo futuro.

---

<sup>2</sup><https://carla.org/>

## Capítulo 2

# Motivación y trabajo relacionado

En este capítulo se profundizarán los conceptos y la información adquirida previa al desarrollo. Se dará un breve resumen de los aspectos más importantes e interesantes de la investigación realizada y la bibliografía consultada.

Existe información publicada sobre las diferentes causas de accidentes viales, tanto las más frecuentes como las que representan un mayor riesgo para la vida de los conductores. Sin embargo, son pocos los estudios que abordan la realidad específica de Argentina.

Uno de los estudios más importantes y de mayor escala es fuente de referencia para el tránsito en rutas y autopistas del mundo occidental es el denominado Strategic Highway Program 2 (SHR2) <sup>1</sup> desarrollado en Estados Unidos. La información y las conclusiones obtenidas se pueden aplicar a la mayoría de los países. Efectivamente, los sistemas de información y dispositivos electrónicos que se usan en el auto como celulares, GPS y pantallas informativas o de contenido multimedia tienen hoy un alcance global y su utilización está difundida por todo el mundo.

### 2.1. Accidentes viales en Argentina y en el mundo

Las distracciones al manejar son consideradas una de las mayores causas de accidentes de tránsito en el mundo. Según la Organización de las Naciones Unidas (ONU) cada año se pierden en el mundo aproximadamente 1.250.000

---

<sup>1</sup><http://www.trb.org/StrategicHighwayResearchProgram2SHRP2/Blank2.aspx>

vidas como consecuencia de los siniestros viales, entre 20 y 50 millones de personas sufren traumatismos no mortales, y a su vez, una proporción de éstos quedan con alguna forma de discapacidad permanente.

Como ya se sabe, la mayoría de los siniestros viales y sus consecuencias pueden ser evitables si se actúa correctamente sobre sus determinantes. Esto significa, intervenir sobre los factores que incrementan la probabilidad de ocurrencia de un siniestro, que abarcan desde la conducta humana hasta las condiciones del entorno. En este contexto, y ante la tendencia creciente de la siniestralidad vial en Argentina, se creó en el año 2008 la Agencia Nacional de Seguridad Vial (ANSV) mediante la sanción de la ley 26.363. Este organismo tiene entre sus misiones la reducción de la tasa de siniestralidad en el territorio nacional, mediante la promoción, coordinación, control y seguimiento de las políticas de seguridad vial, nacionales e internacionales.

Por su parte, en el año 2016 el Ministerio de Transporte de la Nación ha desarrollado el Plan Federal de Seguridad Vial encuadrado en el marco del Plan Federal de Movilidad Segura, cuyo objetivo es disminuir la cantidad de siniestros viales ocurridos en todo el territorio nacional mediante la aplicación de políticas y acciones que apelan a la participación conjunta de distintos actores de la sociedad.

Alineado con el Plan Federal, el Observatorio Nacional de Seguridad Vial es el organismo dentro de la ANSV que se dedica a investigar, evaluar y analizar los actos y hechos vinculados con el entramado vial, su entorno, estructura y los usuarios de las vías públicas. Como parte de sus objetivos, el Observatorio trabaja para alcanzar el registro sistemático de los siniestros viales y sus consecuencias en términos de morbilidad, procurando relevar el universo de hechos ocurridos en todo el territorio nacional con el objetivo de generar información que contribuya a la toma de decisiones.

En Argentina, según el Observatorio Nacional de Seguridad Vial, fallecen cerca de 5.500 personas al año, como consecuencia de los siniestros viales y más de 120.000 resultan con heridas de distinta gravedad [4].

Además, según datos de la Dirección de Estadísticas e Información en Salud (DEIS), del Ministerio de Salud de la Nación, las muertes por causas externas, donde se incluyen las defunciones por lesiones de tránsito, representan la cuarta causa de muerte en el país, siendo la principal en personas de 15 a 34 años de edad. En la Figura 2.1 se puede ver un gráfico que contrasta la cantidad de víctimas fatales diferenciadas por sexo y edad [4].

En marzo de 2011 la Asamblea de las Naciones Unidas lanzó la Década de Acción por la Seguridad Vial con un objetivo: colocar a la seguridad vial en la agenda de todos los gobiernos del mundo. El diagnóstico que dio marco a esta acción, determinó que de no actuar e implementar acciones que

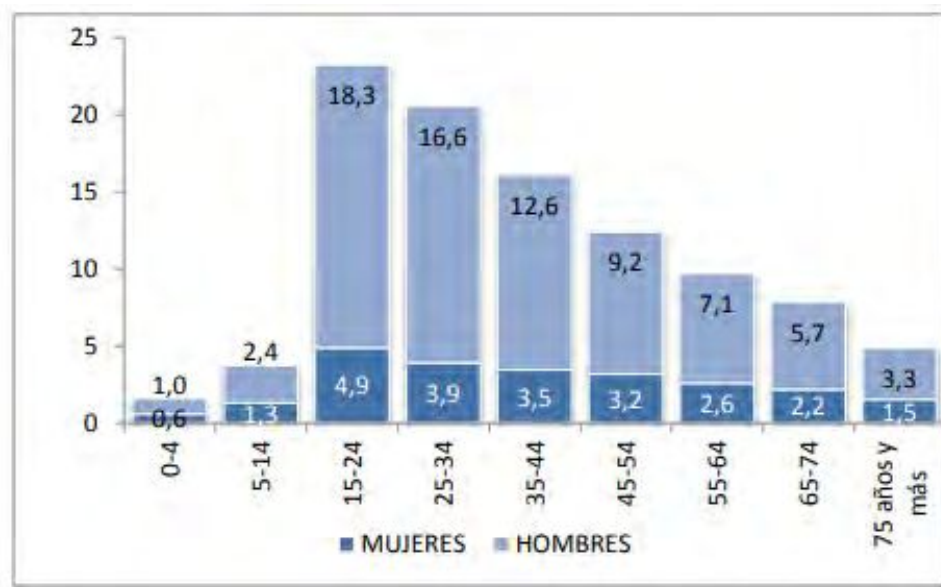


Figura 2.1: Víctimas fatales por siniestros viales según sexo y edad. En porcentaje. Total Nacional. Año 2016. Fuente: Seguridad Vial Argentina [4]

podieran dar un vuelco a las tendencias mundiales, se esperaba un aumento exponencial de las víctimas fatales producidas por el tránsito y de todas sus consecuencias e impactos negativos sobre la población.

La Década de Acción alienta a los gobiernos nacionales, provinciales o locales, y también a todos aquellos organismos involucrados en esta problemática a comprometerse con la seguridad vial y a asumirla como un área de acción prioritaria. Para ello, la Asamblea de las Naciones Unidas insiste en desarrollar acciones en torno a cinco pilares fundamentales:

- Gestión de la seguridad vial
- Vías de tránsito y movilidad más segura
- Vehículos más seguros
- Usuarios de vías de tránsito más seguros
- Respuesta tras los accidentes

Desde entonces, la seguridad vial como área de la política pública ha estado en la agenda internacional de las Naciones Unidas y de otros organismos

internacionales y multilaterales, que se sumaron a esta campaña internacional con distintos niveles de adhesión y compromiso por parte de los países. Como pudo observarse durante la revisión de medio término de la Década de Acción, que se realizó en Brasilia en noviembre del 2015, los resultados que se obtuvieron estaban lejos de cumplir con las metas esperadas. En La Conferencia de Brasilia, se declaró que los siniestros de tránsito continúan siendo un importante obstáculo para el desarrollo, un grave problema de salud pública y una de las principales causas de muerte y lesiones en todo el mundo.

Según expone la OMS en el Informe sobre la Situación Mundial de la Seguridad Vial [5], cada año se producen en todo el mundo aproximadamente 1,24 millones de muertes por accidentes de tránsito. El 80 % de las muertes tienen lugar en los países de ingresos medios, los que representan el 72 % de la población mundial con tan sólo el 52 % de los vehículos registrados en todo el mundo. Por lo expuesto, los países de ingresos medios presentan una mortalidad desproporcionada en comparación con su nivel de motorización, con mayores tasas de mortalidad por siniestros viales (20,1 cada cien mil habitantes) que el resto de los países (8,7 en países de ingresos elevados y 18,3 en los de ingresos bajos). Por otra parte, la tasa de mortalidad promedio cada cien mil habitantes para distintas regiones del mundo evidencia que las regiones con menor desarrollo humano son las que presentan tasas más elevadas. Latinoamérica es la segunda región con tasa más alta sólo por detrás de África. Esta situación sugiere la existencia de una relación, entre las condiciones socioeconómicas de una población y su índice de siniestralidad, junto con las consecuencias en términos de mortalidad.

Las muertes por lesiones de tránsito en el mundo afectan principalmente a los jóvenes entre 15 y 29 años – constituyendo la primera causa de muerte en este grupo etario, y la mitad de las víctimas fatales totales por siniestros viales corresponden a peatones (22 %), ciclistas (5 %) y motociclistas (23 %), los denominados “usuarios vulnerables de la vía pública”. Cabe resaltar que si bien la motorización ha aumentado en todos los países y ha incrementado significativamente la exposición a riesgos relacionados con el tránsito, el deterioro de la seguridad de los motociclistas es un problema particularmente grave en América Latina, debido en gran parte al exponencial aumento de la flota de vehículos motorizados de dos ruedas en esta región.

La dinámica de conducir requiere que el conductor realice varias tareas al mismo tiempo, algunas de las cuales demandan mucha atención. Sin embargo, encontramos distintos factores de distracción a la hora de conducir, como por ejemplo, comer, sintonizar la radio, utilizar el teléfono celular, mirar las publicidades o a la gente que va por la calle. Estas actividades

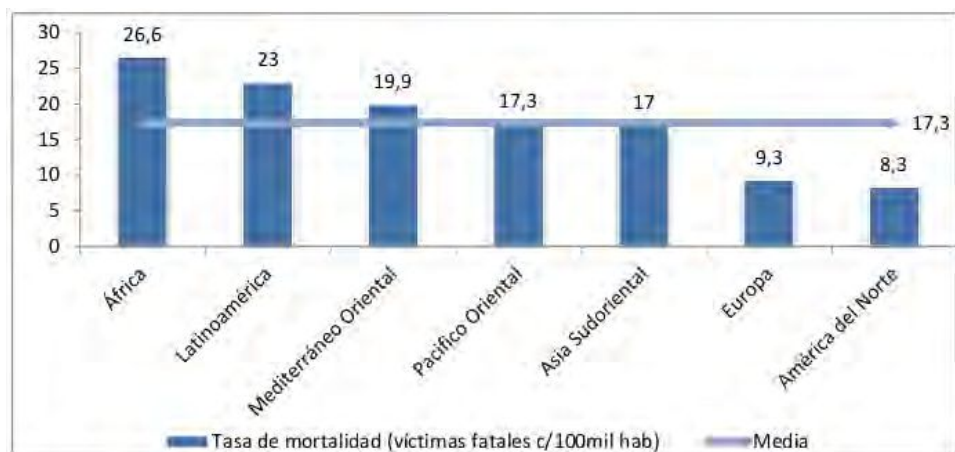


Figura 2.2: Tasa de mortalidad por siniestros viales. Regiones del mundo. Año 2013.

pueden derivar en la pérdida de atención en la conducción del vehículo o en reacciones tardías ante una alerta de peligro.

Los elementos de distracción en la conducción constituyen una preocupación creciente. El uso del celular se identifica como el elemento más frecuente de distracción entre conductores de vehículos de 4 ruedas, mientras que para los conductores de vehículos de 2 ruedas se encuentra en segundo lugar.

En Argentina durante el año 2019 se llevó a cabo el Estudio Observacional del comportamiento de conductores y ocupantes de vehículos motorizados de 4 (o más) y 2 ruedas [6]. Este estudio, tiene como objetivo determinar la tasa de uso de elementos de seguridad vial de conductores y ocupantes de vehículos motorizados, y obtener información sobre la presencia de factores de distracción entre conductores. Dentro de la categoría de vehículos de 4 o más ruedas se consiguió observar 95.782 vehículos y 138.555 ocupantes, mientras que en la categoría de 2 ruedas se observaron 38.606 vehículos y 50.518 ocupantes.

Entre las métricas obtenidas se puede ver que el 16,8 % de los conductores observados presentaron al menos un factor de distracción. Este porcentaje presenta un crecimiento significativo con respecto al anterior Estudio Observacional realizado en 2016, en el cual solo 12,1 % de los conductores presentaban algún tipo de distracción. En la figura 2.3 se puede ver que el celular es el principal factor de distracción en vehículos de 4 o más ruedas, por una amplia diferencia con el resto.

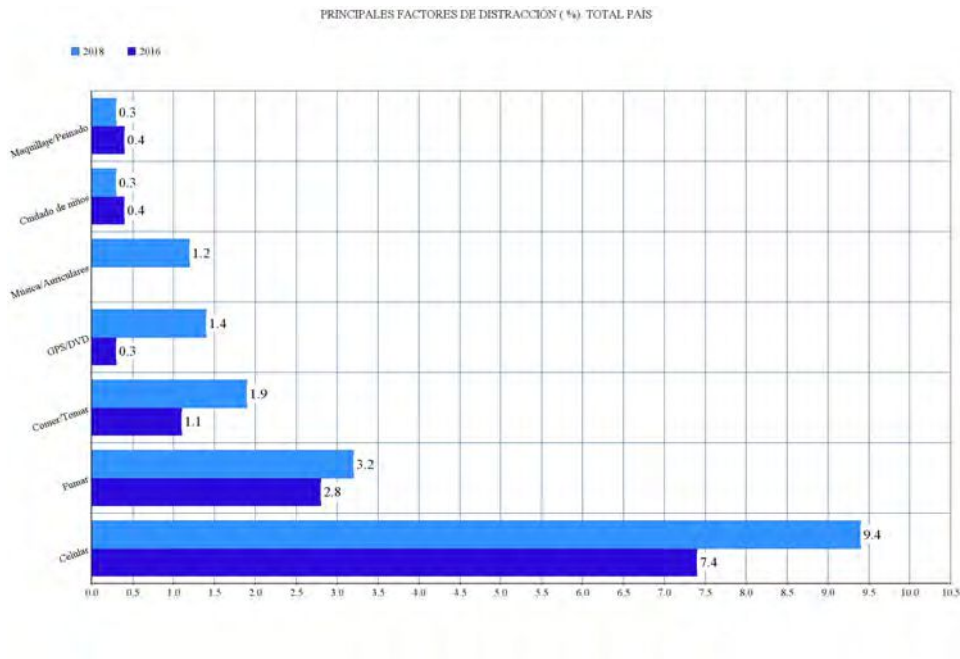


Figura 2.3: Principales factores de distracción observados en 2018 con respecto a 2016.

