



Universidad de Valladolid

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS
ESPECÍFICAS DE TELECOMUNICACIÓN - MENCIÓN
TELEMÁTICA

DESARROLLO DE UNA PLATAFORMA WEB PARA EL
ACCESO A LOS DATOS DE UN SIMULADOR DE
CONDUCCIÓN

Autor:

D. Francisco José Aguilar Marcos

Tutor:

D. David González Ortega

Valladolid, 28 de febrero de 2020

TITULO: Desarrollo de una plataforma web para el acceso a los datos de un simulador de conducción

AUTOR: D. Francisco José Aguilar Marcos

TUTOR: D. David González Ortega

DEPARTAMENTO: Departamento de Teoría de la Señal y Comunicaciones e Ingeniería Telemática

TRIBUNAL

PRESIDENTE: D. Míriam Antón Rodríguez

VOCAL: D. Mario Martínez Zarzuela

SECRETARIO: D. David González Ortega

SUPLENTE: D. Francisco Javier Díaz Pernas

SUPLENTE: D. José Fernando Díez Higuera

FECHA: 28 de febrero de 2020

CALIFICACIÓN:

Resumen

El objetivo principal de este trabajo de fin de grado ha sido la modificación y actualización de una plataforma web para el acceso a los datos de un simulador de conducción que actualmente está en funcionamiento. La plataforma funciona sobre el CMS (*Content Management System*) Drupal, que facilita la gestión de un servicio REST que se utiliza para la comunicación remota con el simulador de conducción.

En esta actualización de la plataforma se han incluido características y mejoras que no se tuvieron en cuenta en su desarrollo inicial. La necesidad de esta actualización se ha hecho patente con el paso del tiempo, al comprobar que no abarcaba todas las posibilidades del simulador y necesidades de los usuarios y que los contenidos de la misma no estaban correctamente referenciados.

Palabras clave

Simulación de conducción, tipo de vehículo, plataforma web, CMS Drupal, REST.

Abstract

The main objective of this end-of-degree project has been the modification and updating of a web platform to access to the data of a driving simulator that is currently in operation. The platform works on the Drupal CMS (*Content Management System*), which facilitates the management of a REST service that is used for remote communication with the driving simulator.

This platform update has included features and improvements that were not taken into account in its initial development. The need for this update has become apparent over time as it did not cover all the simulator possibilities and users' needs and that its contents were not correctly referenced.

Keywords

Driving simulation, type of vehicle, web platform, Drupal CMS, REST.

Agradecimientos

“Al equipo de Drupal y su comunidad que gracias a ellos he obtenido mucha información sobre este CMS consultando por internet”

“A mi tutor del trabajo de fin de grado David González Ortega por su soporte y gran apoyo”

Índice de contenidos

1.	Introducción.....	20
1.1	Motivación del proyecto	20
1.2	Objetivos del proyecto	21
1.3	Fases del proyecto.....	21
1.4	Medios utilizados (hardware y software).....	22
1.4.1	Hardware	22
1.4.2	Software	22
1.5	Estructura de la memoria	24
2.	Simuladores de conducción para la formación vial de usuarios.....	26
2.1	Simuladores de conducción de uso profesional con experiencia inmersiva	27
2.1.1	ENTAC - Simulador de conducción de vehículos pesados	27
2.1.2	Simulador de camión Gold Truck	28
2.1.3	Simulador de Vehículos Pesados – Simbio.....	30
2.1.4	Simulador de autobús Gold Bus	30
2.1.5	SMARTSIM - Simulador de conducción de automóvil	32
2.1.6	Silver - Simulador de automóvil de ALTA inmersión	33
2.1.7	El simulador de coche LANDER	34
2.1.8	Ambar - Simulador de automóvil de media inmersión.....	34
2.2	Simuladores de conducción de uso profesional para ordenador	35
2.2.1	DriverSim.....	36
2.2.2	Virtual Driver Interactive.....	37
2.3	Simuladores de conducción de uso doméstico para ordenador	37
2.3.1	City Car Driving	38
2.3.2	Driver Test.....	39
2.4	Simuladores de conducción para Android	39
2.4.1	Car Driving School Simulator	40
2.4.2	Dr. Driving 2	40
2.5	Comparativa de los simuladores de conducción.....	41
3.	Plataforma web: posibilidades de desarrollo	43
3.1	Frameworks para el desarrollo de aplicaciones web, ¿Qué son?	43
3.1.1	Zend Framework.....	43
3.1.2	Fuel PHP	44

3.1.3	Slim	44
3.2	Gestores de contenido (CMS)	45
3.2.1	CMS WordPress	45
3.2.2	Joomla!	46
3.2.3	Drupal	47
3.3	Comparativa de tecnologías y opción elegida	48
3.4	Instalación y requisitos	50
4.	Desarrollo de la plataforma web	53
4.1	Modificaciones relacionadas con la BBDD	53
4.2	Modificaciones de la página de inicio y el core de Drupal	58
4.3	Modificaciones del apartado Analysis of vehicle data	61
4.3.1	Modificaciones de la página Simulation data	62
4.3.2	Modificaciones de la página Simulation data per group	68
4.3.3	Modificaciones de la página Simulation data per user	72
4.4	Modificaciones del apartado Analysis of traffic offenses	80
4.4.1	Modificaciones de la página Analysis of traffic offenses	80
4.4.2	Modificaciones de la página Analysis of traffic offenses per group	87
4.4.3	Modificaciones de la página Analysis of traffic offenses per user	93
4.5	Apartado My data about the simulations y Analysis of Simulations	101
4.5.1	Modificaciones de la página My data about the Simulation	101
4.5.1.1	Modificaciones de la parte superior de la página My data about the Simulation	102
4.5.1.2	Modificaciones de la parte intermedia de la página My data about the Simulation	109
4.5.1.3	Modificaciones de la parte inferior de la página My data about the Simulation	115
4.5.2	Modificaciones de la página Simulations in scenario X	120
4.5.2.1	Modificaciones de la parte superior de la página Simulations in scenario X.	120
4.5.2.2	Modificaciones de la parte inferior de la página Simulations in scenario X.	127
4.5.3	Modificaciones de la página Simulation Data	132
5.	Manual de usuario de la plataforma web	145
5.1	Manual de usuario de la plataforma para el administrador	145
5.2	Manual de usuario de la plataforma para un usuario común	159
6.	Presupuesto económico	168

7. Conclusiones y líneas futuras	169
8. Bibliografía.....	171

Índice de figuras

Figura 1: Puesto del usuario del simulador ENTAC (Indra - b, 2019).....	28
Figura 2: Dimensiones de la cabina del simulador Gold Truck (TECNOTRON – a, 2020).	29
Figura 3: Puesto del usuario del simulador Simbio (LANDER - a, 2019).....	30
Figura 4 Dimensiones de la cabina del simulador Gold Bus (TECNOTRON – b, 2020).	31
Figura 5: Simulador SMARTSIM puesto del conductor (Indra – c, 2019).	32
Figura 6: Dimensiones de la cabina del simulador Silver (TECNOTRON - c, 2020)....	33
Figura 7: Simulador LANDER (LANDER - b, 2019).....	34
Figura 8: Dimensiones de la cabina del simulador Ambar (TECNOTRON – d, 2020). 35	
Figura 9: DriverSim (DriveSim, 2019).....	36
Figura 10: Virtual Driver Interactive (Virtual Driver Interactive, 2019).....	37
Figura 11: City Car Driving (City Car Driving, 2019).	38
Figura 12: Driver Test (Blog de la Auto escuela, 2019).....	39
Figura 13: Car Driving School Simulator (Google Play - a, 2019).	40
Figura 14: Dr. Driving 2(Google Play - b, 2019).	41
Figura 15: Interfaz de usuario de XAMPP.....	50
Figura 16: Interfaz de usuario de phpMyAdmin.	51
Figura 17: Estructura de la Base de datos después de la actualización.....	54
Figura 18: Modificaciones realizadas en el archivo rjsimulador.install y en el hook rjsimulador_schema().....	55
Figura 19: Modificaciones realizadas en el método loadPartidaById(\$id_partida) en el archivo DBDataProvider.php.	56
Figura 20: Modificaciones realizadas en el método savePartida(Partida \$partida) en el archivo DBDataSaver.php.....	56
Figura 21: Modificaciones realizadas en el archivo Partida.php.	57
Figura 22: Nuevo método getTipoVehiculo() creado en el archivo Partida.php.....	57
Figura 23: Nuevo método getTipoVehiculoNombre() creado en el archivo Partida.php.	58
Figura 24: Nuevo método setTipoVehiculo(\$tipo_vehiculo) creado en el archivo Partida.php.....	58
Figura 25: Página de inicio de sesión de la plataforma web.	59
Figura 26: Página de principal de la plataforma web antes de la actualización.	60
Figura 27: Página de principal de la plataforma web después de la actualización.	60

Figura 28: Página “Data analysis” de la plataforma web antigua, apartado “General data analysis”	62
Figura 29: Modificaciones realizadas en el hook rjsimulador_install() del fichero rjsimulador.install.	63
Figura 30: Página “Analysis of vehicle data” de la plataforma web, apartado “Simulation data”.....	64
Figura 31: Código incluido en la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el selector de tipo de vehículo.	64
Figura 32: Código incluido en la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos del selector de tipo de vehículo.....	65
Figura 33: Código incluido en la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el título en función del selector de tipo de vehículo.	65
Figura 34: Código incluido en la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el título de la gráfica.	66
Figura 35: Código modificado en la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos para el grafico y para pintar los datos en el gráfico.	66
Figura 36: Nueva función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) dentro del fichero UsuarioSimulacion.php.	67
Figura 37: Página “Data analysis” de la plataforma web antigua, apartado “Data analysis by group”.....	68
Figura 38: Página “Analysis of vehicle data” de la plataforma web, apartado “Simulation data per group”.....	69
Figura 39: Código incluido en la función generate_form_data_group_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el selector de tipo de vehículo.	69
Figura 40: Código incluido en la función generate_form_data_group_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos del selector de vehículos.....	70
Figura 41: Código incluido en la función generate_form_data_group_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el título en función del selector.	70
Figura 42: Código incluido en la función generate_form_data_group_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el título de la gráfica.	71
Figura 43: Código modificado en la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos para el gráfico.	71

Figura 44: Código modificado en la función <code>generate_form_data_general_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para pintar los datos en el gráfico.	71
Figura 45: Nueva función <code>retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo)</code> dentro del fichero <code>ListaUsuariosDataRetriever.php</code>	72
Figura 46: Página “Data analysis” de la plataforma web antigua, apartado “Data analysis by user”.	72
Figura 47: Página “Analysis of vehicle data” de la plataforma web, apartado “Simulation data per user”.	73
Figura 48: Código incluido en la función <code>generate_form_data_user_analysis ()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para generar el selector.	74
Figura 49: Código incluido en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para obtener los datos del selector de vehículos.....	74
Figura 50: Código incluido en la función <code>generate_form_data_user_analysis ()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para generar el título en función del selector.	75
Figura 51: Código incluido en la función <code>generate_form_data_user_analysis ()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para generar el título de la gráfica.	75
Figura 52: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para obtener los datos del usuario seleccionado.	75
Figura 53: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para pintar los datos del usuario seleccionado.	76
Figura 54: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para obtener los datos del resto de usuarios.....	76
Figura 55: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para pintar los datos del resto de usuarios.....	76
Figura 56: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para obtener los datos de los usuarios con la misma edad que el usuario a analizar.	77
Figura 57: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para pintar los datos de los usuarios con la misma edad que el usuario a analizar.	77
Figura 58: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms</code> , para obtener los datos de los usuarios con la misma experiencia conduciendo que el usuario a analizar.	77

Figura 59: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms</code> , para representar los datos de los usuarios con la misma experiencia conduciendo que el usuario a analizar.	78
Figura 60: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para obtener los datos de los usuarios con el mismo número de kilómetros recorridos que el usuario a analizar.	78
Figura 61: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para pintar los datos de los usuarios con el mismo número de kilómetros recorridos que el usuario a analizar.....	78
Figura 62: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para obtener los datos de los usuarios con la misma experiencia en videojuegos que el usuario a analizar.....	79
Figura 63: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para pintar los datos de los usuarios con la misma experiencia en videojuegos que el usuario a analizar.	79
Figura 64: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para obtener los datos para el grupo creado.	79
Figura 65: Código modificado en la función <code>generate_form_data_user_analysis()</code> dentro del fichero <code>rjsimulador.data_analysis.forms.inc</code> para representar los datos en el gráfico del grupo creado.....	80
Figura 66: Página “Infractions analysis” de la plataforma web antigua, apartado “General infraction analysis”.....	81
Figura 67: Página “Analysis of traffic offenses” de la plataforma web, apartado “Analysis of traffic offenses”.	82
Figura 68: Modificaciones realizadas en el hook <code>rjsimulador_install()</code> del fichero <code>rjsimulador.install</code>	83
Figura 69: Código incluido en la función <code>generate_form_infractions_general_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el selector.	83
Figura 70: Código incluido en la función <code>generate_form_infractions_general_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos del selector de vehículos.	84
Figura 71: Código incluido en la función <code>generate_form_infractions_general_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el título en función del selector de vehículos.....	84
Figura 72: Código incluido en la función <code>generate_form_infractions_general_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el título del gráfico.	85
Figura 73: Código modificado en la función <code>generate_form_infractions_general_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos para el grafico y para pintar los datos en el gráfico.....	85