Según el departamento de Seguridad Vial de Argentina, una llamada que dura solo un minuto, mientras se maneja a la máxima velocidad permitida en una autopista (130 km/h), significa que el conductor recorrió más de 2,16 kilómetros sin prestar atención en el camino. De la misma forma, en la ciudad, cuando un conductor mira el celular durante 3 segundos a 40 km/h, llega a recorrer 33 metros a ciegas, casi media cuadra.

## 2.2. SHRP 2 y NDS

El Strategic Highway Research Program 2 [1] es el segundo programa realizado encargado de recopilar información sobre el tránsito y el comportamiento de los conductores. Tuvo como objetivo el desarrollo de estrategias para aumentar la seguridad en autopistas, reducir la congestión del tránsito y mejorar en la metodología para la creación de carreteras y puentes.

El SHRP2 fue aprobado por el Congreso de los Estados Unidos y tuvo

influencia sobre las siguientes cuatro áreas:

- Seguridad: Se basa en recolectar una cantidad sin precedentes de datos acerca del comportamiento de los conductores. Este análisis provee información sobre las diferentes causas de accidentes y las medidas que se pueden tomar para evitarlos.
- Renovación: Se encarga de llenar los vacíos durante la investigación y desarrollo, con el fin de producir herramientas fáciles de usar que permitan agilizar el diseño y la construcción de diferentes proyectos relacionados con la información recolectada.
- Fiabilidad: Provee las herramientas funcionales para que las agencias de transporte puedan acceder al sistema de forma analítica y organizacional.
- Capacidad: Se ocupa de identificar los aspectos claves para el planeamiento y la aprobación de los diferentes proyectos de transporte.

Nuestro interés reside en el área de Seguridad, particularmente en el rol del comportamiento humano para la seguridad vial.

El SHRP 2 inició y ejecutó el Naturalistic Driving Study (NDS), en el cual se recopiló información sobre 3.362 vehículos privados, todos conducidos por ciudadanos voluntarios que fueron observados por un período de 3 años. El proyecto NDS se basó principalmente en que se puede tener un mayor entendimiento de los problemas de seguridad al volante, mediante el estudio de los conductores, su comportamiento y los diferentes factores que los afectan, como por ejemplo, las condiciones del clima, el entorno en el que se conduce, los dispositivos electrónicos presentes en el auto, etc.

### 2.2.1. NDS: análisis sobre distracciones

Luego de la etapa de reclutamiento de voluntarios para llevar a cabo el estudio, los vehículos de los participantes fueron provistos con cámaras, radares y sensores ocultos, con el fin de capturar datos mientras estos realizaban sus actividades con normalidad. Se obtuvo información sobre 6.650.519 viajes, totalizando casi 50 millones de millas recorridas [2].

Esta información posibilitó la realización de múltiples publicaciones por parte del NDS, analizando diferentes factores que influyen sobre la conducción. Particularmente nos interesa la publicación *Analysis of Naturalistic Driving Study Data: Safer Glances, Driver Inattention, and Crash Risk* [3].



Este estudio resulta interesante, ya que, la pregunta que busca responder es **por qué y cómo** una tarea distrae y no tanto **qué** tareas distraen. Como vimos en los anteriores capítulos, sabemos que las distracciones conllevan un cierto grado de peligro y tenemos una idea aproximada de las actividades que más distraen. Pero ahora las preguntas que nos hacemos son:

- ¿Por qué manejar utilizando el celular es peligroso?
- ¿Existe alguna forma de reducir el riesgo?
- ¿Puede ser que usar el celular sea más peligroso que el resto de las distracciones por tener una interfaz más complicada?
- ¿Cuánto disminuiría el peligro si modificamos esa interfaz?

### 2.2.2. Análisis y resultados obtenidos

En general, las distracciones ocurren muy frecuentemente, y sobre todo suceden en accidentes y situaciones cercanas a ser accidentes. Las actividades demandantes visualmente son las que más riesgo conllevan, mientras que las actividades como hablar/escuchar, por ejemplo, durante llamadas del teléfono, son notablemente menos riesgosas.

Se calcula además que las distracciones visuales de 2 o más segundos, previas a la ocurrencia de un evento inesperado como la desaceleración y/o frenado de otro vehículo, son bastante más peligrosas que las distracciones de menor tiempo. Por otra parte, estos cálculos fueron contrastados con las mismas distracciones visuales en situaciones donde no ocurre un evento precipitado, mostrando un porcentaje de riesgo drásticamente bajo en comparación.

Por lo tanto, podemos concluir que los factores determinantes del riesgo son la presencia de un evento inesperado y el tiempo que mantenemos la vista fuera del camino. Esto tiene relación directa con el nivel de incertidumbre y el tiempo de reacción que dispone el conductor, esto es, una vez que recobra la concentración en el camino luego de una distracción.

La información se analizó teniendo en cuenta 3 características en conjunto, debido a que separadas generan un modelo incompleto y de poca profundidad. Para analizar las diferentes situaciones y calcular la probabilidad de riesgo de accidente se tuvo en cuenta:

- La proporción de tiempo que los ojos estaban fuera del camino, entre 5 segundos antes de un evento inesperado y 1 segundo después del evento.

- La duración media de los vistazos fuera del camino.
- Una puntuación, producto de la estimación de la inseguridad de un conductor en una situación de manejo.

Las distracciones visuales más peligrosas son aquellas en las que el conductor es expuesto al riesgo de una situación con cambios bruscos y rápidos. Esto se relaciona con la duración de la distracción, siendo que mientras más tiempo se pierde la vista del camino, más grande es la posibilidad de encontrarse en una situación difícil de afrontar.

Entre las recomendaciones para reducir el riesgo al manejar, se considera importante diseñar interfaces que minimicen la necesidad de interacción visual por parte del conductor. Otra cosa a tener en cuenta, es que la eliminación de vistazos largos (más de 2 segundos), no va a eliminar los problemas de distracción, esto se debe a que la mayoría de los accidentes son producto de pequeñas distracciones en momentos inoportunos.

### 2.3. Crecimiento y desafíos en la industria de IVIS

Se definen los *Sistemas de Información en Vehículos* como aplicaciones encargadas de procesar información sobre el vehículo y datos de diferentes fuentes, con el fin de proveer información relevante y de valor que permita tomar decisiones a conductores y/o usuarios [7]. Cuando el sistema de información se encuentra dentro del auto, se lo considera “In-Vehicle Information System” o IVIS.

Los IVIS, pueden ser aplicaciones instaladas en dispositivos móviles como celulares y tablets, o incluso pueden ser introducidos como sistemas OEM, instalados permanentemente en el vehículo desde su creación.

En el caso de los IVIS introducidos en el vehículo desde su fabricación, éstos se encuentran diseñados por compañías con la capacidad y los recursos para realizar estudios con el fin de mantener condiciones seguras de conducción durante la utilización del software[8].

Existen diferentes proyectos con el objetivo de realizar certificaciones y entregar guías de diseño para el desarrollo de IVIS, algunos de éstos son Android Auto, Apple Car Play, Mirror Link. Lamentablemente, es muy difícil para un desarrollador utilizar alguna de estas opciones, ya que, todas presentan algún tipo de conflicto o inconveniente que desmotiva los intentos de realizar un producto de calidad. En algunos casos, las guías de diseño se encuentran incompletas o no especifican los métodos de evaluación para las certificaciones. En otros, son muy restrictivos y se requieren de muchos



Figura 2.4: In-Vehicle Information System utilizando Android Auto. Fuente: Road Show by Cnet

recursos para obtener las licencias que posibilitan el desarrollo. Y para otros proyectos que no poseen los anteriores impedimentos, se encuentran discontinuados o sin soporte desde hace muchos años[8].

### 2.3.1. Mecanismos de evaluación de IVIS

Existen varios mecanismos de evaluación de IVIS. Se los pueden clasificar en cuatro tipos:

- Evaluación de la demanda visual del conductor: consiste en realizar observaciones, analizando empíricamente la atención del conductor y los elementos que observa durante la conducción. Este tipo de evaluación, requiere del uso de dispositivos adicionales, que permitan analizar el comportamiento de los ojos del conductor. A estos dispositivos se los conocen con el nombre de *Eye trackers*. En cuanto al costo de estos productos, para fines profesionales, puede alcanzar los 10.000 dólares [9].

- Evaluación cognitiva: este tipo de evaluación es útil para diseñadores y desarrolladores de software, ya que permite determinar la usabilidad de una aplicación. Por ejemplo, para presentar información de diferentes formas o para poder elegir qué actividades son menos demandantes cognitivamente. Para este tipo de evaluaciones, se consideran los Detection Response Tasks (DRT) como uno de los métodos más populares y convenientes para evaluar la carga cognitiva de los conductores.
- Evaluación analítica: en este caso la evaluación se realiza mediante modelos predictivos, que permiten calcular el potencial de distracción de diferentes elementos o interfaces, sin la necesidad de generar pruebas o prototipos. Para esta técnica de evaluación, se requiere un gran conocimiento para poder crear, aprender y entender el modelo.
- Evaluación Subjetiva: ésta es el tipo más simple de evaluación, y consiste en obtener la opinión de los usuarios de IVIS. En este caso se considera la satisfacción de los conductores, y se les pide calificar sus experiencias e interacciones con el sistema.

## 2.4. Detection Response Tasks

Las distracciones durante el manejo, suceden como el resultado de un incremento de la carga cognitiva en la persona y que no está relacionada con la actividad de conducir.

La carga cognitiva se la puede definir como el nivel de utilización de los recursos de la memoria a corto plazo, utilizada generalmente para aprender y resolver tareas o problemas [10]. Durante la conducción, estos recursos mentales son utilizados para planificar, tomar decisiones, detectar errores y modificar malos hábitos. Todas estas operaciones pueden suceder mientras se realiza el procesamiento de información de diferentes tareas visuales o manuales. Estas tareas pueden llegar a ser, el uso de dispositivos electrónicos e incluso el simple hecho de pensar en acciones irrelevantes a la conducción de un vehículo.

Si bien no se puede evitar que los conductores piensen en otras cosas mientras manejan, lo que si se puede hacer, es buscar la manera de reducir la carga cognitiva requerida para el uso de dispositivos electrónicos como celulares e In-Vehicle Information Systems (IVIS). De esta forma, se puede tener en cuenta la carga cognitiva de los conductores en las etapas iniciales de desarrollo, y probar diferentes diseños de interfaces, con el objetivo de reducir lo mínimo posible la atención requerida por parte de los conductores.

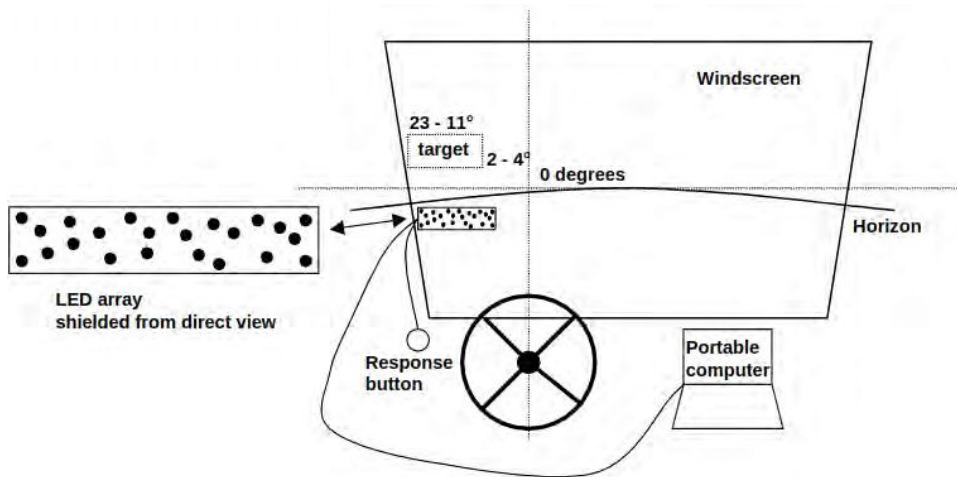


Figura 2.5: Organización de dispositivos para realizar Detection Reponse Tasks

Para poder medir la carga cognitiva de los conductores, vamos a utilizar un método conocido como Detection-Response Task [10]. De esta forma, se puede evaluar la carga mental mediante la observación del desempeño de los conductores en la resolución de tareas secundarias mientras manejan. Este método presupone que un incremento en la carga cognitiva, reduce la capacidad de atención del conductor para percibir estímulos visuales, táctiles y auditivos. Al tener una carga cognitiva alta, es más probable fallar al atender el estímulo DRT. Por ejemplo, se les podría presentar a los conductores un estímulo sensorial, como un dispositivo que vibre (táctil), una luz (visual) o un sonido (auditivo), cada 3 o 5 segundos al azar y cada vez que se percibe presionar un botón adherido a un dedo de la mano para marcar el evento.

Luego con la información recopilada, se puede calcular el tiempo de respuesta y la tasa de aciertos, con el fin de interpretar estos valores como indicadores del nivel de atención y carga cognitiva del conductor. Para poder conocer la carga impuesta por una determinada actividad secundaria, lo que se suele hacer es evaluar los puntajes obtenidos con DRT una vez, sin realizar una tarea secundaria, y luego, compararlo con los puntajes obtenidos mientras al realizarla. Mientras más grande sea la diferencia entre los tiempos de respuesta, y más baja sea la tasa de aciertos, se indicará una mayor carga cognitiva.

## 2.5. Creación de ambientes de simulación de bajo costo

Dependiendo de la etapa en la que se encuentre el desarrollo, la fidelidad de los requerimientos y el presupuesto disponible, los investigadores pueden elegir realizar estudios en diversos simuladores de conducción. Éstos pueden variar en cuanto a complejidad y calidad, pudiéndose llevar a cabo desde experiencias “In-Vehicle” en papel, hasta experiencias en simuladores capaces de generar interacciones con la aceleración y el movimiento del torque.

Por un lado, los simuladores de baja calidad permiten un desarrollo rápido y con poca inversión, posibilitando un diseño temprano. Mientras que los simuladores de conducción avanzados, requieren inversiones significativas. Sin embargo, proveen un mayor nivel de exactitud y conocimiento sobre el conductor y su rendimiento.

En el paper *Developing a Low-Cost Driving Simulator for the Evaluation of In-Vehicle Technologies* [11] escrito en 2008, se demuestra que es posible construir un simulador profesional por menos de 60.000 dólares, se lo compara con otras versiones empresariales de más de 100.000 dólares y se describen sus diferencias.

La construcción física del ambiente de simulación consistió en:

- Una computadora: compuesta por una CPU de 3.0 GHz Intel Core 2 Extreme, con una memoria RAM de 4GB y dos tarjetas gráficas Nvidia GeForce 8800 Ultra.
- Un asiento: en este caso es una silla gamer D-Box GP Pro-200 RC, que posee el mismo diseño que una silla de auto. Además, la silla se encuentra montada en tres actuadores hidráulicos, que mueven al conductor en respuesta a eventos que suceden en la simulación.
- Un volante para juegos: es un Logitech G25, que se encuentra soldado a la silla, ofreciendo muchas funcionalidades que incrementan el realismo y lo asemejan a los controles reales de un auto.
- Pantallas: El ambiente de simulación dispone de 3 pantallas con una resolución de 1024 x 768. Estas pantallas se encuentran unidas, conformando una gran pantalla con una resolución de 3072 x 768.

En cuanto al software, luego de evaluar múltiples alternativas open-source y comerciales, decidieron utilizar una opción comercial como plataforma para el simulador llamada rFactor <sup>2</sup>. Éste simulador ofrece una

---

<sup>2</sup><http://www.rfactor.net/>



Figura 2.6: Low-Cost driving simulator configuration. Fuente: [11]

experiencia de conducción convincente y realista. Si bien no posee toda la flexibilidad que se puede conseguir con un producto open-source, el juego permite un profundo grado de personalización y modificación.

Por último, como dispositivo adicional se decidió agregar un *Eye Tracker*, para esto se compró el sistema “Seeing Machines FaceLab”. Existe un gran consenso que dice que la medición de los vistazos, es crucial para determinar que tan distractiva es una interfaz en el vehículo. Los conductores distraídos tienden a reducir su comportamiento táctico y estratégico, concentrándose solo en el área inmediatamente en frente del vehículo y perdiendo visión periférica.

Las principales ventajas de este simulador frente a los más tradicionales, es el costo y el tiempo. En 2008, el precio de los simuladores rondaba en 100.000 dólares o más y usualmente no incluían *Eye Trackers*, siendo éste uno de los componentes más caros. En el caso del paper analizado, se pudo construir un simulador superior en cuanto a sensores y calidad, por menos

de 60.000 dólares y con un Eye Tracker incluido.

## 2.6. Análisis del comportamiento humano y uso de información en tiempo real

Se han producido muchos avances en los últimos años, los cuales posibilitan la capacidad de obtener información en tiempo real, sobre el tráfico y el tránsito en dispositivos personales (celulares e IVIS). Esta información, a diferencia de los sistemas comunes de información pública (carteles de tránsito, pinturas en el asfalto, etc.) tiene la ventaja de ser personalizada y mucho más específica. De ésta forma, los conductores pueden recibir datos relacionados al viaje, como la ruta de navegación, información del tráfico, el clima, colisiones, etc. Además, pueden recibir información que no está relacionada con el viaje, como llamadas, emails, diagnósticos del vehículo, entre otras.

Si bien, obtener información en tiempo real de todos los aspectos antes mencionados, se considera una gran ventaja para los conductores y su organización. La realidad, es que también posee implicaciones negativas con respecto a la percepción y al procesamiento de la información por parte de los usuarios.

Por defecto, la acción de conducir impone un alto grado de *multitasking*, la cual requiere que el conductor realice muchas funciones (acelerar, girar, frenar, revisar los espejos) mientras interactúa con su entorno. Si sumamos a todas estas acciones toda la información en tiempo real que puede resultar de una aplicación (sus opciones y notificaciones), puede resultar en una sobrecarga cognitiva del conductor. Esta sobrecarga, finalmente desemboca en un pobre rendimiento en ambas áreas, conducción y procesamiento de la información.

En el estudio *Driving Simulator Based Interactive Experiments: Understanding Driver Behavior, Cognition and Technology Uptake under Information and Communication Technologies* [12], a diferencia de muchos otros que solo se enfocan en los beneficios de la información en tiempo real para la toma de decisiones, se busca exponer la importancia de los factores humanos, sus limitaciones y los efectos psicológicos en conductores durante la toma de decisiones.



### 2.6.1. Experimentos en simuladores de conducción

El simulador, fue desarrollado en el laboratorio de Nextrans Center en Purdue University. Consiste de un panel principal, un volante, una silla ergonómica de auto, tres pantallas grandes, entre otras cosas.



Figura 2.7: Simulador de conducción interactivo, desarrollado en Nextrans Center en la universidad Purdue. Fuente [12]

Además, en este estudio se incluyen dos experimentos basados en el uso de simuladores de conducción interactivos, con el fin de analizar los efectos cognitivos inducidos por información en tiempo real durante la conducción.

Estos experimentos, consisten en transitar de forma realista, una red de rutas construidas en base al mapa del norte de Indianapolis, Indiana. Durante la experimentación se pueden controlar diferentes factores, alguno de estos son:

- Nivel de congestión
- Accidentes en el camino
- Origen, cantidad, contenido y modalidad sensorial de los sistemas de información

- La ruta indicada desde el origen hasta el destino

Durante el trayecto, se espera que el conductor tome diferentes decisiones relacionadas con el camino preseleccionado. De esta forma, se logra un entorno real, en el que el conductor deberá aprovecharse de los sistemas de información en tiempo real, para poder tomar las decisiones necesarias que lo lleven al destino.

Por último, se utilizan diferentes sensores biológicos para obtener información sobre la carga cognitiva del conductor durante la experimentación. Luego de un experimento de práctica con el simulador, los conductores son equipados con dispositivos que permiten realizar electroencefalografías y electrocardiografías. Estos sensores son recalibrados regularmente para obtener una buena calidad en la información obtenida. Los conductores, también son equipados con un Eye Tracker, el cual resulta indispensable para obtener datos más concretos sobre el accionar y las distracciones del usuario.

### **2.6.2. Factores psicológicos de la información en tiempo real para la toma de decisiones**

Para este estudio, se consideraron cuatro facetas de la percepción de la información, con el fin de caracterizar los efectos psicológicos en usuarios de la información provista en tiempo real, estos son:

- **Facilidad de comprensión:** implica la percepción de la información en términos de complejidad cognitiva (que tan clara es la información presentada) y carga cognitiva (la cantidad de información).
- **Suficiencia:** está relacionada con la satisfactibilidad de las necesidades informativas del conductor para la toma de decisiones.
- **Consistencia:** representa la información que se percibe consistente entre la información provista en viajes previos o la información de múltiples fuentes.
- **Favorabilidad:** está dada por la relevancia de la información en un momento dado en correspondencia con los eventos que suceden en la simulación.

Luego, basándose en esta información y algunos aspectos individuales del conductor, se pueden observar y analizar, tres efectos psicológicos de la información en tiempo real en los conductores. Estos efectos son:

- Agobio cognitivo: Se relaciona con la cantidad de esfuerzo mental que se requiere, para el procesamiento de la información relacionada con la conducción en el entorno.
- Decisión cognitiva: Se refiere al nivel de conciencia, para comprender el estado del viaje y el nivel de incertidumbre a la hora de tomar decisiones, basado en la información provista.
- Alivio emocional: Es el nivel de alivio mental dado por la anticipación de resultados futuros, basado en la información provista.

### 2.6.3. Resultado de la experimentación

Luego de realizar las experimentaciones propuestas sobre más de 200 participantes adultos, y obteniendo entre 3 y 5 muestras de cada uno, se obtuvieron interesantes resultados.

El rol de los efectos psicológicos dados por la información para la toma de decisiones al volante, ha demostrado ser estadísticamente significativo. El *Agobio cognitivo*, implica que si todas las otras condiciones son iguales, los viajantes con mayor carga cognitiva son más propensos a quedarse en el mismo camino. Esto significa, que el estrés generado por la sobrecarga de información, puede debilitar la influencia o la efectividad de la información para alterar el camino elegido por un conductor. Por contraste, la *Decisión cognitiva* tiene un impacto positivo en el cambio de rutas; si el conductor posee más claridad con respecto al ambiente de tráfico en la ruta actual y la ruta alternativa, es más probable que cambie el camino si la alternativa tiene menos tráfico. El nivel de *Alivio emocional* también tiene una influencia positiva, relacionada a la toma de decisiones y el cambio de rutas prefijadas.

Incorporar los efectos psicológicos de la información en tiempo real, puede mejorar el entendimiento de los conductores y las decisiones que toman para elegir sus rutas. El resultado de los estudios, puede proveer a operadores de sistemas los datos suficientes para poder desarrollar estrategias efectivas. Así, se pueden crear sistemas e interfaces que tengan en consideración el comportamiento de los conductores, y de esta forma se puedan mejorar otros sistemas, por ejemplo reduciendo el congestionamiento.

## 2.7. Desarrollo de simuladores de conducción

La mejora continua en el rendimiento del hardware es un hecho bien conocido. Esto permite el desarrollo de simuladores de conducción más complejos. La inmersión de los conductores en la simulación, se ve incrementada

por el alto nivel de fidelidad en la retroalimentación recibida. Es decir, un mejor sistema visual, con mayor tasa de refrescos, un entorno realista y un buen modelado 3D de los elementos en la simulación, posibilitan un gran grado de inmersión [13].

En este sentido, la aplicación de nuevas técnicas y la necesidad de proveer más realismo, produjeron nuevos requerimientos para el rendimiento de los simuladores. Así, se estimulan diferentes sentidos de los conductores, como el visual, el auditivo, el táctil y el sistema vestibular. La silla, que conforma el sistema de movimiento y vibración, usualmente consiste en seis grados de movilidad y está dado por una plataforma que reproduce las sensaciones de aceleración y velocidades angulares.

Si bien experimentos de conducción en la vida real proveen ambientes mucho más realistas, los simuladores son necesarios para reducir los costos y riesgos de los experimentos. Además, en ambientes reales, los experimentos pueden sufrir interferencias o resultan muy difíciles de replicar. En cambio, en los simuladores, se provee de un “mundo” completamente controlado, en el que las anteriores desventajas no existen.

Otra funcionalidad importante de los simuladores, que no está relacionada con el área de investigación, tiene que ver con la capacidad de ser utilizados como entrenamiento para conductores novatos y experimentados. Según el proyecto GADGET, el entrenamiento de conducción está dividido en cuatro niveles: maniobras, situaciones de tráfico, metas de contexto y habilidades [14]. Pero en la práctica, se utilizan los simuladores solamente para cubrir los primeros dos niveles. Esto se debe a las limitaciones actuales de los simuladores [15].

### **2.7.1. Arquitectura de simuladores de conducción**

Existen muchas configuraciones y enfoques a la hora de usar simuladores de conducción, pero en general presentan la misma estructura básica.

La estación de control del simulador es la interfaz principal de la aplicación. Desde este lugar se pueden seleccionar los diferentes experimentos o ejercicios a ejecutar. Usualmente, se provee un modo de edición para la creación de nuevos experimentos, vehículos, condiciones de clima, etc. Estas sesiones simuladas se pueden guardar para su posterior evaluación y análisis.

El sistema visual, como su nombre lo indica, se encarga del procesamiento visual de la simulación. Abarcando efectos de iluminación, el clima y los elementos con movimiento en la escena representada. Ésta es una de las áreas más demandantes de la simulación en cuanto a procesamiento, y usualmente se necesita de una *Graphic Processing Unit* (GPU) para tener

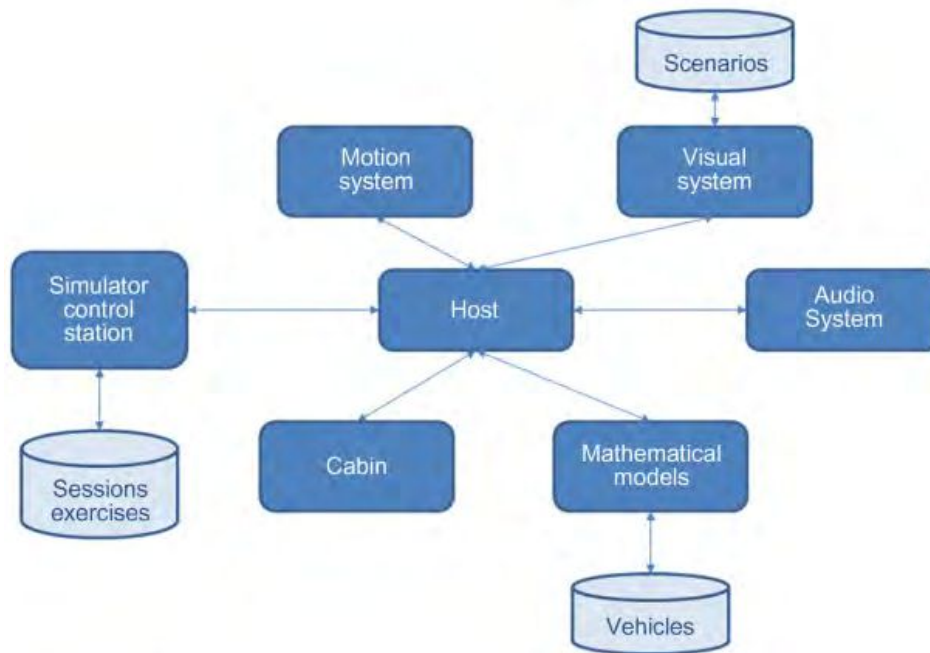


Figura 2.8: Arquitectura básica de un simulador. Fuente: [13]

un buen rendimiento durante la conducción.

Los elementos presentes en el entorno de la simulación, son muy importantes para lograr un alto grado de inmersión. Se pueden simular muchos accidentes y situaciones en base a otros vehículos, peatones, animales, etc.