Figura 74: Nueva función <code>getAverageInfraccionesByPartidaVehiculo(\$idInfraccion, \$idSimulation, \$vehiculo)</code> dentro del fichero <code>UsuarioSimulacion.php</code>	86
Figura 75: Nueva función <code>retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo(\$idInfraccion, \$idSimulation, \$vehiculo)</code> dentro del fichero <code>UsuarioSimulacion.php</code> . ..	86
Figura 76: Nueva función <code>retrieveAllInfraccionesByIdSimulacionVehiculo(\$idSimulation, \$vehiculo)</code> dentro del fichero <code>UsuarioSimulacion.php</code>	86
Figura 77: Página “Infractions analysis” de la plataforma web antigua, apartado “Infraction analysis by group”.....	87
Figura 78: Página “Analysis of traffic offenses” de la plataforma web, apartado “Analysis of traffic offenses per group”.....	88
Figura 79: Código incluido en la función <code>generate_form_infractions_group_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el selector. 89	89
Figura 80: Código incluido en la función <code>generate_form_infractions_group_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos del selector de vehículos.	89
Figura 81: Código incluido en la función <code>generate_form_infractions_group_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el título en función del selector.	90
Figura 82: Código incluido en la función <code>generate_form_infractions_group_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el título de la gráfica.....	90
Figura 83: Código modificado en la función <code>generate_form_infractions_group_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos para el gráfico.....	91
Figura 84: Código modificado en la función <code>generate_form_infractions_group_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para pintar los datos para el gráfico.	91
Figura 85: Nueva función <code>getAverageInfraccionesByPartidaVehiculo(\$idInfraccion,\$idSimulation,\$vehiculo)</code> dentro del fichero <code>ListaUsuariosDataRetriever.php</code>	91
Figura 86: Nueva función <code>retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo(\$idInfraccion, \$idSimulation,\$vehiculo)</code> dentro del fichero <code>ListaUsuariosDataRetriever.php</code>	92
Figura 87: Nueva función <code>retrieveAllInfraccionesByIdSimulacionVehiculo(\$idSimulation, \$vehiculo)</code> dentro del fichero <code>ListaUsuariosDataRetriever.php</code>	92
Figura 88: Página “Infractions analysis” de la plataforma web antigua, apartado “Infraction analysis by user”.	93
Figura 89: Página “Analysis of traffic offenses” de la plataforma web, apartado “Analysis of traffic offenses per user”.	94

Figura 90: Código incluido en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el selector.	95
Figura 91: Código incluido en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos del selector de vehículos.	95
Figura 92: Código incluido en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el título en función del selector.	95
Figura 93: Código incluido en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para generar el título de la gráfica.	96
Figura 94: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos del usuario seleccionado.	96
Figura 95: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para pintar los datos del usuario seleccionado.	96
Figura 96: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos del resto de los usuarios.	97
Figura 97: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para pintar los datos del resto de usuarios.	97
Figura 98: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos de los usuarios con la misma edad que el usuario a analizar.	97
Figura 99: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para pintar los datos de los usuarios con la misma edad que el usuario a analizar.	98
Figura 100: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos de los usuarios con la misma experiencia conduciendo que el usuario a analizar.	98
Figura 101: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para pintar los datos de los usuarios con la misma experiencia conduciendo que el usuario a analizar.	98
Figura 102: Código modificado en la función <code>generate_form_infractions_user_analysis()</code> dentro del fichero <code>rjsimulador.infractions_analysis.forms.inc</code> para obtener los datos de los usuarios con el mismo número de kilómetros recorridos que el usuario a analizar.	99

Figura 103: Código modificado en la función generate_form_infractions_user_analysis() dentro del fichero rjsimulador.infractions_analysis.forms.inc para pintar los datos de los usuarios con el mismo número de kilómetros recorridos que el usuario a analizar.	99
Figura 104: Código modificado en la función generate_form_infractions_user_analysis() dentro del fichero rjsimulador.infractions_analysis.forms.inc para obtener los datos de los usuarios con la misma experiencia en videojuegos que el usuario a analizar.	99
Figura 105: Código modificado en la función generate_form_infractions_user_analysis() dentro del fichero rjsimulador.infractions_analysis.forms.inc para pintar los datos de los usuarios con la misma experiencia en videojuegos que el usuario a analizar.	100
Figura 106: Código modificado en la función generate_form_infractions_user_analysis() dentro del fichero rjsimulador.infractions_analysis.forms.inc para obtener los datos para el grupo creado.	100
Figura 107: Código modificado en la función generate_form_infractions_user_analysis() dentro del fichero rjsimulador.infractions_analysis.forms.inc para pintar los datos para el grupo creado.	100
Figura 108: Página desde la que el administrador tiene acceso a los datos de las simulaciones de cada usuario.	101
Figura 109: Parte superior de la página “My data about the Simulation” antes de la actualización de la plataforma.	102
Figura 110: Parte superior de la página “My data about the Simulation” después de la actualización de la plataforma.	103
Figura 111: Código incluido en la función rjsim_simulaciones_upper_left_form() dentro del fichero rjsimulador.forms.inc para generar el selector de la gráfica “Total number of simulations fulfilled by the user”.....	103
Figura 112: Código incluido en la función rjsim_simulaciones_upper_left_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del selector de vehículos.	104
Figura 113: Código incluido en la función rjsim_simulaciones_upper_left_form() dentro del fichero rjsimulador.forms.inc para generar el título en función del selector.	104
Figura 114: Código de la función rjsim_simulaciones_upper_left_form() dentro del fichero rjsimulador.forms.inc que genera la gráfica.	105
Figura 115: Código modificado de la función rjsim_simulaciones_upper_left_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del usuario actual.	105
Figura 116: Nueva función getListasPartidasVehiculo(\$vehiculo) dentro del fichero Simulacion.php.	106
Figura 117: Nueva función ajax_partidas_vehiculo_callback() dentro del fichero rjsimulador.forms.inc.	106

Figura 118: Código incluido en la función <code>rjsim_simulaciones_upper_right_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> para generar el selector de la gráfica “Average number of traffic offenses”.....	106
Figura 119: Código incluido en la función <code>rjsim_simulaciones_upper_right_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> para obtener los datos del selector de vehículos.	107
Figura 120: Código incluido en la función <code>rjsim_simulaciones_upper_right_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> para generar el título en función del selector.	107
Figura 121: Código de la función <code>rjsim_simulaciones_upper_right_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> que genera la gráfica.	108
Figura 122: Nueva función <code>ajax_partidas_vehiculo_callback()</code> dentro del fichero <code>rjsimulador.forms.inc</code>	108
Figura 123: Parte intermedia de la página “My data about the Simulation” antes de la actualización de la plataforma.	109
Figura 124: Parte intermedia de la página “My data about the Simulation” después de la actualización de la plataforma.	110
Figura 125: Código incluido en la función <code>rjsim_simulaciones_upper_left_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> para generar el selector de la gráfica “Total number of simulations fulfilled by the user in the last X days”.	110
Figura 126: Código incluido en la función <code>rjsim_simulaciones_upper_left_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> para obtener los datos del selector de vehículos.	111
Figura 127: Código de la función <code>rjsim_simulaciones_upper_left_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> que genera la gráfica.	111
Figura 128: Código modificado de la función <code>rjsim_simulaciones_upper_left_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> para obtener los datos del usuario actual.	112
Figura 129: Nueva función <code>ajax_partidas_vehiculo_callback()</code> dentro del fichero <code>rjsimulador.forms.inc</code>	112
Figura 130: Código incluido en la función <code>rjsim_simulaciones_upper_right_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> para generar el selector de la gráfica “Average time to drive in each scenario”.	113
Figura 131: Código incluido en la función <code>rjsim_simulaciones_upper_right_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> para obtener los datos del selector de vehículos.	113
Figura 132: Código incluido en la función <code>rjsim_simulaciones_upper_right_form()</code> dentro del fichero <code>rjsimulador.forms.inc</code> que genera la gráfica.	114
Figura 133: Nueva función <code>ajax_infracciones_vehiculo_tiempo_partida()</code> dentro del fichero <code>rjsimulador.forms.inc</code>	114
Figura 134: Parte inferior de la página “My data about the Simulation” antes de la actualización de la plataforma.	115

Figura 135: Parte inferior de la página “My data about the Simulation” después de la actualización de la plataforma.	116
Figura 136: Código incluido en la función rjsim_simulaciones_main_form() dentro del fichero rjsimulador.forms.inc para generar el selector del gráfico “Percentages of traffic offences for all simulations”.....	117
Figura 137: Código incluido en la función rjsim_simulaciones_main_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del selector de vehículos.....	117
Figura 138: Código incluido en la función rjsim_simulaciones_main_form() dentro del fichero rjsimulador.forms.inc que genera el gráfico.	118
Figura 139: Código modificado de la función rjsim_simulaciones_main_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del usuario actual.	118
Figura 140: Nueva función retrieveAllPartidasVehiculo(\$vehiculo) dentro del fichero UsuarioSimulacion.php.	119
Figura 141: Nueva función ajax_partidas_vehiculo_callback() dentro del fichero rjsimulador.forms.inc.	119
Figura 142: Parte superior de la página “Partidas by Simulation X” antes de la actualización de la plataforma.	120
Figura 143: Parte superior de la página “Simulations in scenario X” después de la actualización de la plataforma.	121
Figura 144: Código incluido en la función rjsim_partidas_simulacion_upper_left_form() dentro del fichero rjsimulador.forms.inc para generar el selector de la gráfica “Comparison of average fuel consumption”.	121
Figura 145: Código incluido en la función rjsim_partidas_simulacion_upper_left_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del selector de vehículos.	122
Figura 146: Código de la función rjsim_partidas_simulacion_upper_left_form() dentro del fichero rjsimulador.forms.inc que genera la gráfica.	122
Figura 147: Código modificado de la función rjsim_partidas_simulacion_upper_left_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del usuario actual.....	123
Figura 148: Código modificado de la función rjsim_partidas_simulacion_upper_left_form() dentro del fichero rjsimulador.forms.inc para obtener los datos de todos los usuarios.....	123
Figura 149: Nueva función dataTodosConsumoMedio(\$id_simulacion,\$vehiculo,\$ultimasPartidas,\$listaPartidasUsuario) dentro del fichero rjsimulador.forms.inc para obtener los datos de todos los usuarios	123
Figura 150: Nueva función ajax_consumo_medio_vehiculo_callback() dentro del fichero rjsimulador.forms.inc.	124

Figura 151: Código incluido en la función rjsim_partidas_simulacion_upper_right_form() dentro del fichero rjsimulador.forms.inc para generar el selector de la gráfica “Average time to drive in scenario”.....	124
Figura 152: Código incluido en la función rjsim_partidas_simulacion_upper_right_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del selector de vehículos.....	124
Figura 153: Código en la función rjsim_partidas_simulacion_upper_right_form() dentro del fichero rjsimulador.forms que genera la gráfica.....	125
Figura 154: Código en la función rjsim_partidas_simulacion_upper_right_form() dentro del fichero rjsimulador.forms. para obtener los datos del usuario actual.....	125
Figura 155: Código en la función rjsim_partidas_simulacion_upper_right_form() dentro del fichero rjsimulador.forms. para obtener los datos de todos los usuarios.....	126
Figura 156: Nueva función dataTodosTiempoTotal(\$id_simulacion,\$vehiculo,\$ultimasPartidas,\$listaPartidasUsuario) dentro del fichero rjsimulador.forms.inc para obtener los datos de todos los usuarios	126
Figura 157: Nueva función ajax_consumo_medio_vehiculo_callback() dentro del fichero rjsimulador.forms.inc.....	126
Figura 158: Parte inferior de la página “Partidas by Simulation X” antes de la actualización de la plataforma.....	127
Figura 159: Parte inferior “Simulations in scenario X” después de la actualización de la plataforma.....	128
Figura 160: Nueva función getTipoVehiculoNombre() dentro del fichero Partida.php.....	129
Figura 161: Código incluido en la función rjsim_partidas_simulacion_main_form() dentro del fichero rjsimulador.forms.inc para generar el selector del gráfico “Traffic offenses in last simulations in the scenario”.....	130
Figura 162: Código incluido en la función rjsim_partidas_simulacion_main_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del selector de vehículos.....	130
Figura 163: Código de la función rjsim_partidas_simulacion_main_form() dentro del fichero rjsimulador.forms.inc que genera la gráfica.....	131
Figura 164: Código modificado en la función rjsim_partidas_simulacion_main_form() dentro del fichero rjsimulador.forms.inc para obtener los datos necesarios para el grafico.....	131
Figura 165: Nueva función ajax_infracciones_simulacion_vehiculo_callback() dentro del fichero rjsimulador.forms.inc.....	132
Figura 166: Primera tabla de la página “Partida Data” antes de la actualización de la plataforma.....	132