Los modelos matemáticos, conforman un módulo clave para el rendimiento del simulador. Afectan directamente el alcance del grado de realismo de una simulación. Estos modelos, se utilizan para representar todas las físicas relacionadas con la interacción entre los diferentes elementos de la simulación. Por ejemplo, las ruedas y el asfalto, el motor y la transmisión, el efecto de suspensión, etc.

No es tan habitual que los simuladores vengan equipados con un sistema de movimiento. Este sistema, intenta brindar información sobre el movimiento del auto, al conductor. La idea es transmitir la sensación de aceleración lineal y de velocidades angulares al conductor a través de la silla y los dispositivos que utiliza.

También es importante tener una cabina real (o como mínimo controles reales), consiguiendo así, eliminar la barrera inicial que separa la simulación

de la realidad y lograr que el conductor confíe en la simulación.

Por último, y no menos importante es el sistema de sonido. El sonido, es otra pieza clave en la simulación, que ayuda a la inmersión del conductor y a la conducción general del vehículo. Mucha información se obtiene del sonido del motor, es importante percibir el cambio de velocidad en el motor. Incluso, algunos desperfectos en el auto se pueden percibir con el sonido.

## 2.8. Simulación del tráfico

La simulación del tráfico ha sido un campo muy activo en el área de simulación desde hace un tiempo. Por tráfico, en este contexto, se entiende como una palabra colectiva, que se refiere a un grupo de medios de transporte operando simultáneamente y definidos dentro de un área geográfica. Ésta puede ser una ciudad, una metrópolis, un grupo de ciudades interconectadas, regiones, etc.



Figura 2.9: Representación del modelo del tráfico en la ciudad de Londres. Fuente [16]

Depende de la aplicación, restringir el rango de medios y modos de transporte dentro de las diferentes áreas geográficas. Por ejemplo, se podría limitar solo al uso de vehículos terrestres, bicicletas o peatones. Como es de esperar, mientras más modos y medios de transporte son incluidos, mayor será la complejidad y el procesamiento requerido de la aplicación.

En cuanto a la parte de simulación, lo definimos como la recreación de un fenómeno del mundo real, a través de la instanciación de un modelo matemático definido previamente. La creación de modelos, es una de las partes más difíciles del acto de simulación. Los modelos se deben definir con antelación, incluyendo todos los elementos, conceptos y variables necesarios. Además, los modelos deben pasar por una serie de pruebas y validaciones, para saber si son lo suficientemente precisos para utilizarlos en la aplicación.

Hay una frase famosa de George Box, que traducida dice “Todos los modelos están mal, pero algunos son útiles”. Un modelo es una representación formal de un fenómeno natural. En principio, todos los fenómenos serán siempre muy complejos para ser representados completamente con exactitud.



Figura 2.10: Modelado de vehículos, peatones y elementos de una ciudad. Fuente [16]

Sin embargo, no es necesario tener una precisión absoluta en los modelos. Siempre existirá un balance entre la precisión y el rendimiento. Muchas veces, es necesario que los modelos sean rápidos, por ejemplo para su uso en aplicaciones de tiempo real. En otras ocasiones, podría no importar la velocidad del modelo, ya que se requiere que sea lo más realista posible.

Todo lo dicho anteriormente puede ser extendido para el tráfico. Cualquier situación vehicular puede ser un proceso muy complejo y aleatorio,

especialmente debido a la intervención humana. Definitivamente es necesaria la utilización de modelos, para describir y analizar los procesos que transcurren, y además para predecir futuros resultados.

Los mecanismos para modelar el tráfico han evolucionado a lo largo del tiempo. En sus inicios, el tráfico fue modelado basándose en físicas y dinámicas de fluidos; el tráfico era interpretado como un todo continuo. A este tipo de modelado se lo conoce como modelado del tráfico *macroscópico*.

Este tipo de modelado fue bastante eficiente para muchos tipos de aplicaciones y tareas, pero luego surgió un enfoque distinto. En este nuevo enfoque, los vehículos eran modelados individualmente, todos con un nivel de detalle similar, y más importante, las interacciones entre los mismos también fue modelada. Este tipo de modelado del tráfico se lo conoce como *microscópico*.

Por un tiempo ambos enfoques filosóficos para el modelado coexistieron, pero luego surgieron técnicas híbridas de modelado. Técnicas que incorporaron las ventajas de ambos enfoques y dejaron de lado sus desventajas. Así, se crearon y aplicaron los modelos *mesoscópicos*.

## 2.9. Utilización de cabinas virtuales en simuladores de bajo costo

En el estudio *Steering Control in a Low-Cost Driving Simulator: A Case for the Role of Virtual Vehicle Cab* [17], se realiza un análisis respecto a la credibilidad de los resultados obtenidos con simuladores de bajo costo, a partir de la falta de una cabina para el conductor.

Dependiendo de la inversión inicial para la construcción de un ambiente de simulación, se los puede clasificar en tres categorías generales:

- Bajo costo: poseen un volante para juegos, con pedales, varias pantallas y sin cabina de conducción.
- Medio costo: Una proyección grande, una base fija, una cabina de conducción parcial o completa, etc.
- Alto costo: Visión de 360° del simulador, cabina de un vehículo real montada en sensores de movimiento, etc.

La credibilidad de los resultados obtenidos con simuladores de conducción, en general han sido causa de preocupación y debate en la comunidad científica por años [18], [19]. Algunos estudios han podido evidenciar que las



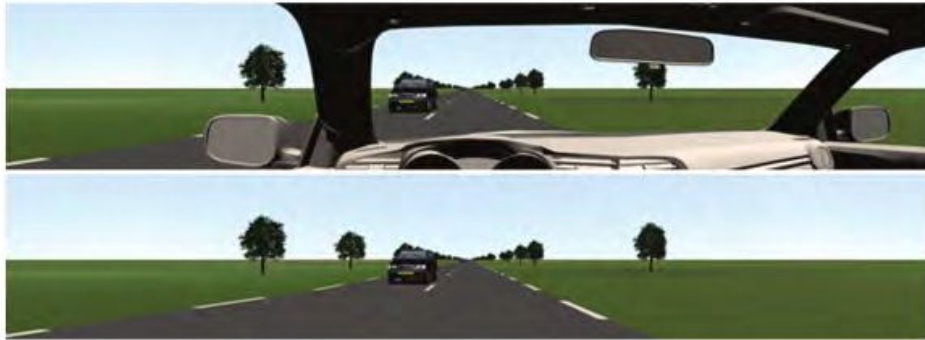


Figura 2.11: Vista del conductor y el tráfico con cabina y sin cabina en el simulador. Fuente [17]

diferencias entre los simuladores y los datos obtenidos del mundo real, son del mismo orden y la misma dirección [20], [21].

Otros estudios se encargaron de comparar los simuladores de bajo costo, con otros más costosos y complejos, y así poder determinar si los primeros son lo suficientemente válidos para tenerlos en cuenta [22]. Los resultados indican que los simuladores de bajo costo, son buenos indicadores para el estudio de los efectos de IVIS en el comportamiento de los conductores.

Un factor que nunca había sido estudiado, son los efectos en la conducción como resultado de tener o no cabina en la simulación del auto. Durante la conducción, las personas usualmente tienen una respuesta correctiva ante un error, más que una respuesta preventiva. Esto significa por ejemplo, que los conductores toman acciones para corregir la dirección del vehículo una vez que se perciben cerca de uno de los bordes del camino, y no antes.

La pregunta que surge en relación a la falta de cabina en la simulación, es si los conductores logran percibir de la misma forma, los límites y el tamaño del auto. Para responder esta pregunta, se realizaron diferentes experimentos con varias configuraciones. Además, para poder contrastar los resultados se desarrolló una cabina virtual, a diferencia de una física, con el objetivo de mantener la idea de “simuladores de bajo costo”.

Los resultados obtenidos luego de la ejecución de los experimentos, indicaron que si bien, la falta de cabina no afecta directamente a la velocidad percibida por los conductores. Sí afecta en gran medida, a la percepción del posicionamiento en el carril. Se puede observar, que los conductores se desvían del centro del carril con mucha facilidad cuando no poseen una ca-



Figura 2.12: Configuración virtual del simulador con cabina y sin cabina.  
Fuente [17]

bina, sobre todo en las curvas.

Esto sugiere, que debido a la falta de información visual sobre el ancho del auto. Los conductores, que se encuentran ubicados en el extremo izquierdo del auto, son propensos a desviarse del camino en curvas hacia la dirección opuesta en la que se encuentran, en este caso a la derecha.

Por esta razón, se sugiere agregar cabinas en simuladores de bajo costo que requieran un mayor nivel de precisión por parte del conductor, a la hora de moverse por los carriles. Esto generará información más confiable y precisa con una cantidad de presupuesto similar.

## 2.10. Skyline

Por último se destacará un proyecto que sirvió como inspiración para esta tesina. Skyline: a Rapid Prototyping Driving Simulator for User Experience [23], es un paper en el cual se describe una plataforma de prototipado para el desarrollo de experiencias de usuario, basado en la simulación de

conducción. Fue desarrollado en Intel Labs, siguiendo principios de flexibilidad, integración y personalización, para permitir el prototipado rápido de experiencias de usuario en el vehículo.

El proceso de desarrollo que impone el diseño como prioridad, se ha ganado el reconocimiento de muchas industrias. Siendo un proceso probado, que es capaz de resistir problemas complejos desde el punto de vista del usuario. Así, las soluciones identificadas son prácticas e interesantes.

La ventaja más importante de un taller de diseño, es la habilidad de construir sobre conceptos identificados durante la sesión de diseño realizada con especialistas. De esta forma, se puede determinar qué elementos son los más importantes para realizar investigación. Los conceptos son estructurados en rangos en base a su prioridad, y los más convincentes son divididos en casos de uso.



Figura 2.13: Configuración física de Skyline. Fuente [23]

Debido a los requerimientos creados como resultado de éste proceso, se necesita de un simulador de conducción que opere como una plataforma de desarrollo. Para ésto, se necesita un cierto equipamiento conformado por:

- Una televisión de 50 pulgadas
- Un volante Logitech G27 con pedales

- Un panel de vidrio polarizado, que refleje el contenido de una tablet a modo de tablero electrónico de auto
- Un asiento de cuero para el conductor y el pasajero

En cuanto al software, Skyline es un entorno que se ejecuta en tiempo real. Utiliza tecnologías web open source para conectar cada aplicación que se ejecuta en los componentes. La computadora principal ejecuta un programa compilado de un mundo 3D, creado con *Unity gaming framework*. Node.js provee un entorno multiplataforma, en el cual se utiliza un paradigma de eventos para la comunicación entre los diferentes aplicativos Web.

En segundo plano, se ejecuta un bus que permite el intercambio de mensajes entre el simulador de conducción, y los diferentes dispositivos conectados. A este bus se lo denomina *Message Queue Telemetry Transport* (MQTT). En cada uno de los dispositivos, se deberá ejecutar un navegador Web en modo de pantalla completa, el cual se conecta por Wi-Fi a la red de Skyline.

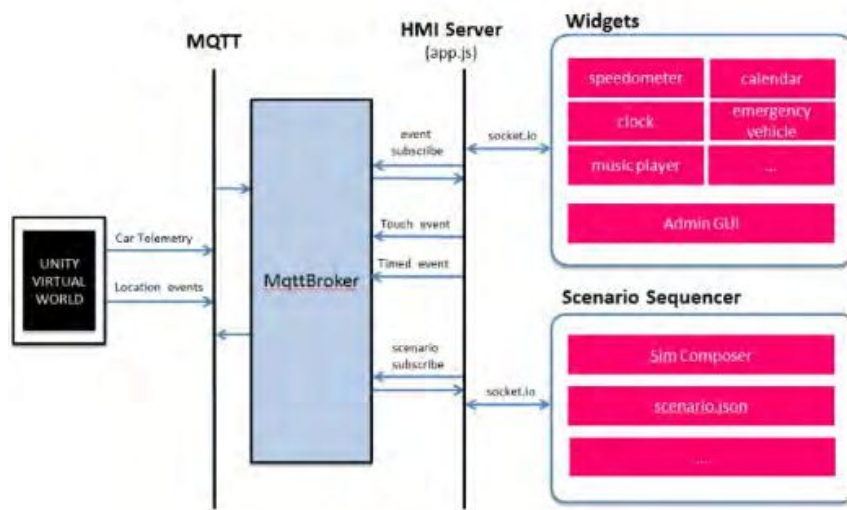


Figura 2.14: Configuración del Back-End en Skyline y utilización del MQTT. Fuente [23]

MQTT es un protocolo de transferencia de mensajes open source, diseñado para conectar ubicaciones remotas con muy poco procesamiento y

bajo uso del ancho de banda. El patrón de publicación-suscripción de mensajes, requiere de un agente encargado de distribuir los mensajes a los clientes interesados, basandose en el tipo de mensaje. Es un mecanismo muy popular en servicios de conexión de “Machine to Machine” (M2M) e Internet de las cosas (IOT).

Adicionalmente, las experiencias de usuario utilizadas en Skyline, son administradas por un *Secuenciador de Escenarios*. Éste, se encarga de controlar las actualizaciones de las interfaces de los dispositivos, y muchos de los eventos que se disparan durante la ejecución de la simulación.

La principal diferencia que se puede destacar entre Skyline y esta tesis, es el ámbito de aplicación y el enfoque del ambiente de simulación. En nuestro caso, se disponen de sensores de reconocimiento facial, y se implementaron mecanismos para medir la carga cognitiva del conductor. Así, con esta información se puede obtener un perfil más detallado del conductor y su comportamiento, con el fin de desarrollar interfaces más seguras y orientadas a su contexto de utilización.

## Capítulo 3

# Propuesta

Con la finalidad de implementar una suite de prueba y simulación, capaz de recopilar información sobre el conductor, su nivel de carga cognitiva e información general sobre las causas de distracción, se desarrollaron una serie de componentes que trabajan en conjunto, brindando un sistema modular y extensible. A continuación, se detalla la estrategia general de la propuesta y posteriormente, el diseño y desarrollo específico de la suite.

### 3.1. Estrategia general

El proceso de desarrollo de software para la creación de IVIS incluye cuatro grandes bloques de tareas: diseño de aplicaciones, implementación, diseño de experimentos, y evaluación, como se esquematiza en la Figura 3.1.

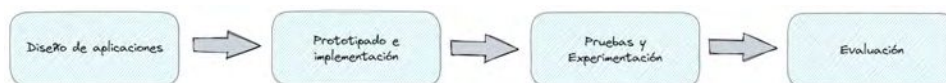


Figura 3.1: Proceso de desarrollo, prueba y evaluación de un IVIS.

El alcance de esta tesina busca cubrir o facilitar los últimos dos bloques del proceso de desarrollo. De esta forma, nos enfocamos en la generación de pruebas y diseño de experimentos desde el desarrollo temprano y su posterior evaluación.

En esta sección presentamos los objetivos, los requerimientos para el desarrollo de esta tesina y las situaciones en las que se puede maximizar los beneficios de la suite de simulación.

### **3.1.1. Objetivos**

Con la implementación de la suite de simulación de IVIS se espera obtener un ambiente de prueba económico y de fácil alcance que permita la adopción de este tipo de sistemas para una mayor cantidad de desarrolladores de aplicaciones orientadas a vehículos y medios de transporte. De esta forma, contribuimos al diseño de productos y prototipos de mayor calidad y seguridad para los usuarios finales.

Para esto, en esta tesina se propone un conjunto de componentes, recomendaciones e información; que facilite la creación de un ambiente de simulación competente y extensible.

### **3.1.2. Requerimientos a satisfacer**

Dentro de los requerimientos necesarios para poder llevar a cabo la suite de prueba y experimentación se encuentran los siguientes.

#### **Requerimientos de uso y resultados esperados**

La plataforma debería satisfacer los siguientes requerimientos mínimos expresados en formato de historias de usuario.

- Como usuario de la suite, quiero utilizar el ambiente de simulación de forma individual o acompañado por una persona encargada, para configurar y administrar los experimentos realizados.
- Como usuario individual de la suite, quiero ser el encargado de configurar la simulación y conducir el auto dentro de la simulación, para poder analizar los resultados de mi propia experimentación (Este caso de uso está enfocado principalmente para desarrolladores que se encuentran construyendo interfaces y prototipos, y deseen probar diferentes combinaciones de forma rápida y sencilla. El desarrollador deberá establecer una serie de experimentos y tareas, con una duración determinada y luego comparar los resultados obtenidos con previas ejecuciones).
- Como administrador de la suite, quiero realizar pruebas complejas de duración prolongada con un gran grupo de personas de diferentes edades y contexturas para reunir varios experimentos que intenten explorar diferentes escenarios, factores de distracción y luego compararlos con respecto a las diferencias en el grupo de participantes.

- Como analista de experimentación, quiero tener una base de datos cargada con todos los eventos sucedidos durante las diferentes experimentaciones para poder realizar un análisis integro de los datos y generar conclusiones sobre la carga cognitiva que mantuvo el conductor durante la experimentación, las situaciones que generaron un aumento en la tasa de distracciones, el nivel de inmersión del usuario en la simulación, etc. (Los eventos se deberán encontrar clasificados por las experimentaciones en las que se producen y estarán compuestos por información sobre, el tipo, la fecha y hora en la que sucede, y el estado del evento; siendo este último el campo que contiene información textual sobre el evento sucedido)
- Como analista de experimentación, quiero poder sacar conclusiones con respecto a los datos para determinar si una interfaz o aplicación es más segura y práctica que otra para la utilización en un ambiente de conducción.

## **Requerimientos de Hardware**

En cuanto al Hardware de soporte, se necesita mantener en un mínimo el presupuesto requerido para la construcción de la suite de forma que sea accesible su replicación para la mayor cantidad de desarrolladores posible.

Así, se deberá poder ejecutar la suite con una computadora de propósito general, acompañada de unos pocos periféricos de mayor o menor calidad, que posibiliten la ejecución de todas las funcionalidades implementadas en la suite y obtener buenos resultados.

Sin embargo, la suite debe ser lo suficientemente flexible y contemplar la posibilidad de que algunos usuarios realicen mayores inversiones, comprando dispositivos y periféricos especializados, con el fin de incrementar la inmersión en la simulación y tener un mejor análisis del conductor.

Siendo que en las simulaciones se deben controlar un auto y la suite se beneficia de la inmersión del conductor, se debe proveer soporte para varios tipos de mandos y controles. Es especialmente importante que se pueda controlar el auto en la simulación mediante la utilización de un volante y pedales, conectados a la computadora como periféricos.

También se debe contemplar la posibilidad de ejecutar la suite en uno o más monitores, teniendo la capacidad de redimensionar la ventana del simulador o ejecutarlo en modo pantalla completa, abarcando todas las pantallas.



## Requerimientos de software

En cuanto al software, todas las herramientas y componentes utilizados deben ser preferentemente sin licencia y de código abierto, lo cual reduce el costo respecto de sistemas propietarios de manera drástica.

Se requiere que la suite esté compuesta por un simulador de conducción, en el que se puedan controlar diferentes tipos de vehículos, siendo posible seleccionar entre diferentes ciudades y terrenos. El simulador debe disponer de calles con semáforos, peatones y otros automóviles; con la finalidad de añadir dificultad y variabilidad a las ejecuciones de los experimentos.

Además del simulador, se debe disponer de elementos que permitan estudiar y analizar al sujeto de prueba que ejecuta el experimento. Para esto, se necesita explorar diferentes mecanismos para medir la carga cognitiva del conductor, sus reacciones y reflejos.

Particularmente resulta de gran interés la implementación del mecanismo DRT para obtener información sobre la carga mental del conductor. Esto se puede combinar con otras formas de identificación, como lo son Eye Trackers o similares, obteniendo más información sobre el estado de la visión del conductor mientras realiza los experimentos

Se necesita que los experimentos a realizar permitan medir el tiempo de respuesta de un conductor en diferentes situaciones, se debe poder simular un entorno de conducción realista y se deben poder manifestar eventos durante la conducción.

De esta forma, teniendo los anteriores componentes se deberá poder examinar el desempeño en conducción de una persona mientras utiliza una aplicación en un dispositivo secundario, como puede ser un celular o una tablet. Esto se logra ya que, mientras el usuario conduce y realiza las tareas de medición de la carga cognitiva, deberá utilizar la aplicación provista (por ejemplo, la interfaz de una aplicación prototipo que se quiera probar). En el caso de que la aplicación dada sea muy demandante cognitivamente, el conductor deberá prestarle mayor atención a la aplicación y en menor medida a la conducción. Esto se verá reflejado en los resultados obtenidos por el componente encargado de medir la carga cognitiva.

Se requiere disponer de la información recopilada durante las experimentaciones. Se deben poder conocer los eventos que sucedieron durante la simulación, en el orden cronológico adecuado. Las diferentes acciones que tomó el conductor y el estado de su visión durante las experimentaciones. Estos datos deben estar disponibles en todo momento y deben ser de fácil acceso, con la posibilidad de exportarlos a diferentes formatos para su posterior procesamiento.

Adicionalmente se implementará una interfaz, cuyo propósito es consumir los datos almacenados previamente y mostrar, mediante algún tipo de gráfico o herramienta de visualización, todos los eventos y situaciones que sucedieron durante la experimentación. Esto servirá para poder evaluar el desempeño y el comportamiento del conductor.

### 3.2. Diseño y desarrollo de propuesta

A partir de los requerimientos de uso, hardware y software especificados arriba, queda planteado que para realizar la evaluación de la carga cognitiva de una persona requerimos de un ambiente controlado que permita la inmersión del conductor en el entorno de simulación, cuya implementación sea sencilla y accesible, focalizando en su uso práctico para cualquier desarrollador de aplicaciones para automóviles.

Para llevar a cabo las pruebas se requiere como mínimo:

**Una computadora** capaz de ejecutar los programas de simulación y visualización de resultados.

**Un conjunto de periféricos** que simulen los controles correspondientes necesarios para la conducción, por ejemplo un teclado o joystick.

**Una pantalla o televisor** para simular el parabrisas, si no se dispone de una pantalla de gran tamaño, se pueden utilizar varias pantallas y combinarlas en una sola vista. Siendo que esta última funcionalidad la provee la suite desarrollada.

**Una cámara web** para poder realizar el reconocimiento facial del conductor y poder determinar en que momentos el conductor pierde atención. No es necesario disponer de una cámara de gran calidad, incluso es posible obtener buenos resultados utilizando la cámara de un celular común y corriente.

Esta configuración se puede extender con otros artefactos disponibles para mejorar la simulación de la conducción de un auto. Por ejemplo, para las pruebas realizadas en esta tesina se utilizó un volante, pedales y palanca de cambios (G29 Driving Force Racing Wheel) conectados a la computadora para lograr un mayor nivel de inmersión (ver Figura 3.2).



Figura 3.2: Ambiente de prueba y desarrollo.

### 3.2.1. Ejecución de componentes

En cuanto a la parte de software, se decidió realizar un desarrollo modular y organizado en componentes con el fin de facilitar la administración y mantenibilidad de los componentes en el futuro.

El lenguaje de programación utilizado en la mayor parte de los componentes es Python, ya que provee una gran flexibilidad y dinamismo a la hora del desarrollo.

Esta arquitectura basada en componentes fue pensada con el objetivo de tener un entorno de ejecución distribuido. Por lo tanto, en caso de requerir un mayor nivel de procesamiento, se pueden ejecutar los componentes de la suite en diferentes computadoras que se comunican a través de la red.

En las siguientes secciones se describirá con mayor detenimiento las diferentes conexiones entre los componentes y su conformación.

La opción de ejecución distribuida es fundamentalmente interesante a la hora de decidir en dónde ejecutar el simulador CARLA [24], ya que este componente en específico puede resultar muy demandante si no se posee el poder de computo adecuado. Según su documentación oficial se exponen los siguientes requerimientos:

- Requerimiento del sistema. Cualquier Sistema Operativo de 64-bits.
- Una GPU adecuada. CARLA realiza simulaciones realistas, así que el servidor necesita una GPU con 6GB de memoria como mínimo, aunque se recomienda una de 8GB.
- Espacio en el disco. CARLA requiere cerca de 20GB de espacio.

Las pruebas realizadas y el desarrollo de la suite, fue llevado a cabo en dos entornos. El primero en una computadora personal con la siguiente especificación:

- Procesador: Intel Core i7-4790K
- GPU: GeForce GTX 1060 6GB
- Memoria RAM: 16GB
- Sistema Operativo: Ubuntu 19/20/21

El segundo fue en el laboratorio del Centro de Investigación LIFIA<sup>1</sup> en la Facultad de Informática de la UNLP<sup>2</sup>, en donde se dispuso de varias computadoras para poder ejecutar dentro de una red LAN los componentes de forma distribuida.

El equipo utilizado mayormente para las actividades relacionadas con el desarrollo y la ejecución del simulador posee las siguientes características:

- Procesador: Intel Core i5 3330 3.20GHZ QuadCore
- GPU: Evga GeForce 6Gb GTX1660
- Memoria RAM: 16 Gb RAM
- Sistema Operativo: Ubuntu 19

---

<sup>1</sup><https://lifa.info.unlp.edu.ar/>

<sup>2</sup><https://www.info.unlp.edu.ar/>

### 3.3. Componentes desarrollados

En esta sección se describirá con un mayor grado de profundidad los componentes implementados, las relaciones entre los mismos y su comunicación.

#### 3.3.1. Plataforma de simulación

Este componente es uno de los más importantes ya que es el encargado de ejecutar la simulación de conducción. Además, es el elemento al cual se suscriben los demás componentes para obtener información sobre los eventos y acciones que suceden durante su ejecución. Se hablará sobre las características del simulador CARLA y como se puede interactuar con él, a través de la API en Python.

#### **CARLA: Open-source simulator for autonomous driving research**

Entre las diferentes opciones a disposición para utilizar como motor de simulación, se decidió usar CARLA en su versión 0.9.6. Este simulador desarrollado en Intel Labs cuenta con la ventaja de ser un proyecto reconocido y en constante crecimiento. El proyecto es de carácter Open Source, lo que hizo muy fácil su integración en la suite y luego su posterior explotación para el desarrollo de nuevas funcionalidades.

Además del motor de simulación, CARLA provee un vasto catálogo de recursos digitales como diseños urbanos, edificios, vehículos y peatones que logran enriquecer las experimentaciones realizadas y las vuelven más atractivas en comparación a otros simuladores.

Otro aspecto interesante de CARLA es que está en constante desarrollo y tiene más funcionalidades que las utilizadas para esta tesina, esto presenta un gran margen de oportunidades y posibilidades de trabajo a futuro.

En cuanto a su estructura, CARLA utiliza Unreal Engine<sup>3</sup> para la ejecución y maquetado de la simulación. El control y la configuración de la simulación está dado mediante una API accesible mediante Python y C++ (ver Figura 3.3).

El simulador está compuesto por una arquitectura cliente-servidor en la cual el servidor es responsable por el procesamiento de todas las acciones y objetos relacionados con la simulación en sí misma: sensores, computación de físicas, actualizaciones en el estado del *mundo* simulado y sus actores, etc.

---

<sup>3</sup>[www.unrealengine.com](http://www.unrealengine.com)

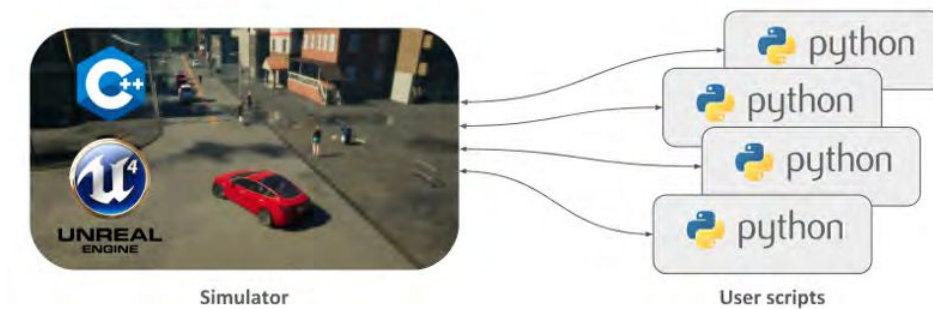


Figura 3.3: Control de la simulación utilizando la API en Python. Fuente: Documentación de CARLA

La parte del cliente consiste en la suma de módulos aportados por el usuario, los cuales utilizan la API desarrollada en Python y se encargan de controlar la lógica de los actores en el mundo, junto con la configuración del mismo. Para las pruebas realizadas en esta tesina los módulos se han ejecutado con la versión de Python 3.7.0.