Figura 167: Código incluido en la función generarTablaResumenDatosPartida(Partida \$partida, ListaPartidas \$listaPartidasSimulacion) dentro del fichero rjsimulador.pages.inc para obtener el tipo de vehículo de una partida.	133
Figura 168: Código incluido en la función generarTablaResumenDatosPartida(Partida \$partida, ListaPartidas \$listaPartidasSimulacion) dentro del fichero rjsimulador.pages.inc para crear la nueva columna con los datos obtenidos.	133
Figura 169: Código incluido en la función generarTablaResumenDatosPartida(Partida \$partida, ListaPartidas \$listaPartidasSimulacion) dentro del fichero rjsimulador.pages.inc para vehículos de combustión.	134
Figura 170: Código incluido en la función generarTablaResumenDatosPartida(Partida \$partida, ListaPartidas \$listaPartidasSimulacion) dentro del fichero rjsimulador.pages.inc para vehículos eléctricos.	134
Figura 171: Primera tabla de la página “Simulation Data” después de la actualización de la plataforma.	134
Figura 172: Segunda tabla de la página Partida Data antes de la actualización de la plataforma.	134
Figura 173: Código incluido en la función rjsimulador_partida_page(\$id_partida) dentro del fichero rjsimulador.pages.inc para generar la tabla “Average RPMs per gear”.	135
Figura 174: Segunda tabla de la página “Simulation Data” después de la actualización de la plataforma.	135
Figura 175: Imagen que muestra el recorrido del usuario en la simulación de la página Partida Data antes de la actualización de la plataforma.	136
Figura 176: Nueva función getTipoInfraccion() dentro del fichero Infraccion.php.	136
Figura 177: Código de la función renderableArray() del archivo Infraccion.php para separar por colores el tipo de infracción.	137
Figura 178: Nueva función selectDataRecorrido() dentro del fichero rjsimulador.pages.inc.	138
Figura 179: Nueva función ajax_recorrido_datos() dentro del fichero rjsimulador.pages.inc.	139
Figura 180: Nueva función selectUserCompararRecorrido() dentro del fichero rjsimulador.pages.inc.	140
Figura 181: Nueva función selectUserCompararRecorrido() dentro del fichero rjsimulador.pages.inc.	140
Figura 182: Nueva función renderableArrayCompara() dentro del fichero DatoInstantaneo.php.	141
Figura 183: Nueva función renderableArrayCompara() dentro del fichero Infraccion.php.	142
Figura 184: Nueva función renderableArrayTitulo() dentro del fichero DatoInstantaneo.php.	143
Figura 185: Imagen que muestra el recorrido del usuario en la simulación de la página “Simulation Data” después de la actualización de la plataforma.	143

Figura 186: Tablas de la parte final de la página Partida Data antes de la actualización de la plataforma.	144
Figura 187: Tablas de la parte final de la página “Simulation Data” después de la actualización de la plataforma.	144
Figura 188: Página principal de la plataforma web cuando se logea un usuario como administrador.	145
Figura 189: Pestaña “Simulation data” de la página “Analysis of vehicle data” de la plataforma web.	147
Figura 190: Pestaña “Simulation data per group” de la página “Analysis of vehicle data” de la plataforma web.	149
Figura 191: Pestaña “Simulation data per user” de la página “Analysis of vehicle data” de la plataforma web.	151
Figura 192: Pestaña “Analysis of traffic offenses” de la página “Analysis of traffic offenses” de la plataforma web.	153
Figura 193: Pestaña “Analysis of traffic offenses per group” de la página “Analysis of traffic offenses” de la plataforma web.	155
Figura 194: Pestaña “Analysis of traffic offenses per user” de la página “Analysis of traffic offenses” de la plataforma web.	157
Figura 195: Pagina “Analysis of Simulation per user” de la plataforma web.	158
Figura 196: Parte inferior de la página “Simulation in scenario X” de la plataforma web.	159
Figura 197: Página de inicio de la plataforma.	159
Figura 198: Parte superior de la página “My data about the Simulation” de la plataforma web.	161
Figura 199: Parte inferior de la página “My data about the Simulation” de la plataforma web.	162
Figura 200: Parte superior de la página “Simulation in scenario X” de la plataforma web.	163
Figura 201: Parte inferior de la página “Simulation in scenario X” de la plataforma web.	164
Figura 202: Parte superior de la página “Simulation data” de la plataforma web.	165
Figura 203: Parte intermedia de la página “Simulation data” de la plataforma web.	165
Figura 204: Parte inferior de la página “Simulation data” de la plataforma web.	166

1. Introducción

La siguiente sección se centra en realizar una introducción a este documento. Hablaremos tanto de la motivación que tiene este proyecto, como de los objetivos que se esperan alcanzar y comentaremos tanto los medios utilizados, como los pasos seguidos para la realización de este trabajo.

Por otro lado, también se hablará del orden y estructura del documento, tratando de manera breve los temas de los que se habla en cada capítulo.

1.1 Motivación del proyecto

Teniendo en cuenta que la simulación y la realidad virtual están en auge, debido a los grandes avances tecnológicos que se están llevando a cabo en la actualidad, estos avances tecnológicos están haciendo que estas tecnologías nos lleven a vivir experiencias cada vez más realistas. Al ofrecer experiencias muy parecidas a la realidad, permiten a los usuarios vivir situaciones que en un entorno real podrían producir algún tipo riesgo para ellos mismos o para otros usuarios.

En nuestro caso, las simulaciones se centran en la conducción de distintos tipos de vehículos, diferenciando entre turismo con motor de combustión, turismo con motor eléctrico, autobús con motor de combustión, autobús con motor eléctrico, camión con motor de combustión y camión con motor eléctrico, en los distintos escenarios implementados en el simulador.

Este Trabajo de Fin de Grado en lo que se centra es en los datos de las distintas simulaciones de conducción y su análisis por medio de una plataforma web que presentará los datos de manera ordenada facilitando su estudio. La actualización que vamos a llevar a cabo nos permitirá analizar los datos de una manera más eficiente, así como diferenciar mejor distintos supuestos y mejorar la experiencia para el usuario.

Los datos que se recaban de las simulaciones pueden ser útiles para evaluar distintos factores como las infracciones de seguridad cometidas por el conductor del vehículo que pueden implicar situaciones de riesgo para él o para otras personas, infracciones de eficiencia que generan un exceso en el consumo normal del vehículo, datos sobre el consumo del vehículo en los distintos escenarios de simulación en función de la edad del conductor y la experiencia, tiempo en realizar un recorrido y otros. Todos estos datos pueden ser útiles en un futuro para empresas del sector del automóvil, ya que pueden aplicar mejoras a sus nuevos diseños intentando evitar los factores que motivan que se produzcan cierto tipo de situaciones en la vida real, que pueden perjudicar al usuario tanto físicamente como económicamente.

Desde otro punto de vista, estos datos también pueden ser útiles para los propios usuarios que realizan las simulaciones, puesto que pueden analizar los errores cometidos y que podrían haber evitado modificando su forma de conducir. Esto les puede ayudar a intentar evitar situaciones de riesgo, mejorar su forma de

conducir haciéndola más eficiente o compararse con otros usuarios, para saber si su comportamiento es normal (R.J.Corchero, 2016).

1.2 Objetivos del proyecto

Este Trabajo de Fin de Grado se centra en actualizar y adecuar a las necesidades de los usuarios y a las nuevas características del simulador de conducción, la plataforma web que actualmente almacena los datos de las simulaciones de los usuarios en el simulador de conducción.

La actualización se ha llevado a cabo con la idea principal de mantener el formato y la estructura actual de la misma, pero añadiendo nuevas características que faciliten el análisis de los datos para cierto tipo de situaciones.

En la actualidad esta plataforma está construida sobre el CMS Drupal 7, del que aprovecha ciertas características muy útiles para nuestra plataforma, como son los módulos creados por la comunidad Drupal. Utiliza una API (*Application Programming Interface*) REST (*Representational State Transfer*), que es el medio de comunicación con el simulador de conducción a través de la recepción de datos en formato JSON. Como eje central de la plataforma está el módulo Drupal que contiene el código de la misma, que es sobre el que se va a centrar el Trabajo de Fin de Grado (R.J.Corchero, 2016).

1.3 Fases del proyecto

Para poder llevar a cabo este trabajo de fin de grado basado en mejorar y actualizar una plataforma web para almacenar los datos de un simulador de conducción, ha sido necesario seguir una serie de pasos bastante diferenciados.

- Estudio previo al desarrollo: en esta etapa se comenzó analizando el CMS que tiene como base esta plataforma. Se optó por instalar Drupal de manera local y así poder estudiar el funcionamiento de este CMS apoyándose en la documentación del mismo. Después de la toma de contacto con Drupal, se procedió a analizar el propio código de la plataforma que se iba a tomar como punto de partida. Este análisis facilitó conocer el funcionamiento del código y permitió plantear el futuro desarrollo que se iba a llevar a cabo sobre este código.
- Desarrollo de la plataforma: una vez que se analizó el código como se comentó en el anterior apartado y tomando como punto de partida los requisitos funcionales que se habían propuesto para este proyecto, se comenzó con el desarrollo del mismo. Cuando se comenzó el desarrollo buscando obtener los resultados esperados, fue necesario seguir apoyándose en documentación en la medida de lo posible, para poder cumplir los requisitos propuestos.

- Documentación del desarrollo: cuando se finalizó el desarrollo se comenzó a realizar el documento actual, con la intención de documentar el trabajo realizado y las modificaciones realizadas en la plataforma. Este documento busca facilitar a los usuarios de la plataforma y a futuros desarrolladores de la misma a comprender que se ha buscado con este desarrollo y como se ha realizado. Para ello en este documento se ha incluido secciones que se han centrado en el proceso de desarrollo realizado y que explican el funcionamiento de la misma.

1.4 Medios utilizados (hardware y software)

Los medios que hemos utilizado los vamos a dividir en dos categorías diferenciadas: hardware y software.

1.4.1 Hardware

Para el desarrollo de este trabajo de fin de grado, no ha sido necesario utilizar un hardware específico, ya que este proyecto se ha centrado en el desarrollo de una plataforma web.

El equipo que se ha utilizado para realizar el desarrollo es un portátil ASUS A55V. Este portátil cuenta con las siguientes características:

- Procesador: Intel Core i7-3630QM CPU @ 2.40GHz
- Memoria RAM: 8.00 GB
- Disco duro: 500GB
- Tarjeta Gráfica: NVIDIA GeForce 610M 2GB

Por otro lado, también ha sido necesario utilizar periféricos para facilitar el trabajo y hacerlo más cómodo. Los periféricos utilizados han sido los siguientes:

- Un teclado de la marca Lenovo
- Un ratón de la marca Logitech
- Un monitor de 27 pulgadas de la marca LG

1.4.2 Software

Como se ha comentado en el apartado anterior, al tratarse este proyecto de un desarrollo de una plataforma web, los recursos necesarios han sido mayoritariamente software. A continuación, se irán hablando de ellos y de sus características más relevantes:

- Sistema operativo: del que dispone el portátil mencionado en el apartado anterior es el sistema Windows 8.1.

- Entorno de desarrollo integrado (IDE) PhpStorm: el IDE PhpStorm creado por la empresa JetBrains ha sido el entorno de desarrollo integrado elegido (IDE, *Integrated Development Environment*) ya que aporta la posibilidad de trabajar con una licencia de estudiante, aunque se trata de un IDE que no es gratuito. Por otro lado, es un IDE muy versátil ya que se puede trabajar en distintos lenguajes de programación como PHP, que es el lenguaje de programación en el que está desarrollado el módulo Drupal de la plataforma, HTML y JavaScript. También aporta características muy útiles como prevención de errores sobre la marcha, autocompletar y la posibilidad de utilizar un sistema de control de versiones como GitHub (PhpStorm, 2019).

- Notepad++: es un editor de código fuente libre que es compatible con muchos lenguajes de programación. Este programa tiene soporte para Microsoft Windows. Este programa es muy útil y fácil de usar, ya que la interfaz de usuario es muy similar al Bloc de notas de Windows, pero interpreta muchos lenguajes de programación y, además, es un programa muy ligero que no influye en exceso en el rendimiento de la CPU. En este proyecto se ha utilizado como soporte para poder ver distintas versiones del código de la plataforma a la vez, comparar distintos ficheros, y modificar ficheros de manera rápida para hacer comprobaciones de funcionamiento (Notepad++, 2019).

XAMPP v3.2.2: es un programa que implementa una distribución Apache que tiene una serie de características muy interesantes, como que es completamente gratuito. Por otro lado, es un programa que es muy fácil de instalar, utilizar y configurar. Es uno de los mejores programas para desarrolladores principiantes (XAMPP, 2019).

- Postman: es una herramienta muy útil en entornos de desarrollo de API, ya que con ella podemos probar y comprobar el funcionamiento de un servicio REST. Esta ha sido una de las herramientas más importantes para el desarrollo de las nuevas características de la plataforma, ya que ha permitido incluir los nuevos datos que caracterizan las nuevas características de la plataforma en la base de datos. Otra gran ventaja es que la manera de almacenar los nuevos datos en la plataforma ha sido simulando los envíos de datos del simulador contra el servicio REST de la plataforma. Por lo tanto, ha facilitado probar la implementación de las nuevas características en el entorno local, como si se trata del entorno real donde la plataforma funcionara en un futuro (Postman, 2019).

- Mozilla Firefox: es un navegador de código abierto. Es un navegador web libre que es compatible con sistemas operativos como Microsoft

Windows, Linux, Android, IOS y macOS. Es un navegador muy versátil y consume menos recursos del ordenador en el que se trabaja con él como, por ejemplo, otros navegadores como Google Chrome.

Ha sido el navegador web elegido para trabajar en el desarrollo de la plataforma, hacer pruebas de funcionamiento y comprobar los cambios de estructura en la plataforma ya que, como se ha comentado antes, al consumir menos recursos del ordenador que otros navegadores, permite trabajar con todos los programas que se han utilizado para el desarrollo de las nuevas características de la plataforma sin tener problemas de rendimiento (Mozilla Firefox, 2019).

1.5 Estructura de la memoria

En este apartado se hablará del esquema que se ha seguido para desarrollar este documento y el porqué de cada sección expuesta en él.

- Introducción: este es el apartado actual. En él se intenta aportar una visión general del proyecto realizado, los pasos que se han llevado a cabo para realizarlo y los medios empleados para poder desarrollarlo.
- Simuladores de conducción para la formación vial de usuarios: como el desarrollo de este proyecto se ha centrado en modificar una plataforma web que almacena los datos de un simulador de conducción y este simulador está enfocado a la formación vial, resultado interesante comentar también qué otros simuladores existen para este propósito en el mercado y sus características más relevantes.
- Plataforma web: posibilidades de desarrollo: en este apartado se ha buscado presentar las posibilidades tecnológicas que existen actualmente para desarrollar una plataforma web. Se hablará de qué son los *Frameworks* y los Gestores de contenido, las posibilidades que ofrecen, cuales son los más utilizados actualmente y el porqué de la elección en su momento del CMS Drupal 7 para el desarrollo de la plataforma.
- Desarrollo de la plataforma: se comentará cómo se ha realizado en la plataforma, explicando de manera detallada los pasos que se han seguido y el porqué de los mismos.
- Manual de usuario de la plataforma web: como se han realizado en la plataforma bastantes modificaciones relevantes, se ha pensado que sería interesante realizar un pequeño manual de usuario que sirva como toma de contacto con la plataforma para posibles futuros usuarios.

- Presupuesto económico: en este apartado se analiza cuál sería el coste de desarrollo de este proyecto en un entorno laboral real, enfocando este análisis al sector de la consultoría.
- Conclusiones y líneas futuras: como el mismo nombre del apartado expresa, en él se realizan una serie de reflexiones sobre el proceso de desarrollo de la plataforma, así como cuál sería el posible enfoque de futuros desarrollos sobre esta plataforma web.
- Bibliografía: en esta sección se presenta toda la documentación utilizada como soporte para poder realizar el desarrollo de este proyecto.

2. Simuladores de conducción para la formación vial de usuarios

El eje principal sobre el que gira este Trabajo de Fin de Grado es el simulador de conducción que almacena los datos de las simulaciones en la plataforma web.

Por este motivo, es interesante comentar la importancia que están cogiendo este tipo de tecnologías en nuestra vida diaria.

Los simuladores de conducción son herramientas muy útiles para familiarizarse con situaciones reales que se pueden dar en un día normal en la carretera, sin tener que salir de un entorno de confort y seguridad.

En el caso de las autoescuelas, puede ser una tecnología muy útil de cara a que los alumnos puedan aprender las normas de seguridad en escenarios realistas, antes de ponerse al frente de un vehículo en una carretera. Otro de los puntos fuertes de los simuladores para el caso de la aplicación de los mismos en las autoescuelas, es la capacidad de exponer a los alumnos a situaciones de riesgo, pero sin ningún peligro. Estas situaciones nunca se producirán realizando prácticas de conducción reales con el instructor, pero sí se pueden dar cuando el alumno obtenga el permiso de conducción y pueda conducir libremente por la ciudad. Estas prácticas usando la realidad virtual pueden preparar al futuro conductor para reaccionar ante estas situaciones de riesgo y así evitar peligros para el mismo o para otros usuarios de la vía. Estas situaciones pueden ser, por ejemplo, reaccionar ante un accidente, conducir sobre pavimento deslizante o que el usuario sea consciente de las consecuencias de la pérdida de atención al utilizar el móvil, entre otras. Los simuladores nunca podrán sustituir a las prácticas de conducción en la vida real, pero pueden ser usados como complemento para reforzar el conocimiento y mejorar el aprendizaje. (Cuatro, 2019)

Los simuladores también son una herramienta muy útil para los conductores profesionales, ya que les permiten mejorar sus actitudes de conducción. Entre estos conductores profesionales se encuentran los camioneros, autobuseros, taxistas, policías, etc. Estas personas pasan la mayor parte de su jornada laboral circulando por la vía pública, por lo que la probabilidad de estar expuestos a situaciones de riesgo es mayor que un usuario normal. Gracias a los simuladores, los conductores pueden realizar prácticas en situaciones extremas que se pueden dar en su día a día y que son muy difíciles de simular en un entorno controlado real. De esta manera, pueden mejorar su reacción natural ante determinados supuestos, mejorar sus reflejos y prevenir posibles riesgos para ellos mismos o para los usuarios de los que son responsables de su seguridad en determinadas circunstancias. (DGT, 2019)

Para el área de investigación y desarrollo de los principales fabricantes de vehículos, los simuladores de conducción son muy útiles y por eso los llevan usando desde hace años. Los simuladores de conducción les permiten analizar ciertas características de conducción de los usuarios de sus vehículos. De esta manera, pueden desarrollar tecnologías de seguridad activa y sistemas de asistencia a la conducción para prevenir riesgos ante situaciones como fatiga, somnolencia al volante, conducción bajo los efectos de sustancias estupefacientes o enfermedad, entre muchas otras. (DGT, 2019)

A continuación, se realiza una pequeña clasificación de una serie de simuladores de conducción enfocados a la formación vial de los usuarios, según sus características y el

entorno de utilización para el que están diseñados. Esta lista se centra en los simuladores que simulan los tipos de vehículos que aparecen en la plataforma web desarrollada en este Trabajo Fin de Grado (R.J.Corchero, 2016).

2.1 Simuladores de conducción de uso profesional con experiencia inmersiva

Cuando nos referimos a simuladores de conducción de uso profesional con experiencia inmersiva, hablamos de los que cuentan con los controles reales de un vehículo que proporcionan una conducción realista, cuentan con un asiento integrado, un puesto de conducción con un entorno similar al de un vehículo de verdad, sonido envolvente, etc.

Todas estas características ayudan a que el usuario se sienta dentro de un vehículo de verdad y permiten que la experiencia sea más realista. La denominación de uso profesional de este tipo de simuladores se debe a que, debido a su elevado coste, suelen ser utilizados en centros de formación o entrenamiento de conductores con un carácter profesional.

Los mencionados a continuación son una serie de simuladores de uso profesional con experiencia inmersiva de importantes empresas dedicadas al desarrollo de simuladores y que se pueden encontrar en el mercado.

2.1.1 ENTAC - Simulador de conducción de vehículos pesados

El simulador de conducción de vehículos pesados ENTAC ha sido desarrollado por la empresa Indra para poder ser configurado y realizar simulaciones de distintos tipos de vehículos pesados como camiones, autobuses, vehículos 4x4, vehículos de uso militar, etc. Este simulador tiene la capacidad de simular más de veinte vehículos distintos.

El puesto de conducción para el usuario es genérico, lo que le hace ser similar a muchos tipos de vehículos pesados y que el usuario se sienta en el propio vehículo mientras realiza la simulación.

El simulador está equipado con un sistema visual formado por cuatro pantallas full HD de cincuenta y cinco pulgadas y una resolución total de 7680x1080.

Está compuesto por una cabina semicerrada con techo y plataforma de acceso que aporta mayor realismo a la hora de simular la conducción de un vehículo real, como se puede apreciar en la figura 1.



Figura 1: Puesto del usuario del simulador ENTAC (Indra - b, 2019).

Este simulador dispone de modo cooperativo entre simuladores, algo que puede ser interesante de cara a entrenamientos de conducción en caravana para varios usuarios, ofreciendo la posibilidad de realizar varios usuarios simulaciones en el mismo escenario y al mismo tiempo, lo que les servirá de entrenamiento para cuando en un futuro deban circular de manera conjunta con sus vehículos por un entorno real.

También dispone de la posibilidad de elegir el tipo del vehículo según la forma y lo que implica esto para la conducción, pudiendo elegir entre vehículos rígidos, vehículos articulados y tren de carretera. Por otro lado, también cuenta con la posibilidad de diferenciar los sistemas de tracción entre 4x2, 4x4, 6x4 y 6x2.

El puesto del instructor está equipado para poder controlar simultáneamente varios simuladores a la vez con la posibilidad de comunicarse con los usuarios por voz. De este modo, el instructor puede controlar el proceso de formación de varios usuarios a la vez, cambiar el escenario de simulación o el tipo de vía, si lo cree oportuno en cada momento (Indra - a, 2019) (Indra - b, 2019).

2.1.2 Simulador de camión Gold Truck

El simulador *Gold Truck* ha sido desarrollado por la empresa TECNOTRON. Es un simulador de camiones exclusivamente, pero con la posibilidad de simular nueve tipos de vehículos pesados en una misma cabina. Los tipos de vehículos pesados que se pueden simular en esta cabina son, por ejemplo, camión rígido, camión articulado, camión biarticulado, camión cisterna, camión *kenworth*, entre otros.

Esta cabina busca fomentar la conducción eficiente y responsable dentro de un ambiente controlado sin riesgos, con un máximo rendimiento de tiempo y espacio.

Este simulador de conducción de vehículos pesados fomenta habilidades en los usuarios, como aumento de la sensación de confianza, reaccionar

correctamente ante ciertas situaciones, mejorar la conducción bajo condiciones adversas, etc.

Este simulador cuenta en su cabina con un salpicadero de camión real, con todos los controles necesarios para controlar un vehículo real. Cuenta con manetas y volante de un vehículo real que aporta realismo a la simulación. Tiene una columna de dirección con *Force feedback* integrado. Este sistema transmite al usuario las vibraciones que se producirían al recorrer el escenario con un vehículo real, aportando mayor sensación de realismo a la simulación. También dispone de una peladera real con el pedal de freno dotado de un sistema de vibración para emular el sistema ABS de un vehículo real.

También dispone de una plataforma de movimiento con tres grados de libertad que permite reproducir movimientos de aceleración, frenadas, paso por curvas, vibración del terreno y movimientos provocados por el nivel de carga.

Tiene un campo de visión de ciento cincuenta grados y punto de vista dinámico, gracias a tres pantallas LED de cuarenta y tres pulgadas y un panel de control de quince pulgadas. También incorpora un sistema de sonido envolvente 5.1 que hace la experiencia mucho más real para el usuario.