## Módulos

Para la ejecución del cliente se disponen de 3 módulos.

El primero y más importante es el encargado de iniciar la simulación, para esto se conecta con el servidor de CARLA, indicando el mapa en el cual correr la simulación. Internamente, las partes relacionadas con el manejo de eventos y la ventana encargada de mostrar la simulación se encuentra desarrollada con la librería para videojuegos de Python, Pygame<sup>4</sup>.

El segundo módulo se puede ejecutar opcionalmente, para controlar de forma dinámica el clima presente en la simulación. Este programa se encarga de movilizar el sol con respecto al paso del tiempo y generar tormentas ocasionales (por ejemplo, en la Figura 3.4 se muestran variantes de climas para una misma escena).

El tercero es interesante ya que permite controlar la cantidad de Non-Playable Characters (NPCs) distribuidos por el mapa, estos son autos y peatones a los cuales CARLA internamente denomina como *actores*. Los actores son importantes para la simulación porque le brindan vida, dinamismo y un mayor grado de dificultad a la hora de recorrer el mapa.

<sup>4</sup><https://www.pygame.org/>



Figura 3.4: Diferentes climas en la simulación. Fuente carla.org

Debido a que el módulo encargado de iniciar la simulación es el principal y los demás dependen de él, en las siguientes secciones se lo considerará como “cliente” a pesar de ser una de las partes que lo componen.

### Configuración

El cliente es altamente configurable, a la hora de ejecutar la aplicación, se puede especificar a modo de parámetros en su ejecutable diferentes aspectos (ver la Figura 3.5). Algunos de los más destacables son:

- El host (IP) donde se ejecutará el servidor para realizar la conexión.
- El puerto en el que el servidor estará escuchando a la espera de nuevas conexiones.
- Si la simulación correrá en piloto automático, esto significa que el jugador no necesita conducir el auto para interactuar con la simulación.
- El tipo de periférico a utilizar, por ejemplo si se utilizará el teclado, un volante o un joystick.

- La resolución deseada, en caso de querer modificar el tamaño de la ventana.
- Si se debe proyectar la simulación en todos los monitores disponibles en pantalla completa.

## Sensores

Se dispone de varios sensores que permiten reaccionar ante eventos y conocer las características del entorno en el que se conduce. Estos sensores también sirven para añadirle funcionalidad a los actores dentro de la simulación. Algunos de los sensores más destacados son:

- Sensor de colisión: registra un evento cada vez que el actor asignado colisiona contra alguna cosa en el mundo simulado.
- Sensor GNSS: permite conocer la ubicación exacta del actor.
- Sensor de invasión de carril: registra un evento cada vez que el actor asignado cruza una calle o superficie determinada. Las diferentes superficies son definidas en el momento de la construcción del mapa.
- Cámara RGB: es la encargada de capturar las imágenes de las distintas escenas durante la simulación y transmitir las al cliente para mostrarlas con PyGame.

## Eventos

Como se nombró, muchos de los elementos que conforman al cliente generan eventos, los cuales pueden resultar interesantes a la hora de suscribir nuevos componentes con el fin de utilizar esa información para nuevas funcionalidades. Para facilitar esto, se desarrolló el cliente de forma tal que sea sencillo agregar nuevos componentes que escuchen eventos de la simulación. A grandes rasgos existen 3 tipos de eventos:

- Frame events: se produce un evento por cada frame de la simulación y se obtiene información sobre el mundo junto con el reloj interno usado por la simulación para sincronizar eventos.
- Display events: se producen en cada actualización de la ventana y se transmite un objeto display sobre el cual se pueden dibujar y controlar elementos que se muestran en pantalla.





Figura 3.5: Vista de la simulación y su display con información sobre el entorno.

- Key events: en caso de que los componentes sean interactivos y necesiten estar pendientes de eventos del teclado, se puede registrar un callback bajo una tecla específica en el controlador del periférico.

### 3.3.2. Detection Response Tasks

Este componente se encarga de la administración de las actividades antes mencionadas como Detection Response Task (DRT) que se ejercitarán durante la ejecución del simulador.

#### Funcionamiento

El funcionamiento está dado por un aviso que aparece cada cierto tiempo, el cual debe ser respondido con la interacción del usuario en el menor tiempo posible. Concretamente para esta tesina, el aviso toma forma de un pequeño círculo de un color ostentoso, simulando una luz en el parabrisas.

Luego de una determinada cantidad de segundos, este aviso se dibuja en la pantalla y si no se presiona una tecla especial para confirmar el aviso en un pequeño lapso, este aviso desaparece y se registra como una falla.

Para el funcionamiento de este componente se aprovecharon algunas de

las suscripciones a eventos brindadas por el simulador, las cuales fueron comentadas en el anterior apartado. Para su correcto funcionamiento, el componente DRT debe poder manipular el display de la simulación y así dibujar los elementos correspondientes en pantalla.

Además se debe registrar la funcionalidad a ejecutar luego de presionar determinadas teclas.

### Configuración

El componente DRT es configurable mediante un archivo en formato JSON que se encuentra en el directorio base del proyecto. Las opciones a configurar son:

- El radio del círculo determinado en píxeles.
- El color expresado en hexadecimal.
- El tiempo en segundos que dura la luz prendida antes de apagarse.
- El modo de ejecución: éste puede ser manual si se desea prender y apagar la luz a voluntad, o aleatorio en caso de querer que la luz aparezca luego de un lapso elegido al azar de segundos. Se puede determinar un intervalo para la selección del tiempo.
- La ubicación de la luz, en este caso se puede optar por una ubicación fija en la pantalla determinada por coordenadas en el eje cartesiano. Como alternativa a esta opción se puede elegir una posición aleatoria en la pantalla. Para evitar entorpecer la vista del conductor mientras se mueve por el mundo, se pueden elegir cuadrantes de la pantalla en donde puede aparecer la luz.

#### 3.3.3. Reconocimiento facial

Con el objetivo de obtener más información sobre el estado del conductor durante la simulación, se desarrolló un componente encargado de monitorear la vista del usuario a través de una cámara. Esto es muy valioso ya que complementa el objetivo de las tareas DRT y ayuda a identificar la atención del conductor durante las pruebas.

Para el desarrollo de este componente se utilizó la librería OpenCV<sup>5</sup>, la cual brinda una gran cantidad de funcionalidades para la detección de

---

<sup>5</sup><https://opencv.org/>

elementos y personas en tiempo real. De esta forma, utilizando el mecanismo de *Haar Cascade Classifier* es posible identificar objetos en imágenes o videos, sin importar su ubicación ni tamaño. Además, este algoritmo es muy rápido lo cual permite detectar elementos en tiempo real.

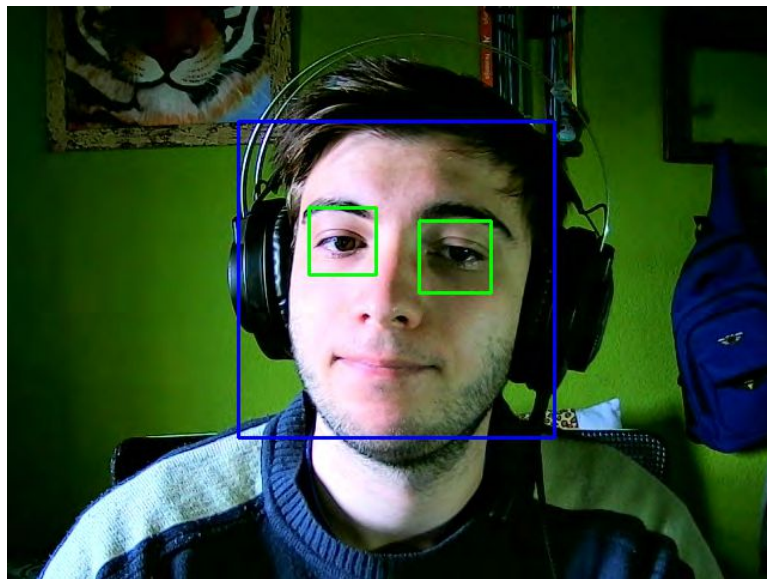


Figura 3.6: Reconocimiento facial y de ojos en vivo.

Así, con un modelo Haar Cascade pre-entrenado y utilizando las funcionalidades provistas por OpenCV, es posible detectar en tiempo real los movimientos del conductor. Marcando como pérdida de atención los momentos en los que el algoritmo deja de detectar un rostro en la imagen, ésto se puede dar por ejemplo, cuando el conductor se distrae revisando una aplicación en una pantalla externa y quita la vista de la pantalla de enfrente o parabrisas (en la Figura 3.6 se muestra un ejemplo).

No hay que olvidar que uno de los objetivos de esta tesina es lograr obtener un entorno de prueba y prototipado rápido y económico, para que sea de alcance para la mayor cantidad de desarrolladores posible. Teniendo esto en mente, se debe destacar que no es necesario disponer de una cámara de gran calidad para un correcto funcionamiento. Incluso es posible, mediante aplicaciones móviles, establecer una conexión entre la cámara de un celular y la computadora, de esta forma convirtiéndolo efectivamente en una cámara web. Con este fin, durante el desarrollo fue de gran utilidad la aplicación

Droidcam<sup>6</sup> que es compatible con Windows y Linux.

### 3.3.4. Servidor REST de eventos

Teniendo los anteriores componentes, nos encontramos con la necesidad de centralizar la información que producían por separado, volcándola en una base de datos, para luego poder realizar consultas y trabajar con la información. Además para conservar la posibilidad de la ejecución distribuida del software, era necesario eliminar la posibilidad de que todos los componentes escribieran en un mismo archivo.

Para solucionar esto se construyó un pequeño servidor, siguiendo el modelo REST con la finalidad de almacenar los diferentes eventos que se producen en todos los componentes. Esto requirió agregar código adicional a los mismos, encargado de anunciar periódicamente al servidor las diferentes acciones que transcurren durante la experimentación.

La base de datos utilizada es MySQL<sup>7</sup>, mientras que el servidor fue desarrollado utilizando Django<sup>8</sup> en combinación con la librería Django Rest Framework<sup>9</sup> (DRF) la cual nos facilita la creación de las vistas, URLs y serializadores necesarios para el correcto funcionamiento del servidor. Ésto nos permitió enfocarnos en el modelo de datos utilizado, implementándose dos modelos. El primero contiene información sobre los **eventos** indicando su nombre, hora de ocurrencia y estado. El segundo permite diferenciar los distintos **experimentos** que se ejecutan, abarcando información sobre los eventos que transcurrieron en él, la fecha y hora de inicio y la de finalización.

Otra ventaja que posee disponer de un servidor REST es la posibilidad de ser consultado por cualquier dispositivo o navegador mediante consultas HTTP. De esta forma, es posible consumir la información contenida en la base de datos, con una aplicación web que se encargue de mostrar información relevante sobre los experimentos realizados.

### 3.3.5. Timeline chart

Con el fin de poner en práctica el consumo de la información almacenada en el servidor REST antes mencionado, se desarrolló una aplicación web utilizando la librería React<sup>10</sup> para la visualización de los eventos y control de la simulación.

---

<sup>6</sup><https://www.dev47apps.com/>

<sup>7</sup><https://www.mysql.com/>

<sup>8</sup><https://www.djangoproject.com/>

<sup>9</sup><https://www.django-rest-framework.org/>

<sup>10</sup><https://reactjs.org/>

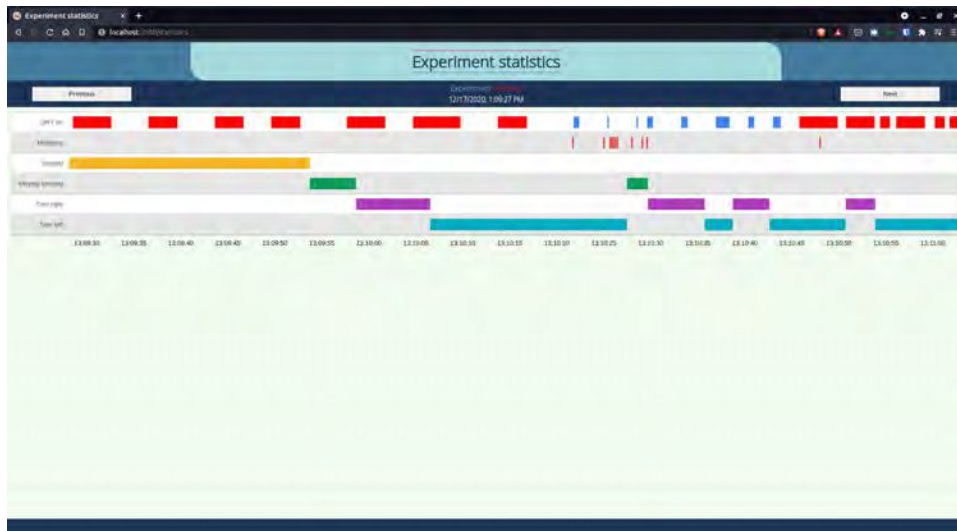


Figura 3.7: Línea de tiempo con los eventos sucedidos en un experimento.

El objetivo principal de esta aplicación es tener una vista general, mediante una línea de tiempo, de todos los sucesos que ocurrieron durante los experimentos con el simulador. Esta línea de tiempo, desarrollada utilizando la librería React Google Charts<sup>11</sup>, nos permite acceder a información almacenada sobre experimentos previos, y además poder visualizar en tiempo real los eventos que se producen en un experimento en vivo (ver Figuras 3.7 y 3.8).

Adicionalmente para facilitar la ejecución de las pruebas, se añadió un botón que permite inicializar todos los componentes requeridos para la ejecución de un experimento. De esta funcionalidad se hablará con mayor profundidad en la siguiente sección.

### 3.3.6. Launcher de componentes

Como vimos hasta ahora, hay muchos componentes con configuraciones distintas y resulta laborioso realizar la ejecución de todos individualmente. Es por ésto que para simplificar el sistema de ejecución y configuración de parámetros, se desarrolló un pequeño servidor encargado de inicializar las diferentes partes de la suite de experimentación.