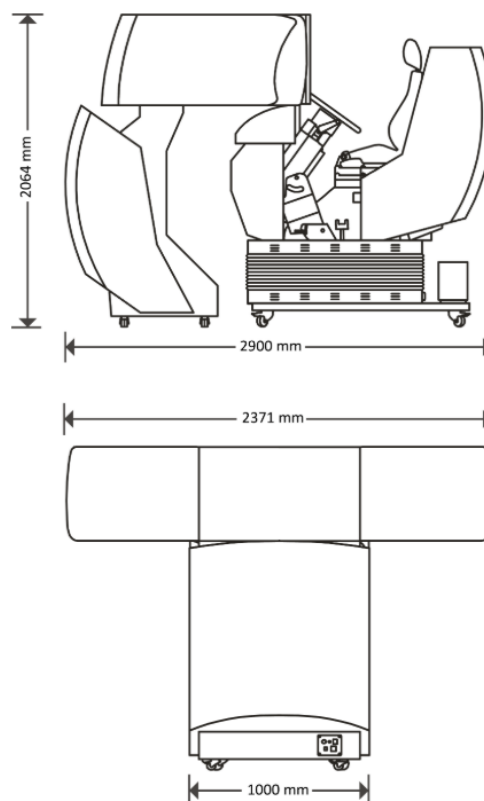


Figura 2: Dimensiones de la cabina del simulador Gold Truck (TECNOTRON – a, 2020).

Este simulador, a pesar de su pequeño tamaño como se puede apreciar en la figura 2, aporta muchas funcionalidades para la formación de futuros conductores y conductores habituales que lo pueden usar para mejorar sus competencias de conducción (TECNOTRON – a, 2020).

2.1.3 Simulador de Vehículos Pesados – Simbio

El simulador Simbio de la empresa Lander cuenta con un puesto de conducción muy realista, con todo lujo de detalles para aproximarse lo máximo a lo que sería un vehículo real.

Estas características aportan mayor realismo a la simulación. El puesto del conductor cuenta con un volante que puede ser regulado al gusto del usuario, al igual que el asiento, llave de contacto, palancas de intermitentes y limpiaparabrisas, cinturón de seguridad, palanca de freno de estacionamiento, botones de apertura y cierre de puertas para los casos en que se simula la conducción de autobuses, etc.

El puesto de conducción cuenta con 3 monitores de 55 pulgadas que aportan una visión de 180 grados, como se puede ver en la figura 3. Esta disposición de las pantallas aporta la sensación de visión circular de un vehículo real y facilita la toma de medidas distancia de referencia durante la simulación. Este simulador dispone de un sistema de movimiento de tres grados de libertad gracias al cual los usuarios pueden experimentar, por ejemplo, las irregularidades del terreno, paso por una curva, efectos de la aceleración y del frenado, etc.



Figura 3: Puesto del usuario del simulador Simbio (LANDER - a, 2019).

El puesto del instructor cuenta con muchas funcionalidades que son muy útiles para la formación de usuarios como, por ejemplo, la posibilidad de impresión de un informe final con los resultados de la sesión, diseño de ejercicios de simulación, capacidad para introducir averías y modificar las condiciones climáticas, etc (LANDER - a, 2019).

2.1.4 Simulador de autobús Gold Bus

Es un producto de la empresa TECNOTRON diseñado para la simulación didáctica que permite simular siete modelos de autobuses entre los que están, microbús, autobús urbano, autobús urbano articulado, autobús urbano biarticulado, autobús rígido interurbano y otros.

Al igual que otros productos de esta empresa, busca fomentar la conducción eficiente y responsable dentro de un ambiente controlado sin riesgos, con un máximo rendimiento de tiempo y espacio.

Este simulador de conducción al igual que otros simuladores de la misma compañía fomenta habilidades en los usuarios, como aumento de la sensación de confianza, reaccionar correctamente ante ciertas situaciones, mejorar la conducción bajo condiciones adversas, etc.

Cuenta con gran número de detalles que hacen la experiencia más realista, dispone de un salpicadero de autobús real con los típicos comandos para controlar la apertura y cierre de puertas del vehículo, el volante y las maneras exactamente iguales que las de un vehículo real.

Dispone de una columna de dirección con *Force feedback* integrado, también dispone de una pedalera real con el pedal de freno dotado de un sistema vibración para emular el sistema ABS de un vehículo real.

Cuenta con una plataforma de movimiento con tres grados de libertad, que permite reproducir movimientos de aceleración, frenadas, paso por curvas, vibración del terreno y movimientos provocados por el nivel de carga.

Tiene un campo de visión de ciento cincuenta grados y punto de vista dinámico, gracias a tres pantallas LED de cuarenta y tres pulgadas y un panel de control de quince pulgadas.

También incorpora un sistema de sonido envolvente 5.1 que hace la experiencia mucho más real para el usuario.

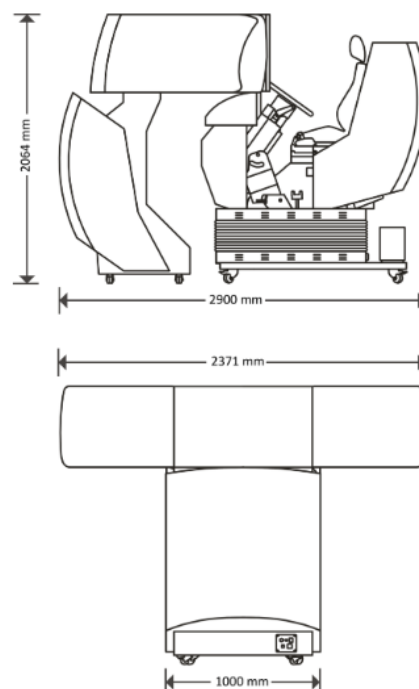


Figura 4 Dimensiones de la cabina del simulador Gold Bus (TECNOTRON – b, 2020).

Este simulador comparte muchas características con otros simuladores creados por la misma compañía. Al igual que otros simuladores de la compañía TECNOTRON, a pesar de su pequeño tamaño, como se puede ver

en la figura 4, aporta muchas funcionalidades que son muy útiles para la formación de conductores (TECNOTRON – b, 2020).

2.1.5 SMARTSIM - Simulador de conducción de automóvil

Este simulador de conducción de automóviles desarrollado por la empresa Indra ha sido galardonado con numerosos premios, gracias a sus capacidades formativas y su gran realismo.

Tiene un diseño robusto que facilita su transporte y su instalación, perfecto para centros de formación de conductores o autoescuelas. Dentro de este diseño robusto se encuentran las últimas tecnologías en hardware y software que existen actualmente en el mercado.

El puesto de conducción como se observa en la figura 5 cuenta con muchas características que aportan gran realismo a la simulación como, por ejemplo, un cuadro de mandos completo y retro iluminado en función del modelo de vehículo y del modo diurno o nocturno, llave de contacto, palanca de intermitentes y de limpiaparabrisas, freno de mano, claxon, regulador de espejos, etc.

Otra de las muchas características interesantes que tiene este simulador es la capacidad de adecuación con distintos entornos y señalizaciones, con transmisión automática o manual. Aporta la posibilidad de que el usuario sea deslumbrado por el sol y luces de otros vehículos en la noche. A todo esto, hay que añadir unos escenarios muy realistas con peatones.



Figura 5: Simulador SMARTSIM puesto del conductor (Indra – c, 2019).

Por todas estas características y muchas más, este simulador es uno de los mejores simuladores para la formación de distintos tipos de usuarios. Las posibilidades didácticas que aporta son casi ilimitadas como, por ejemplo, formación de nuevos conductores, protección de riesgos laborales en la conducción, conducción eficiente, prevención de accidentes, formación de cuerpos de seguridad, etc (Indra – c, 2019).

2.1.6 Silver - Simulador de automóvil de ALTA inmersión

El simulador *Silver* de la empresa TECNOTRON es un simulador con plataforma móvil que está especialmente diseñado para alumnos avanzados, para cursos de prevención de riesgos en la vía pública y concienciación vial. La idea principal con la que se desarrolló este simulador es familiarizar a los usuarios con todos los aspectos de un vehículo real y así generar confianza a la hora de enfrentarse a situaciones reales.

Este equipo dispone de una cabina similar a la de un vehículo real contando con panel de controles realista, asiento, pedales de turismo real con motores de vibración integrados para recrear los efectos del sistema ABS, manetas y una columna de dirección con *Force Feedback*.

Dispone de una plataforma de movimiento de dos ejes que reproduce los movimientos de aceleraciones, frenadas, paso por curva y cambio de terreno como se producen en un vehículo real, las dimensiones de esta cabina se pueden apreciar en la figura 6.

Para recrear mejor la realidad y obtener una experiencia más inmersiva, cuenta con un campo visual de ciento treinta y cinco grados, formado por tres pantallas de treinta y dos pulgadas y un panel de control de diez pulgadas. Además, para mejorar la experiencia, dispone de un sistema de sonido envolvente 5.1 que hace que los usuarios se sumerjan más en la simulación.

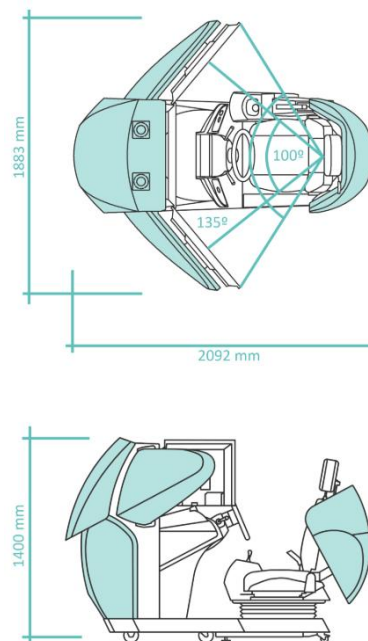


Figura 6: Dimensiones de la cabina del simulador Silver (TECNOTRON - c, 2020).

La cabina del simulador está formada por una carcasa de fibra y una estructura modular, que facilita su transporte y su instalación en lugares con espacio reducido (TECNOTRON - c, 2020).

2.1.7 El simulador de coche LANDER

Este simulador se ha diseñado con controles reales de un vehículo ligero, con un puesto de conducción que evoca las líneas y contornos de este tipo de vehículos. Este conjunto de características aporta una sensación de conducción más realista que permite sentirse dentro de un automóvil independientemente del que estemos conduciendo.



Figura 7: Simulador LANDER (LANDER - b, 2019).

El simulador de coche fabricado por LANDER es el que incorpora el mejor sistema visual de los comentados. Esto se debe a que incorpora tres pantallas de cincuenta y cinco pulgadas 4K *Curved*, como se observa en la figura 7, aportando al usuario un ángulo de visión de ciento ochenta grados con una imagen muy realista.

Por otro lado, este simulador permite la conducción por la derecha o por la izquierda, con las necesidades adaptativas de software y hardware. También cuenta con la posibilidad de simular vehículos con cambio manual o automático.

Teniendo en cuenta que los movimientos que realiza un vehículo son un factor importante a la hora de conducirlo, este simulador incorpora un sistema de movimiento con tres grados de libertad, que permite reproducir con gran realismo los movimientos de aceleración, frenado, paso por curva y muchos otros.

Una de las características más reseñables de este simulador es que el sistema software se hace a medida en función de las necesidades de cada cliente. Esto hace que simulador sea único y la experiencia sea más realista. (LANDER - b, 2019)

2.1.8 Ambar - Simulador de automóvil de media inmersión

Este simulador de la empresa TECNOTRON es el más sencillo en comparación con todos de los que se ha hablado con anterioridad, pero es

válido para seguir considerándolo uno más de la categoría de simuladores de uso profesional con experiencia inmersiva.

Está diseñado para facilitar la primera toma de contacto con los usuarios, pero aportando todos los detalles de un vehículo real y así mejorar la experiencia del usuario.

Aunque se trata del simulador más sencillo de esta categoría, cuenta con muchos detalles que hacen la experiencia más realista, como una columna de dirección *Force Feedback* y un volante de un vehículo real, al igual que otros simuladores de esta compañía. Integrados en la columna de dirección, dispone de bocina, intermitentes, luces y limpiaparabrisas.

La experiencia no es tan inmersiva como con otros simuladores de esta categoría, pero dispone de un ángulo de visión de ciento treinta y cinco grados, formado por tres monitores LED HD de veinticuatro pulgadas. Por otro lado, dispone de un sistema de sonido 2.0 con salida de auriculares, por si el usuario quiere sumergirse más en la simulación.

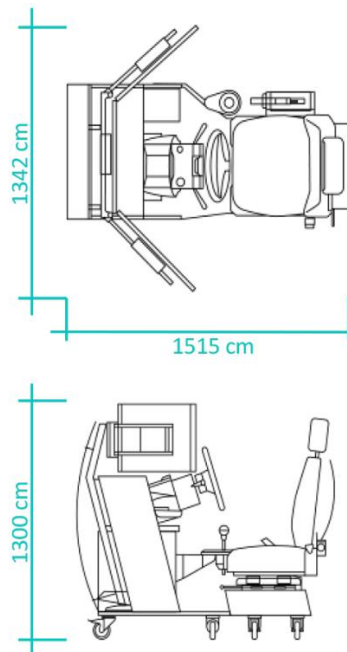


Figura 8: Dimensiones de la cabina del simulador Ambar (TECNOTRON – d, 2020).

Su sencillez también es un punto a favor, ya que con su estructura ligera de madera y metal y con su diseño ergonómico que permite plegar las pantallas laterales, el simulador puede pasar a través de puertas a la hora de ser transportado (TECNOTRON – d, 2020).

2.2 Simuladores de conducción de uso profesional para ordenador

Los simuladores de conducción de uso profesional para ordenador son aquellos que no cuentan con soporte hardware para hacer la simulación más realista. El

fabricante de estos simuladores solo distribuye el software que se instala en el ordenador del cliente.

Por otro lado, el cliente puede añadir periféricos al ordenador que tiene el software del simulador instalado y así hacer la experiencia más realista. La denominación de uso profesional de este tipo de simuladores se debe a que, debido a su elevado coste, suelen ser utilizados en centros de formación o entrenamiento de conductores con un carácter profesional.

A continuación, se habla de dos simuladores de uso profesional para ordenador de los muchos que se pueden encontrar en el mercado.

2.2.1 DriverSim

DriverSim es un simulador de conducción para ordenador dirigido a las autoescuelas y centros de formación de conductores. Este software ha sido especialmente diseñado como una herramienta educativa para conductores. Consta de distintas funcionalidades dependiendo del objetivo de aprendizaje que se quiera conseguir:

- Aprender a conducir: destinado a alumnos que tengan una primera toma de contacto con la conducción, que les permitirá poner en práctica todo lo aprendido en las sesiones teóricas.
- Cursos de amaxofobia: enfocados a usuarios que tienen miedo a conducir, permitiéndoles aprender a reaccionar ante determinadas circunstancias sin estar expuestos a ningún riesgo.
- Cursos de prevención de riesgos laborales: donde los usuarios aprenden a prevenir accidentes cuando circulan por la vía pública.
- Eventos de seguridad vial: este software es una excelente herramienta formativa en el campo de la seguridad vial.



Figura 9: DriverSim (DriveSim, 2019).

Algunas de las características más relevantes que tiene este simulador son, tráfico real y peatones, configuración climatológica, configuración de horario,

configuración de adherencia, formación inicial en pista, diferenciación de vehículos en función de su combustible y el sistema de tracción, etc. En la figura 9 se puede ver lo que aprecia un usuario de esta aplicación.

Todas estas características son las que hacen a este simulador muy reconocido y valorado (DriveSim, 2019).

2.2.2 Virtual Driver Interactive

Virtual Driver Interactive es otro simulador de conducción para ordenador dirigido a las autoescuelas y centros de formación de conductores, cuenta con distintos modos de funcionamiento que refuerzan ciertos puntos de la conducta de los conductores que mejoran su manera de conducir:

- Detección virtual de peligros: es una sesión inmersiva de sesenta minutos que permite al conductor aprender a identificar posibles peligros que se pueden encontrar en la vida, con la posibilidad de adecuar el vehículo y los escenarios haciéndolos similares a los de su día a día.
- Una simple decisión: permite a los futuros conductores experimentar los peligros de conducir distraído.



Figura 10: Virtual Driver Interactive (Virtual Driver Interactive, 2019).

- Driving Essentials: es un simulador de conducción dedicado a futuros conductores donde podrán aprender conductas de conducción segura. En la figura 10 se puede ver una escena de este simulador de conducción (Virtual Driver Interactive, 2019).

2.3 Simuladores de conducción de uso doméstico para ordenador

Los simuladores de conducción de uso doméstico para ordenador son aquellos que no cuentan con soporte hardware para hacer la simulación más realista. El fabricante de estos simuladores solo distribuye el software que se instala en el ordenador del cliente.

Por otro lado, el cliente puede añadir periféricos al ordenador que tiene el software del simulador instalado y así hacer la experiencia más realista. Estos simuladores son más asequibles, por eso están situados en la clasificación de uso doméstico.

A continuación, se habla de dos simuladores de uso doméstico para ordenador de los muchos que se pueden encontrar en el mercado.

2.3.1 City Car Driving

Este simulador está diseñado para que los usuarios superen sus miedos a conducir en la vía pública, sin salir de la comodidad de su casa sin correr ningún tipo de riesgo y por un precio muy bajo.

Dispone de físicas de movimiento y de comportamiento del automóvil realistas, que se intentan aproximar lo máximo a la realidad, como colisiones con obstáculos y otros vehículos.

Cuenta con unos gráficos modernos y de gran calidad, como se puede apreciar en la figura 11, que aportan mayor realismo a la conducción y ayudan al usuario a sumergirse en la simulación. En este simulador se han cuidado detalles como los modelos de los automóviles, los cuadros de mandos del automóvil dependiendo del modelo, recreación realista de condiciones climáticas, etc.



Figura 11: City Car Driving (City Car Driving, 2019).

Este simulador dispone de distintos modos de funcionamiento, en función de las necesidades de cada usuario como, por ejemplo:

- Conducción de automóviles en una gran ciudad: diseñado para que los usuarios aprendan a circular por una ciudad de manera eficiente y segura.
- Entrenamiento básico en habilidades de manejo: la mayoría de los estudiantes nunca antes se han sentado a los mandos de un vehículo y gracias a estas simulaciones comprenden los principios básicos de la conducción de un automóvil.

- Ejercicios estándar y avanzados en el autódromo especial: que permiten a los usuarios enfrentarse a situaciones básicas a las que se tendrán que enfrentar en el examen de conducir.

Este simulador dispone además de un sistema de control de reglas de tráfico que avisa al conductor en caso de cometer infracciones y también le aconseja sobre cómo debe actuar en cada situación que se le presenta (City Car Driving, 2019).

2.3.2 Driver Test

La empresa Arisoft ha creado este simulador de conducción de bajo coste que permite aprender sobre seguridad vial y mejorar la conducción. Una de las características reseñables es que su origen es español.

Cuenta con veintiún ejercicios que permiten al usuario practicar ciertas situaciones ante las que se suele tener dificultad. Además, permite practicar la conducción nocturna y bajo determinadas condiciones climáticas adversas, en la figura 12 se muestra lo que vería un usuario realizando una simulación.



Figura 12: Driver Test (Blog de la Auto escuela, 2019).

Driver Test permite a los usuarios adquirir los conceptos básicos necesarios para conducir desde su casa y por un precio muy asequible (Blog de la Auto escuela, 2019).

2.4 Simuladores de conducción para Android

Este tipo de simuladores es el que menos realismo aporta a la simulación, ya que no se pueden añadir periféricos al dispositivo para que se mejore el realismo, a diferencia de lo que ocurre con los simuladores para ordenador.

Sin embargo, pueden ser útiles para los usuarios para reforzar ciertos conceptos sobre seguridad vial y la conducción de vehículos y con la ventaja de poder practicar con ellos en cualquier lugar, ya que es software diseñado para un *smartphone*. Son simuladores dedicados al aprendizaje y al entretenimiento. Esto

puede ser una ventaja, ya que puede motivar al usuario a usar estas aplicaciones y afianzar conceptos sobre seguridad vial y conducción. A continuación, se habla de dos aplicaciones para Android de las muchas que se pueden encontrar en el mercado.

2.4.1 Car Driving School Simulator

Esta aplicación para dispositivos Android es un simulador de conducción muy realista. El objetivo de este juego es demostrar tus habilidades para conducir correctamente y cumplir con todas las normas de tráfico.



Figura 13: Car Driving School Simulator (Google Play - a, 2019).

Una de las características de este simulador es que dispone de un modo multijugador, en el que gana el jugador que más correctamente conduzca. Al empezar en la aplicación, se enseñan al usuario todas las normas básicas de conducción en un circuito cerrado.

Es un simulador básico al estar desarrollado para móviles, pero aporta muchos conocimientos a los usuarios y puede ser una toma de contacto interesante con la conducción de vehículos. En la figura 13 se muestra el contenido que vería el usuario en su móvil (Google Play - a, 2019).

2.4.2 Dr. Driving 2

Este simulador de conducción ha sido autoproclamado por los creadores “simulador de conducción móvil más grande de todos los tiempos”. En este simulador conduces un vehículo en primera persona y debes respetar todas las normas de tráfico para ganar puntos.



Figura 14: Dr. Driving 2(Google Play - b, 2019).

A nivel visual es bastante atractivo, como se puede observar en la figura 14, y permite jugar en línea con otros usuarios para competir por ver quién es el mejor conductor a la hora de respetar las normas de tráfico (Google Play - b, 2019).

2.5 Comparativa de los simuladores de conducción

En los apartados anteriores se ha hablado de algunos de los simuladores más relevantes que existen actualmente en el mercado. Todos estos simuladores se han desarrollado con la intención de formar a usuarios en la conducción de vehículos de distintos tipos.

Todos estos simuladores son muy distintos teniendo en cuenta sus características y funcionalidades de cara a la simulación de situaciones de conducción, por este motivo se separaron en cuatro grupos en función de las mismas. Debido a esto, realizar una comparación eficaz de todos ellos resulta muy complejo. Por ejemplo, si deseamos compararlos por su precio en el mercado, en los apartados anteriores existen simuladores con un precio muy elevado, como es el caso de los simuladores de conducción de uso profesional con experiencia inmersiva, y simuladores que son gratuitos, como los simuladores de conducción para Android. Esta gran diferencia entre los precios se debe a las características que aportan cada uno de ellos para hacer las simulaciones más realistas, cuanto mayor realismo aporte el simulador a la simulación, mayor es su precio en el mercado.

Otra característica que puede resultar interesante es la capacidad de la que disponen estos simuladores para ser transportados o ser utilizados en cualquier lugar. Esta característica también tiene relación con el precio porque cuantas más características tenga el simulador para hacer las simulaciones más realistas, mayor es el precio y más difícil es su transporte.

Como este proyecto se centra en modificar una plataforma web que se utiliza para almacenar los datos de un simulador de conducción, ha resultado interesante comentar y comparar cómo se almacenan los datos de las simulaciones en estos simuladores.

Todos estos simuladores de conducción de uso profesional con experiencia inmersiva disponen de un puesto para el instructor. Este puesto da la posibilidad de analizar los datos de la simulación en tiempo real. Por ello, todos estos

simuladores disponen de una base de datos para almacenar la información y una plataforma creada por el fabricante del simulador donde el instructor puede ver los datos de las simulaciones de los usuarios. Este tipo de simuladores son los únicos que disponen de algo similar a lo que se ha desarrollado en este proyecto. Los otros tipos, como los simuladores de conducción de uso profesional para ordenador, los simuladores de conducción de uso doméstico para ordenador y los simuladores para Android almacenan y muestran los datos al usuario dentro de la propia aplicación que contiene el simulador.

3. Plataforma web: posibilidades de desarrollo

Existe una infinidad de posibilidades para desarrollar una plataforma web con una API REST, que es lo que se está buscando en este caso. A continuación, se va hablar de las posibilidades que existen y el porqué de la opción seleccionada para realizar esta plataforma.

Para comenzar entre las distintas posibilidades, existe una primera diferenciación clara, ya que para desarrollar una plataforma web se pueden utilizar dos tipos de aplicaciones, un *framework* o un gestor de contenidos (*Content Management System* o CMS). Para entender esta diferenciación, primero hay que saber qué con y cómo funcionan estos tipos de aplicaciones. Por ello, se va a hacer un estudio de estos dos tipos analizando algunos de los más interesantes en cada caso y su utilidad (R.J.Corchero, 2016).

3.1 Frameworks para el desarrollo de aplicaciones web, ¿Qué son?

Un *framework* es un conjunto de herramientas y bibliotecas diseñadas para ayudar y facilitar las operaciones de desarrollo web a los usuarios. La traducción literal del término *framework* es marco de referencia. Esto puede resultar útil como punto de partida para entender qué es y para qué se utiliza este tipo de aplicaciones.

Un *framework* se puede entender como un esquema o patrón que se utiliza para estructurar de mejor manera el código, permitiendo programar escribiendo menos líneas código. Un *framework* también puede entenderse como, una aplicación que está incompleta y que se puede configurar para añadir líneas de código en función del tipo de aplicación que se desea desarrollar.

Este marco de referencia o marco de trabajo ofrece código que puede ser reutilizable en cualquier sistema. Este código permite resolver tareas comunes y simplificar el desarrollo de una aplicación.

A continuación, se hablará de tres *frameworks* conocidos, con características que han resultado interesantes para el desarrollo de una plataforma web (MDM web docs, 2019).