<sup>11</sup><https://react-google-charts.com/>

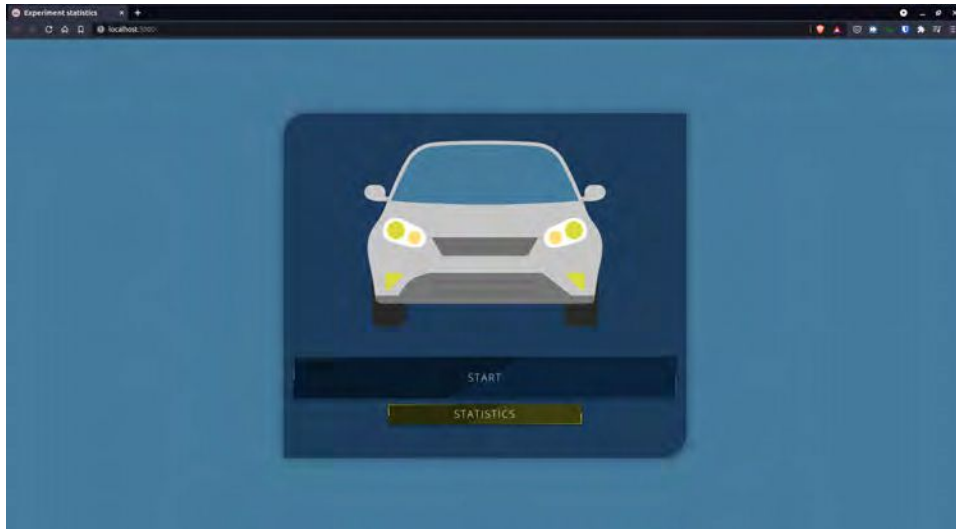


Figura 3.8: Página principal de la aplicación web.

El servidor está desarrollado utilizando el framework Flask<sup>12</sup>, al momento de su inicialización se ejecutan los comandos necesarios para el lanzamiento de los componentes Timeline chart y el Servidor REST de eventos. Ésto es necesario ya que utilizaremos el componente Timeline chart como interfaz principal para comunicarse con el launcher de componentes y realizar la petición de ejecución del simulador. La petición se realiza una vez apretado el botón START en el menú principal de la aplicación. De esta forma, el componente se conecta con el servidor de ejecución de componentes en un endpoint *start-simulation*, que se encarga de lanzar el simulador CARLA y todos los componentes asociados.

---

<sup>12</sup><https://flask.palletsprojects.com/en/2.0.x/>

## Capítulo 4

# Validación

Para poder validar que esta propuesta se puede integrar en un proceso de diseño de IVIS para lograr una medición efectiva del nivel de carga cognitiva y de distracción que impone una aplicación sobre el conductor realizamos experimentos con sujetos de prueba utilizando DRT[25] y el framework de evaluación de usabilidad para IVIS[26].

Para la ejecución de la suite de prueba utilizamos aplicaciones que pueden utilizarse como IVIS y que tienen diferente grado de complejidad en sus interfaces de usuario, en la hipótesis de que las demandas cognitivas y nivel de distracción sobre el conductor puede relacionarse con esa complejidad. Los resultados obtenidos de los experimentos soportan en principio esa hipótesis inicial.

Seguramente serán necesarias pruebas más extensivas para llevar la plataforma a una instancia productiva, pero en la instancia de desarrollo de este trabajo los experimentos realizados nos permiten validar el enfoque y la propuesta que se presenta en esta Tesina.

### 4.1. Objetivo

El principal objetivo es conocer y poder interpretar el comportamiento del conductor en diferentes entornos automovilísticos variando la complejidad de las aplicaciones móviles que se prueban. La hipótesis es que las interfaces más complejas requieren un mayor nivel de atención y compromiso por parte del conductor.

## 4.2. Definición de complejidad en interfaces

En la evaluación de la suite se seleccionarán dos aplicaciones, una con una interfaz compleja y otra con una interfaz sencilla. Definiremos la complejidad de la interfaz en base a su composición, colores, cantidad de botones, tamaño de los botones, cantidad de texto, tamaño del texto, entre otras variables[27]. En el contexto de IVIS, la complejidad de una interfaz de usuario depende del nivel de interacción física e intelectual que requiere para su correcto funcionamiento [28].

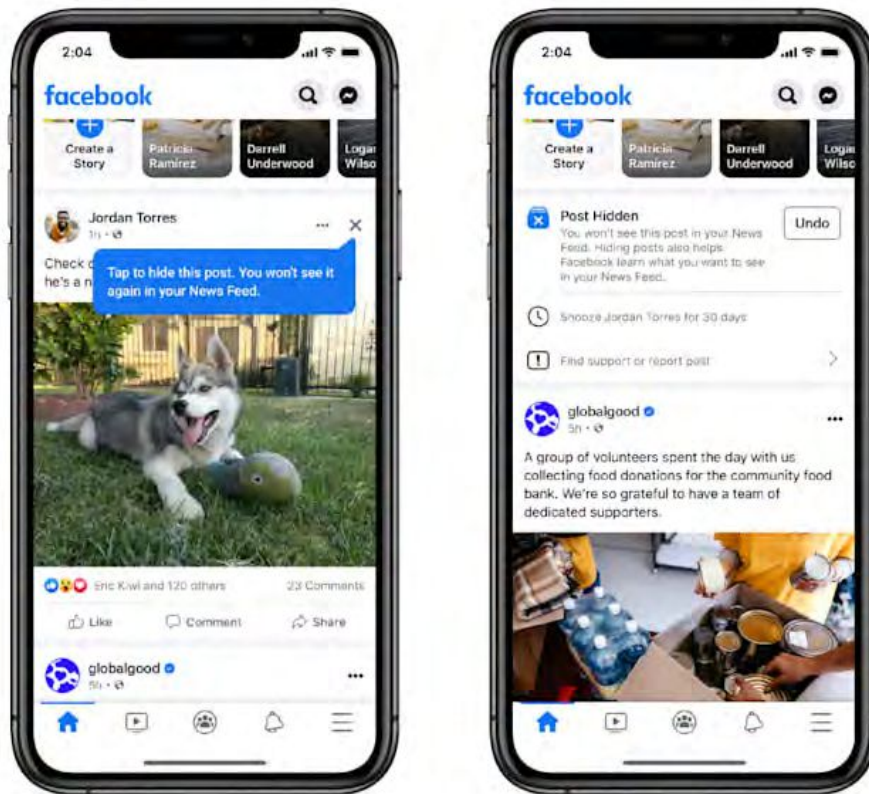


Figura 4.1: Interfaz compleja con mucho texto, imágenes y botones.

Para los experimentos se eligieron dos aplicaciones con distinta complejidad en sus interfaces de usuario.



Para la selección de una interfaz compleja se consideraron las redes sociales. Este tipo de aplicaciones están diseñadas para cautivar la atención de sus usuarios, cuentan con una gran cantidad de colores y botones. Además por la naturaleza de las redes sociales, es probable que despiertan el interés de los conductores teniendo un mayor impacto y presencia en la actividad mental del participante que el mismo acto de conducir. Dentro de las diferentes opciones de redes sociales como Twitter, Instagram, Whatsapp, Reddit se optó por seleccionar Facebook. Ya que ésta es una de las aplicaciones más populares del mercado y además posee un equilibrio entre la cantidad de texto e imágenes mostradas en pantalla (ver Figura 4.1).

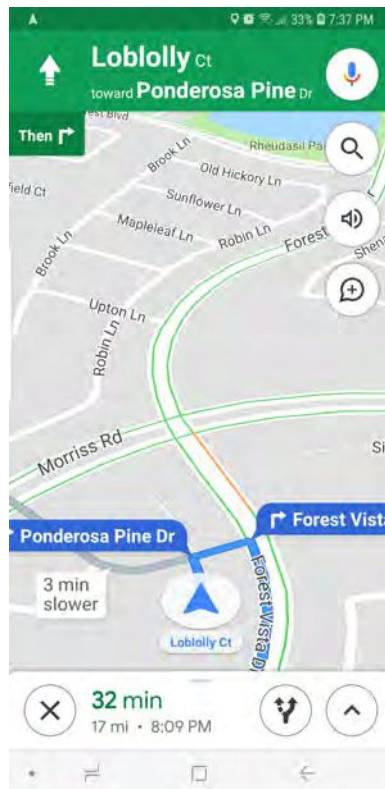


Figura 4.2: Interfaz sencilla con botones grandes y bien definidos.

Por otra parte, para el caso de la interfaz sencilla se optó por una de las aplicaciones más utilizadas por conductores y que se encuentra diseñada pensando en ese mismo público. La aplicación elegida es Google Maps, cuen-

ta con una interfaz intuitiva, fácilmente reconocible, cargada con muy pocos botones y texto. Esta aplicación requiere muy poca interacción por parte del conductor ya que en general solo debe ser manipulada una vez, al momento de su configuración. Luego solo bastará con ver las direcciones que se deben tomar para llegar a destino o incluso, escuchar las indicaciones dictadas por una voz de la aplicación sin mirar la pantalla (ver Figura 4.2).

### 4.3. Participantes y equipamiento

En los experimentos participaron 5 sujetos (2 mujeres, 3 hombres) con un rango de edad de 21 a 56 años (media=33.3, mediana=24). Todos los sujetos cuentan con licencia vigente para conducir vehículos y ninguno posee dificultades visuales o auditivas.



Figura 4.3: Volante y pedales utilizados para la experimentación.

El equipamiento empleado incluye dos computadoras conectadas a la misma red LAN. Una de las computadoras está encargada del procesamiento del servidor y de todos los componentes necesarios para ejecutar la simulación, ésta es utilizada por el conductor designado para realizar el experimento. La otra computadora es utilizada por el personal encargado de facilitar las pruebas, en esta computadora se accede a la información recopilada durante la simulación mediante la interfaz de visualización de

eventos.

Los sujetos se ubican en un asiento ergonómico de oficina frente a 3 pantallas contiguas de 23 pulgadas cada una, las cuales servirán a modo de parabrisas de auto, generando una mayor inmersión del conductor. Además, los participantes controlarán la simulación utilizando un volante y pedales para juegos de conducción que presentan una gran similitud a periféricos de conducción reales presentes en autos (ver Figura 4.3).

Por último se ubicará un pequeño dispositivo (en este caso un teléfono celular) a la altura del volante de forma que no obstruya la visión del parabrisas por el conductor.

#### 4.4. Diseño del experimento

La tarea asignada a los conductores consiste en realizar un recorrido breve por la ciudad durante 5 minutos. El participante deberá intentar respetar las normas de tránsito (detenerse en los semáforos con luz roja, evitar colisionar con otros autos, mantener el control del auto en todo momento y permanecer en el carril correspondiente).



Figura 4.4: Simulador de conducción en un día lluvioso y se muestra el punto rojo correspondiente al DRT en pantalla

De manera aleatoria, se mostrará un punto rojo (luz DRT) en una zona determinada de la pantalla. En esa situación, el conductor debe presionar tan pronto como pueda la tecla correspondiente al botón “R2” en el volante, la cuál desactiva el punto rojo. En caso de no presionar el botón a tiempo, el punto desaparecerá y se almacenará la falla por parte del conductor, teniendo éstas un mayor peso en el cálculo del nivel de distracción (en la Figura 4.5 se observan los botones en el volante utilizado).



Figura 4.5: Primer plano del volante utilizado, se pueden apreciar los distintos botones que controlarán diferentes aspectos de la simulación.

Para completar el ambiente de simulación, se utilizará el celular mencionado antes para ejecutar las aplicaciones que se desean probar. Se le indicará al conductor las tareas que debe realizar en cada aplicación mientras conduce por la ciudad.

En el caso del uso de Facebook se deberán leer 8 publicaciones, darles *Me gusta* y acceder al perfil de una persona.

Para el uso de Google Maps se debe configurar el viaje hacia un destino,

se debe añadir una parada intermedia luego de iniciado el viaje y por último se deberá seleccionar una ruta alternativa.

Resumiendo, el participante deberá conducir un auto de forma correcta mientras utiliza un IVIS, simultáneamente deberá resolver la tarea de DRT la cuál nos permite estimar el nivel de carga cognitiva del conductor.

El factor determinante para decidir la interfaz con menor distracción, será obtener la menor cantidad de avisos fallidos del componente de Detection Response Task (punto rojo). En caso de que se obtenga la misma cantidad o que haya muy poca diferencia de avisos fallidos en ambas interfaces, se tendrá en cuenta el tiempo medio de reacción, desde que aparece la luz hasta que el usuario la apaga.

Cada participante realiza 3 rondas de simulación de 5 minutos cada una con cada una de las aplicaciones (6 simulaciones en total). Para evitar el sesgo de aprendizaje, el orden en el que se presentan las diferentes aplicaciones es aleatorio.

## 4.5. Recolección de datos y resultados

Todas las pruebas se realizaron durante el mismo día con una separación entre usuarios de media hora. Antes de comenzar las pruebas, cada conductor dispuso de un tiempo de 2 minutos para acostumbrarse al manejo de los controles en la simulación. La serie de 6 rondas se realizó con una separación de 5 minutos entre cada una, totalizando un carga de prueba de 55 minutos (30 de conducción, 25 de descanso) por sesión de usuario. Las instrucciones fueron repetidas antes de cada inicio de ejecución de una aplicación.

Al finalizar cada sesión se realizó una encuesta basada en el Índice de Carga de Actividad del Conductor (DALI, por sus iniciales en inglés), que es un método para mediar la carga de trabajo subjetiva del usuario[26].

A continuación se muestran los resultados obtenidos luego de la evaluación de las diferentes interfaces con los participantes.

El primer análisis consiste en evaluar la media de DRT perdidos, errores y tiempos de distracción entre las dos aplicaciones para el conjunto de usuarios:

- Los *DRT perdidos* indican la cantidad de alertas que no fueron atendidas a tiempo y desaparecieron,
- los *Errores* contabiliza la cantidad de veces que se presionó el botón de confirmación de la alerta y no había punto rojo, esto se contabiliza

para evitar que el conductor presione constantemente el botón de DRT mientras no está la notificación en pantalla.

- el *Tiempo de distracción* se lo mide en segundos e indica el tiempo promedio que el conductor mantiene la vista fuera de la simulación cada vez que revisa la aplicación secundaria. Para el cálculo de esta variable se obtuvo el promedio de cada conductor y luego se promediaron los resultados de todos los participantes. Además cabe destacar que ésta variable se mide con el dispositivo de detección de rostros.