3.1.1 Zend Framework

Zend Framework es un *framework* creado por la empresa Zend que forma parte de Rogue Wave. Aunque esta empresa ha tenido el papel principal en la creación de este *framework*, muchas otras como Google, Microsoft y Stikelron han contribuido con componentes importantes para este *framework*. Este *framework* de código abierto se usa para el desarrollo de aplicaciones y servicios web utilizando PHP 5.6+. Sus componentes utilizan el paradigma de programación orientado a objetos y todas las características del lenguaje PHP.

Entre las características interesantes de este *framework* se encuentra que permite implementar el Modelo Vista Controlador (MVC), MVC es una arquitectura de software que separa los datos de la aplicación, la lógica de la misma y la interfaz de usuario en tres capas (Universidad de Alicante, 2020). Dispone de una capa de abstracción de la base de datos, aporta la posibilidad de implementar formularios con el formato HTML, etc.

Este *framework* permite utilizar sus componentes por separado. El conjunto de todos sus componentes forma este *framework* de aplicaciones web, con la posibilidad de facilitar el desarrollo del código de cualquier aplicación (R.J.Corchero, 2016) (ZendFramework, 2019).

3.1.2 Fuel PHP

Este *framework* es relativamente nuevo, comenzó su desarrollo a finales de 2010 por Dan Horrigan y, a su equipo, se unieron Phil Sturgeon, Jelmer Schreuder, Harro Verton y Frank de Jonge. Al equipo se unió también Steve West en 2013. Este equipo tiene una gran experiencia en PHP. La primera versión de este *framework* apareció el 31 de julio de 2011, listo para desarrollar nuevos proyectos.

Una de las características que más llaman la atención de este *framework* es que busca mejorar apoyándose en la comunidad de usuarios, posibilitando que cualquier usuario del *framework* ayude a detectar y corregir errores en él. FuelPHP está basado en el MVC. Por otro lado, es un *framework* modular y extensible, permitiendo al usuario crear una aplicación modular.

También cuenta con un gran número de características que mejoran la seguridad de los desarrollos, como, prevención de inyección SQL, protección de token CSRF, filtrado XSS, etc.

Estas son algunas de las muchas características interesantes de las que dispone este *framework* para PHP 5.4+, que le hacen rápido, simple y flexible. (Fuel PHP, 2019)

3.1.3 Slim

Es un *micro-framework*. Se denomina así porque, a diferencia de otros, cuenta con menos componentes pre-instalados, pero es igual de eficiente que los de mayor complejidad. Cuenta con las características necesarias para escribir aplicaciones web y API simples pero muy eficientes.

Algunas de las características que resultan interesantes en este *framework* tan ligero son:

- Proporciona un enrutador rápido que resulta muy potente, proporcionando devoluciones de llamadas a solicitudes de HTTP y URI específicas.

- Permite crear aplicaciones utilizando *middleware* concéntrico, que permiten ajustar solicitudes y respuestas HTTP en torno a la aplicación desarrollada en el *framework*.
- Es un *framework* de código abierto para PHP que se ha desarrollado desde cero.

Todas estas características y muchas más de las que dispone este *framework* hacen que, a pesar de su sencillez, resulte tan interesante (Slim, 2019).

3.2 Gestores de contenido (CMS)

CMS hace referencia a *Content Management System*, cuya traducción es Sistema de Gestión de Contenido. Estos sistemas de gestión de contenido son aplicaciones normalmente web, que permiten diferenciar a sus usuarios en función de sus permisos de acceso.

En función de los permisos de los que disponga cada usuario, tendrá mayor o menor capacidad para editar o gestionar el contenido de la aplicación.

Esto es una característica muy interesante de cara a la plataforma web, ya que dispondrá de varios usuarios y, en función de los permisos, tendrán capacidad para administrar la plataforma o no.

Otra característica interesante de los CMS es que permiten crear y administrar sitios web por medio de una aplicación. Para poder realizar esto no se necesitan unos conocimientos técnicos elevados. Esta característica en realidad no resulta muy útil en este caso, porque la plataforma web necesita más características que las que proporciona un CMS en su origen.

A continuación, se hablará sobre las características de tres sistemas de gestión de contenidos muy conocidos por los usuarios que se dedican al desarrollo de sitios web en la actualidad (R.J.Corchero, 2016) (Joomla!, 2019) (C.Aubry, 2012).

3.2.1 CMS WordPress

El comienzo de WordPress empieza en 2003 cuando Mike Little y Matt Mullenweg crearon una bifurcación de b2 / cafelog. WordPress es un CMS que está construido en PHP y MySQL con licencia bajo la GPLv2, lo que sea la opción más famosa entre los sitios web.

WordPress es un proyecto de código abierto que ha evolucionado mucho a lo largo del tiempo gracias a la ayuda de científicos, desarrolladores, bloggers y más expertos.

Este CMS da soporte a más del 34% de los sitios webs que existen actualmente, desde sitios web normales, blogs, portales y sitios web empresariales.

WordPress está diseñado para todo tipo de usuarios, aportando simplicidad para los usuarios finales y complejidad para los desarrolladores. Esto lo hace muy flexible y a la vez fácil de usar.

Estas son algunas de las características más interesantes de este CMS tan famoso y tan usado:

- Simplicidad: ofrece la posibilidad de una conexión rápida y poder editar el contenido en el momento.
- Flexibilidad: utilizando este CMS puedes crear y diseñar cualquier tipo de sitio web, modificando los temas y ampliando el propio CMS con complementos.
- Publica con facilidad: WordPress permite crear publicaciones y darlas formato, con la capacidad de publicarlas con un solo clic.
- Gestión de usuarios: con este CMS tienes la capacidad de restringir el permiso a ciertos grupos de usuarios y dárselo a otros. Esto puede ser muy útil para diseñar un perfil de administrador que gestione la web y el resto de perfiles de usuarios normales.
- Cumplimiento completo de los estándares: WordPress cumple con los estándares establecidos por W3C, lo que hace a los sitios webs diseñados con este CMS compatibles con los navegadores en un futuro.
- Instalación y actualizaciones fáciles: WordPress resulta muy fácil de actualizar e instalar para los usuarios.

Todas estas características, y muchas más de las que dispone este CMS, hacen que WordPress que sea uno de los más famosos y más usados de la actualidad (WordPress.org, 2019).

3.2.2 Joomla!

Joomla! es un gestor de contenidos (CMS) que es gratuito y con código abierto que se utiliza para publicar contenidos web. Este sistema de gestor de contenido ha ganado varios premios por ser el mejor CMS.

Está basado en el MVC con el que se puede operar independientemente del CMS. Todo esto es útil para crear aplicaciones web de gran calidad.

Dispone de una comunidad web global de desarrolladores y voluntarios, que ayudan que la plataforma sea sencilla de usar, con la capacidad de ser ampliada, multilingüe, etc.

Este CMS se usa en todo el mundo para distintos tipos de sitios web, como sitios web corporativos, sitios web de pequeñas empresas, sitios web gubernamentales, etc.

Joomla! está desarrollado en PHP y MySQL que permite crear aplicaciones muy potentes. Esto resulta muy útil para que los desarrolladores creen aplicaciones de manera rápida y fácil, como sistemas de control de inventario, catálogos de productos, sistemas de reserva, etc. (Joomla!, 2019).

3.2.3 Drupal

Drupal dio sus primeros pasos en 1999, cuando el estudiante belga Dreis Buytaert desarrolló un sitio web en su ciudad universitaria de Amberes. Esto hizo que muchos desarrolladores se unieran al proyecto para construir un CMS.

Esa unión de todos los desarrolladores llevo a la aparición en 2004 de la versión 4.5 de Drupal que supuso los cimientos de lo que conocemos hoy en día.

Desde su aparición, Drupal fue diseñado como un sistema de gestión de contenidos de carácter modular y con la capacidad de ser configurable, lo que le hace muy interesante de cara a los desarrolladores de software.

Este sistema de gestión de contenidos es de código abierto, desarrollado en el lenguaje PHP con una licencia GNU/GPL y dispone de una activa comunidad de usuarios que aportan soporte a los desarrolladores.

Estas son algunas de las características que hacen a Drupal un CMS muy recomendable para crear y gestionar comunidades de usuarios en Internet:

- Flexibilidad: esta flexibilidad permite diseñar y personalizar campos, que pueden ser utilizados en distintos tipos de contenido. Estos campos aportan la posibilidad de almacenar los datos en bases de datos SQL, NoSQL o almacenamientos remotos.

- Diseño de platillas: Drupal aporta la posibilidad de gestionar el contenido que aparece en la pantalla apoyándose en la Render API de la que dispone y que facilita el proceso de mostrar el contenido al usuario de manera ordenada y en los *hooks* de Drupal, que son funciones que permiten al desarrollador modificar el comportamiento de algunas secciones de código sin tener que modificar esa sección en concreto.

- Accesibilidad: se refiere a la capacidad que aporta Drupal para acceder a las pantallas de administración de los sitios web.

- Ficheros e imágenes: el núcleo de Drupal permite generar versiones diferentes para las miniaturas de las imágenes o *thumbnails*, que resultan útiles como vista previa, con la posibilidad de utilizar la gestión privada y la gestión pública de ficheros.

- Testing: Drupal aporta un entorno de testeo, que da la posibilidad de realizar pruebas de funcionamiento de nuevos componentes que se desean añadir al núcleo.
- Soporte para base de datos: este CMS cuenta con soporte para bases de datos SQLite, MySQL/MariaDB y PostgreSQL *out-of-the-box*.
- Soporte para distribución: Drupal permite utilizar perfiles de instalación para sitios webs personalizados que se basan en Drupal.
- Extensible: Drupal cuenta con gran número de módulos que permiten ampliar sus características de inicio. Todo esto se debe a la gran comunidad de usuarios que existe por detrás de Drupal dando soporte y mejorándolo.

Todas estas características, y más de las que dispone Drupal, hacen a este CMS de código abierto una buena opción para crear un espacio único y funcional para un gran número de usuarios (R.J.Corchero, 2016) (C.Aubry, 2012) (Drupal – a, 2019) (Drupal – b, 2019).

3.3 Comparativa de tecnologías y opción elegida

En este apartado se hablará del porqué de la opción elegida a la hora de realizar el desarrollo de la plataforma, teniendo en cuenta las posibilidades de las que se han estado hablando en el capítulo y sus características.

Esta decisión resulta determinante ya que, la decisión que se tome, condicionará el desarrollo de la plataforma web y las posibilidades que se disponen para una futura actualización de la misma. Aunque en este caso esta decisión se tomó con anterioridad, se comentará el porqué de ella y el motivo por el que se ha optado por mantenerla en el tiempo, para esta actualización.

En el momento de tomar la decisión de qué tipo de tecnología utilizar para realizar un desarrollo de una plataforma web, la primera diferenciación que se ha tomado es dividir las posibilidades entre *frameworks* y sistemas de gestor de contenidos (CMS).

Se comenzará hablando de los *frameworks* y de la comparativa entre los tres de los que hemos hablado en los puntos anteriores. Zend Framework es un *framework* desarrollado en el lenguaje de programación PHP, al igual que los otros *framework* de los que se ha hablado (Fuel PHP y Slim), pero con la diferencia de que el *framework* Zend sigue el paradigma de programación orientado a objetos. Por otro lado, Zend sigue el modelo de estructuración de la codificación por componentes. Esta manera de programar se centra en dividir el funcionamiento de las aplicaciones en componentes funcionales ensamblados entre sí, que facilita la actualización de ciertas partes de la aplicación sin la necesidad de modificar otros componentes (ZendFramework, 2019).

Por otro lado, una característica muy interesante que dispone Fuel PHP, y que no disponen los otros dos *frameworks*, es que sigue el paradigma de programación

Modelo-Vista-Controlador, que resulta muy interesante de cara al desarrollo de un sitio web o una plataforma web. Esto se debe a que este paradigma permite dividir las funcionalidades de un sitio web en estas tres partes, permitiendo separar ligeramente el desarrollo de cada una de ellas (Fuel PHP, 2019).

Lo que resulta muy interesante del *framework* Slim es su simplicidad a diferencia de los otros *frameworks*. Slim es mucho más ligero, pero aportando muchas de las funcionalidades de un *framework* más pesado, como pueden ser los otros dos de los que se ha hablado en apartados anteriores (Slim, 2019).

Hubo dos aspectos que hacen que se descarten estos *framework* de cara al desarrollo de la plataforma web. Por un lado, estos *frameworks* no disponen de tecnología que facilite el diseño de una API REST, que es necesaria en esta plataforma web para la comunicación con el servidor. Por otro lado, cualquier *framework* implica un mayor desarrollo porque no aporta muchas funcionalidades que para este caso son necesarias. Por eso, se descartó la posibilidad de utilizar un *framework* y se optó por un sistema de gestión de contenidos (CMS), porque disponen de forma original de muchas funcionalidades útiles en nuestro caso, que facilitarían el desarrollo.