Se puede ver en la Figura 4.6 el contraste de los resultados obtenidos para las dos interfaces.

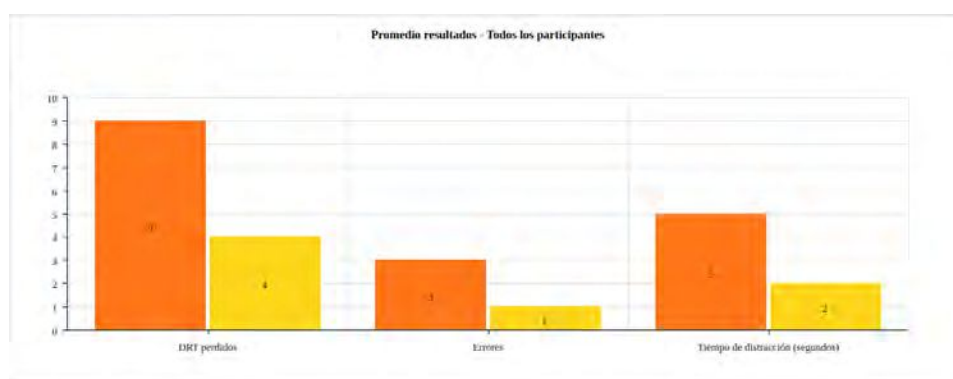


Figura 4.6: Promedio de resultados entre todos los participantes.

Para la interfaz compleja (columna izquierda de cada par) se percibe un mayor grado de carga cognitiva por parte del conductor. Mientras que para la otra interfaz se produjeron muchos menos errores y pérdidas de la alerta DRT.

Un segundo análisis consistió en evaluar posibles diferencias para las aplicaciones en dos rangos etáreos de los sujetos según la mediana del universo poblacional. Quedan dos grupos, A con usuarios de 21 a 24 años y B, 53 a 56. Presentamos dos gráficos, el primero para la aplicación con IU compleja (Fig. 4.7) y el segundo con la IU simple (Fig. 4.8). En los gráficos, para cada indicador la columna izquierda contiene resultados de grupo A y a la derecha, el B.

Es notable que los más jóvenes tienen mayor tiempo de distracción y mayor cantidad de DRTs perdidos, pero cometen menos errores tanto para las

aplicaciones complejas como las simples. Quizá una explicación se relacione con el poder persuasivo de las aplicaciones en adolescentes.

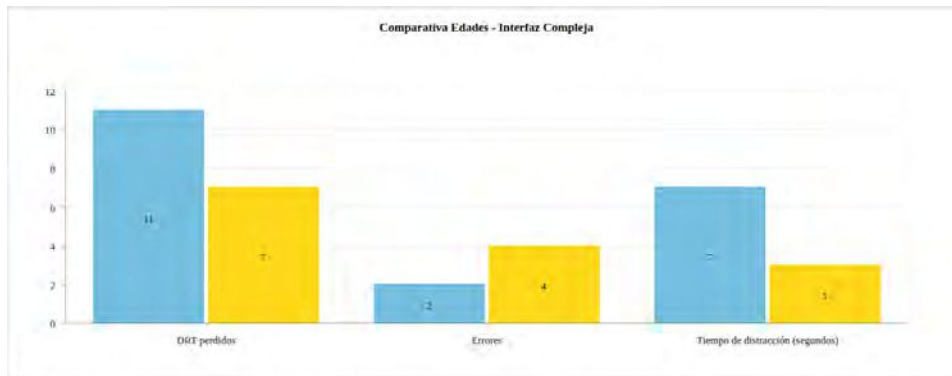


Figura 4.7: Comparativa interfaz compleja con promedio de resultados entre ambos grupos de edades.

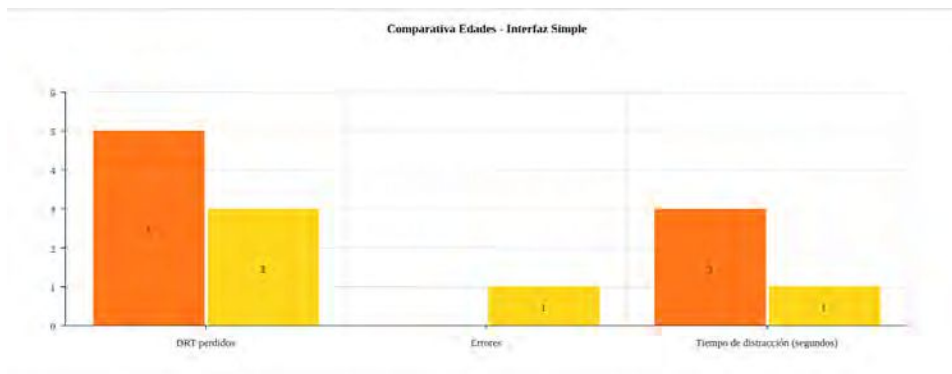


Figura 4.8: Comparativa interfaz simple con promedio de resultados entre ambos grupos de edades.

#### 4.5.1. Encuesta posterior

Finalmente, se realizó un cuestionario a los sujetos según la guía de preguntas del Índice DALI para cada una de las aplicaciones (una para las tres rondas de IU compleja y otra para las 3 de IU simple). El cuestionario

DALI consiste en una escala de Likert de 6 posiciones (0 a 5, bajo a alto) sobre: demanda de atención global, demanda visual, auditiva, táctil, nivel de stress, presión temporal, grado de interferencia de tarea secundaria sobre conducción.

Los participantes describieron niveles altos de DALI para la aplicación de IU compleja, en particular en términos de grado de interferencia sobre la conducción. En forma espontánea, además, los sujetos del grupo etéreo B (53-56) hicieron comentarios sobre la dificultad de lectura de la aplicación y la carga de elementos de la interfaz.

#### **4.5.2. Conclusión**

Luego de los experimentos realizados y la posibilidad de recolectar y analizar una buena cantidad de información para el proceso de diseño y desarrollo de aplicaciones, creemos que el trabajo dedicado para obtener información sobre el estado del conductor durante la experimentación se simplifica en gran medida con la plataforma propuesta.

La plataforma propuesta y su posibilidad de utilización en el marco de un desarrollo de aplicaciones móviles, permite el uso de los frameworks de evaluación habitual para IVIS y satisface los requerimientos planteados en la sección *3.1.2 Requerimientos a satisfacer*.



## Capítulo 5

# Conclusiones y trabajo futuro

En esta tesina, se ha estudiado el estado del arte con respecto a la cantidad de accidentes viales y costos relacionados en diferentes países del mundo. Se investigaron las principales causas de los mismos y se determinó, que un gran porcentaje de los accidentes viales es producido por distracciones utilizando IVIS.

También se analizaron las dificultades que existen a la hora del desarrollo de una aplicación pensada como IVIS, la cantidad de recursos necesarios para las pruebas y la falta de una herramienta económica, que permita comprobar la carga cognitiva de los conductores mientras utilizan aplicaciones.

A partir de lo investigado, se puede ver que los desarrolladores de aplicaciones generales, los cuales no trabajan en fábricas de automóviles, no poseen los recursos necesarios para poder implementar IVIS de forma segura. Es decir, con interfaces que minimicen el nivel de atención requerido por parte del conductor.

Como consecuencia del anterior análisis se decide definir una suite de prueba y simulación sobre la atención humana utilizando IVIS, la cual pretende solucionar los problemas antes mencionados o mínimamente brindar un marco de ideas para trabajo futuro. En dicha suite se busca simular la conducción de un auto mientras se resuelven diferentes desafíos, los cuales buscan determinar la carga cognitiva de una persona mientras utiliza una aplicación dada. Para esto, se logró implementar un mecanismo de DRT y un detector de rostros con los que se puede monitorear el comportamiento del conductor. Luego, la información producida por los componentes se almacena de forma práctica, para facilitar su posterior consumo por otras

aplicaciones y generar estadísticas sobre los datos.

## 5.1. Trabajo futuro

CARLA es un proyecto relativamente nuevo que se encuentra en desarrollo, agregando funcionalidades nuevas constantemente. Esto significa que muy probablemente en el futuro, existan elementos interesantes que sean provechosos para la finalidad de esta tesina y la recopilación de información sobre la simulación. Algunas de estas posibilidades pueden ser nuevos sensores, mayor performance, nuevos mapas, nuevos autos, etc.

Otra oportunidad para extender la tesina, podría ser la implementación de nuevas o mejores aplicaciones consumidoras de la información recopilada durante la simulación. Se podrían generar mejores gráficos y diferentes estadísticas en muchos tipos de interfaces y dispositivos.

Finalmente, otra forma de extender esta tesina, podría ser mediante el agregado de nuevos eventos que permitan contextualizar con mayor facilidad el estado del vehículo y la simulación en todo momento. Por ejemplo, se podría disparar un evento en caso de colisión entre el vehículo conducido y otro actor en la simulación, como puede ser otro vehículo, un peatón, una pared, etc.

# Bibliografía

- [1] Strategic Highway Research Program, “The second Strategic Highway Research Program (2006-2015).” Accessed: 2021-06-26.
- [2] A. Blatt, J. Pierowicz, M. Flanigan, P.-S. Lin, A. Kourtellis, C. Lee, P. Jovanis, J. Jenness, M. Wilaby, J. Campbell, *et al.*, “Naturalistic driving study: Field data collection,” tech. rep., 2015.
- [3] T. Victor, M. Dozza, J. Bårgman, C.-N. Boda, J. Engström, C. Flanagan, J. D. Lee, and G. Markkula, “Analysis of naturalistic driving study data: Safer glances, driver inattention, and crash risk,” tech. rep., 2015.
- [4] Agencia Nacional de Seguridad Vial, “Situación de la seguridad vial en Argentina,” 2018. Accessed: 2021-06-26.
- [5] W. H. Organization *et al.*, *Informe sobre la situación mundial de la seguridad vial: es hora de pasar a la acción*. World Health Organization, 2010.
- [6] A. N. de Seguridad Vial (Ovservatorio vial), “Estudio observacional del comportamiento de conductores y ocupantes de vehículos motorizados de 4 (o más) y 2 ruedas,” 2019. Accessed: 2021-06-26.
- [7] C. Kaiser, A. Stocker, A. Festl, G. Lechner, and M. Fellmann, “A research agenda for vehicle information systems,” 2018.
- [8] C. Carvajal, A. Rodríguez, and A. Fernández, “The growing and risky industry of nomadic apps for drivers,” in *VI Iberoamerican Conference of Computer Human Interaction*, 2020.
- [9] “Eye tracking research in the field.” <https://www.tomshardware.com/news/tobii-pro-glasses-2-eye-tracking,33575.html>. Accessed: 2021-07-28.

- [10] K. Stojmenova and J. Sodnik, “Detection-response task—uses and limitations,” *Sensors*, vol. 18, no. 2, p. 594, 2018.
- [11] G. Weinberg and B. Harsham, “Developing a low-cost driving simulator for the evaluation of in-vehicle technologies,” in *Proceedings of the 1st International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pp. 51–54, 2009.
- [12] S. Agrawal, D. Song, S. Peeta, I. Benedyk, *et al.*, “Driving simulator based interactive experiments: Understanding driver behavior, cognition and technology uptake under information and communication technologies,” tech. rep., NEXTRANS Center (US), 2018.
- [13] F. Jiménez, ed., *Intelligent Vehicles: Enabling Technologies and Future Developments*. Butterworth-Heinemann, 2018.
- [14] M. Peräaho, E. Keskinen, and M. Hatakka, “Driver competence in a hierarchical perspective; implications for driver education,” *Report to Swedish Road Administration*, pp. 1–52, 2003.
- [15] B. Lang, A. Parkes, S. Cotter, R. Robbins, C. Diels, P. Vanhulle, G. Turi, E. Bekiaris, M. Panou, J. Kapplusch, *et al.*, “Benchmarking and classification of cbt tools for driver training,” 2007.
- [16] Adam Frost, “Transport for London converting its ‘One Model’ into Aimsun Next platform.” Accessed: 2021-08-14.
- [17] S. Mecheri and R. Lobjois, “Steering control in a low-cost driving simulator: a case for the role of virtual vehicle cab,” *Human factors*, vol. 60, no. 5, pp. 719–734, 2018.
- [18] D. de Waard, M. van der Hulst, M. Hoedemaeker, and K. A. Brookhuis, “Driver behavior in an emergency situation in the automated highway system,” *Transportation human factors*, vol. 1, no. 1, pp. 67–82, 1999.
- [19] E. I. Farber, “Comments on”driver behavior in an emergency situation in the automated highway system”,” *Transportation Human Factors*, vol. 1, no. 1, pp. 83–85, 1999.
- [20] F. Bella, “Driving simulator for speed research on two-lane rural roads,” *Accident Analysis & Prevention*, vol. 40, no. 3, pp. 1078–1087, 2008.
- [21] S. T. Godley, T. J. Triggs, and B. N. Fildes, “Driving simulator validation for speed research,” *Accident analysis & prevention*, vol. 34, no. 5, pp. 589–600, 2002.

- [22] S. L. Jamson and A. H. Jamson, “The validity of a low-cost simulator for the assessment of the effects of in-vehicle information systems,” *Safety Science*, vol. 48, no. 10, pp. 1477–1483, 2010.
- [23] I. Alvarez, L. Rumbel, and R. Adams, “Skyline: a rapid prototyping driving simulator for user experience,” in *Proceedings of the 7th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, pp. 101–108, 2015.
- [24] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Conference on robot learning*, pp. 1–16, PMLR, 2017.
- [25] K. Stojmenova and J. Sodnik, “Detection-response task—uses and limitations,” *Sensors*, vol. 18, no. 2, 2018.
- [26] C. Harvey and N. A. Stanton, *Usability evaluation for in-vehicle systems*. Crc Press, 2013.
- [27] A. Riegler and C. Holzmann, “Measuring visual user interface complexity of mobile applications with metrics,” *Interacting with Computers*, vol. 30, no. 3, pp. 207–223, 2018.
- [28] Y. W. Kim, D. Y. Kim, and Y. G. Ji, “Complexity in in-vehicle touchscreen interaction: A literature review and conceptual framework,” in *HCI in Mobility, Transport, and Automotive Systems. Automated Driving and In-Vehicle Experience Design* (H. Krömker, ed.), (Cham), pp. 289–297, Springer International Publishing, 2020.