Una vez que se tiene clara la tecnología que se va a utilizar, un sistema de gestión de contenido (CMS), se hablara del porqué de la elección original entre las tres posibilidades que se barajaban y el porqué de seguir manteniéndola en el tiempo en este nuevo desarrollo.

El desarrollador de esta plataforma web en sus orígenes opto por Drupal, debido a que él se dedicaba profesionalmente al desarrollo de sitios webs con este CMS, lo que le facilito el desarrollo de la misma en gran medida. Por otro lado, él también prefirió desarrollar esta plataforma sobre un CMS porque aportan muchas funcionalidades útiles y necesarias para el desarrollo y, si se hubiera optado por un *framework*, tenía que crearlas desde cero.

Se optó por Drupal, aunque cualquiera de las otras dos opciones expuestas podría haber sido viables para este caso. En el caso de WordPress, al ser un sistema de gestión de contenidos muy famoso, se dispone de mucha información de soporte para el desarrollo, aunque ha evolucionado poco su núcleo en los últimos años, por lo que se descartó (R.J.Corchero, 2016).

Probablemente, desde mi punto de vista, la mejor opción podría haber sido WordPress porque es el CMS más conocido y tiene una gran comunidad, aunque Joomla! también me llama bastante la atención, porque resulta menos conocido, dispone de una gran comunidad de usuarios para dar soporte y ha sido premiado varias veces por su desarrollo (WordPress.org, 2019) (Joomla!, 2019).

El por qué de mantener el criterio de utilizar Drupal, a la hora de desarrollar la actualización de la plataforma web, fue primordialmente porque se trataba de una actualización modificando lo que ya había y manteniendo muchas cosas que resultaban interesantes, como el formato de las plantillas que se ha seguido en la plataforma y que con otro CMS podría ser más complicado, la API Rest, etc.

Por eso se optó por mantener Drupal, porque comenzar un desarrollo de la plataforma con otro CMS probablemente haría el desarrollo muy extenso. Por otro lado, los usuarios actuales de la plataforma ya están acostumbrados a trabajar con Drupal y están contentos con su funcionamiento. Si algo funciona bien, no hay

por qué cambiarlo a riesgo de que no cumpla las expectativas (R.J.Corchero, 2016) (C.Aubry, 2012) (Drupal – a, 2019) (Drupal – b, 2019).

3.4 Instalación y requisitos

Teniendo en cuenta que Drupal está desarrollado en PHP y necesita de una base de datos, la mejor opción de cara a una instalación en local de Drupal sería utilizar un stack LAMP (Linux-Apache_MySQL-PHP) como, por ejemplo, que facilite trabajar en local.

En este caso, se ha utilizado XAMPP porque se trabajó con él con anterioridad, en desarrollos de sitios web para asignaturas del grado. Ha sido un recurso muy útil para el desarrollo de la plataforma ya que ha permitido comprobar el funcionamiento de la plataforma en local, corregir errores y utilizar algunas de las características de este programa, como la posibilidad de implementar una base de datos e instalar un servidor PHP en local, que es lo que se buscaba para este desarrollo. En suma, ha sido una gran ayuda porque ha permitido trabajar con los datos reales de los que se disponen actualmente en la plataforma y, por lo tanto, las pruebas de las modificaciones de la plataforma han sido más próximas a la realidad (XAMPP, 2019).

Retomando el proceso que hay que seguir para instalar Drupal y poder trabajar con él en local, una vez que se tiene XAMPP instalado, ya podremos desplegar el sitio web en local. Solo se necesitará copiar los archivos que forman el sitio web en la carpeta C:\xampp\htdocs\“directorio con el nombre de la aplicación” (suponiendo que el programa se haya sido instalado en la unidad C directamente) y, una vez copiado, abrir XAMPP y arrancar el servidor Apache y la base de datos MySQL de este programa pulsando el botón *Start* que se observa en la figura 15 bajo el apartado *actions* en ambos. Ahora ya se dispone de la aplicación desplegada en local y se podrá acceder a ella abriendo un buscador y escribiendo la URL [http://localhost/ “directorio con el nombre de la aplicación”](http://localhost/“directorio con el nombre de la aplicación”).

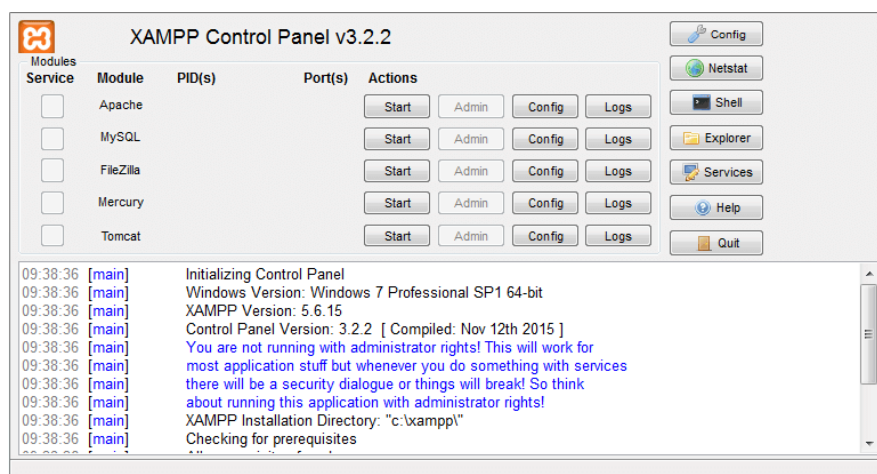


Figura 15: Interfaz de usuario de XAMPP.

Como se va a trabajar con Drupal 7, lo más recomendable es descargar la última versión en el sitio oficial, copiarlo en la carpeta htdocs, desplegarlo en local y

comenzar la instalación accediendo a la URL antes mencionada y seguir los pasos que indica el instalador.

Hay que tener en cuenta que Drupal necesita una base de datos para poder instalarse, pero al instalar XAMPP se ha instalado MySQL que es uno de los tipos que soporta Drupal.

Crear la base de datos que necesita Drupal para la instalación es muy sencillo utilizando XAMP, simplemente hay que abrir XAMPP y pulsar el botón *Admin* de la MySQL que se observa en la figura 15 y se abrirá la interfaz de phpMyAdmin donde el usuario verá lo que se muestra en la figura 16. En esta interfaz resulta muy sencillo crear una nueva base de datos para la aplicación.

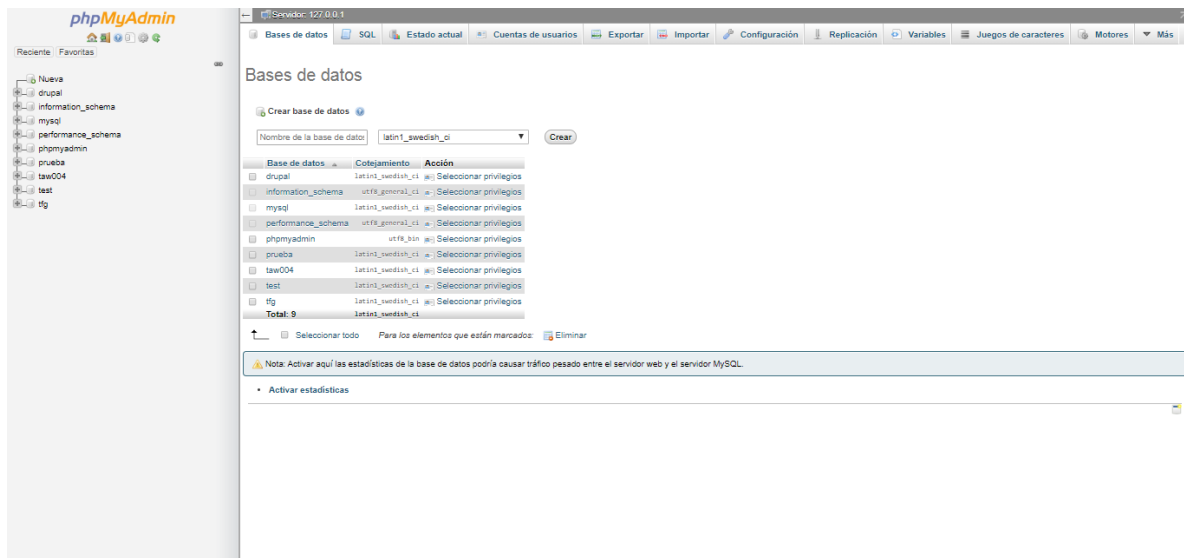


Figura 16: Interfaz de usuario de phpMyAdmin.

Con la base de datos creada y Drupal instalado, solo quedaría instalar los módulos necesarios para el correcto funcionamiento de la plataforma web y el módulo creado para analizar los datos del simulador de conducción. Estos son los módulos que habría que instalar principalmente (probablemente muchos de ellos tengan dependencias con otros y sea necesario instalarlos también):

- Libraries API.
- Services.
- Charts.
- XAutoload.

Una vez instalados estos módulos y los módulos con los que tienen dependencias, solo quedaría entrar en la sección Modules de Drupal como administrador y activar los módulos que se acaban de instalar.

Todo este proceso que se ha explicado sería con el fin de desarrollar un módulo de Drupal desde cero, pero como en este caso se trata de una actualización de un módulo que ya está funcionando, el proceso tendría algunas variaciones. En este caso, solo habría que copiar el código de la plataforma actual en la carpeta htdocs de XAMPP, como se ha explicado antes, e importar en phpMyAdmin la base de datos que dispone la plataforma actual, para poder trabajar en local con los datos

de usuarios reales. Siguiendo estos pasos, ya dispondríamos de la plataforma original montada en el servidor local y ahora todas las modificaciones que hagamos del código en el IDE, en este caso phpStorm, se verán reflejadas en la plataforma al refrescar la URL de la plataforma montada en local (R.J.Corchero, 2016) (C.Aubry, 2012).

4. Desarrollo de la plataforma web

Para entender qué es lo que se ha pretendido hacer en esta actualización, primero hay que entender qué es la plataforma y cómo se desarrolló en sus orígenes.

La plataforma web para almacenar los datos de un simulador de conducción tomó como eje principal Drupal 7, se desarrolló como un módulo nuevo que se instaló como uno más de los que dispone Drupal. Por lo tanto, la actualización de la plataforma se ha basado en modificar el módulo que se desarrolló en sus orígenes para crearla, buscando mejorar el análisis de los datos y adaptarla a las necesidades que han surgido a los usuarios con el paso del tiempo.

Además de estas modificaciones que resultaban necesarias para los usuarios, se ha aprovechado para mejorar y corregir algunos aspectos de la plataforma que podían llevar a confusión a los usuarios. Una de las ideas principales era permitir a los usuarios poder separar las simulaciones por el tipo de vehículo con el que se realizaron en el simulador de conducción. Se tomó como punto de partida que el tipo de vehículo sería un dato que enviaría el simulador de conducción a la plataforma junto con los datos de la simulación realizada por el usuario en cada momento.

A continuación, se va a proceder a explicar cómo se ha llevado a cabo la actualización de la plataforma comentando solo algunos trozos del código que resultan interesantes, ya que explicar todo el código haría este apartado demasiado extenso. Para poder entender el desarrollo del código de la actualización, las explicaciones se van a apoyar en una comparativa de las distintas pantallas, teniendo en cuenta el antes y después de la actualización de la plataforma y así poder apreciar mejor cuáles han sido las modificaciones sobre lo que se disponía antes en la plataforma.

4.1 Modificaciones relacionadas con la BBDD

Para poder analizar las simulaciones en función del tipo de vehículo empleado para realizarlas, se ha tenido que modificar la base de datos para disponer de una entrada para almacenar el tipo de vehículo. Teniendo en cuenta la estructura de la base de datos antes de la actualización, en ese momento la base de datos contaba con cinco tablas, siendo la principal la tabla `rjsim_partida`, de la que dependían todas las demás tablas. Por eso se pensó en añadir una columna en esta tabla para almacenar el tipo de vehículo.

Entonces se decidió que el simulador mandaría el tipo de vehículo con el que el usuario realiza la simulación como un entero. Por ese motivo, se incluyó en la tabla `rjsim_partida` de la base de datos una nueva columna para incluir este entero. Esta modificación y el estado actual de la base de datos se puede ver en la figura 17. Los tipos de vehículos disponibles actualmente en el simulador son:

(Identificador. Nombre en español: “Nombre en inglés”)

1. Turismo con motor de combustión: “*Internal combustion engine passenger car*”

2. Turismo con motor eléctrico: “*Electric passenger car*”
3. Autobús con motor de combustión: “*Internal combustion engine city bus*”
4. Autobús con motor eléctrico: “*Electric city bus*”
5. Camión con motor de combustión: “*Internal combustion engine truck*”
6. Camión con motor eléctrico: “*Electric truck*”

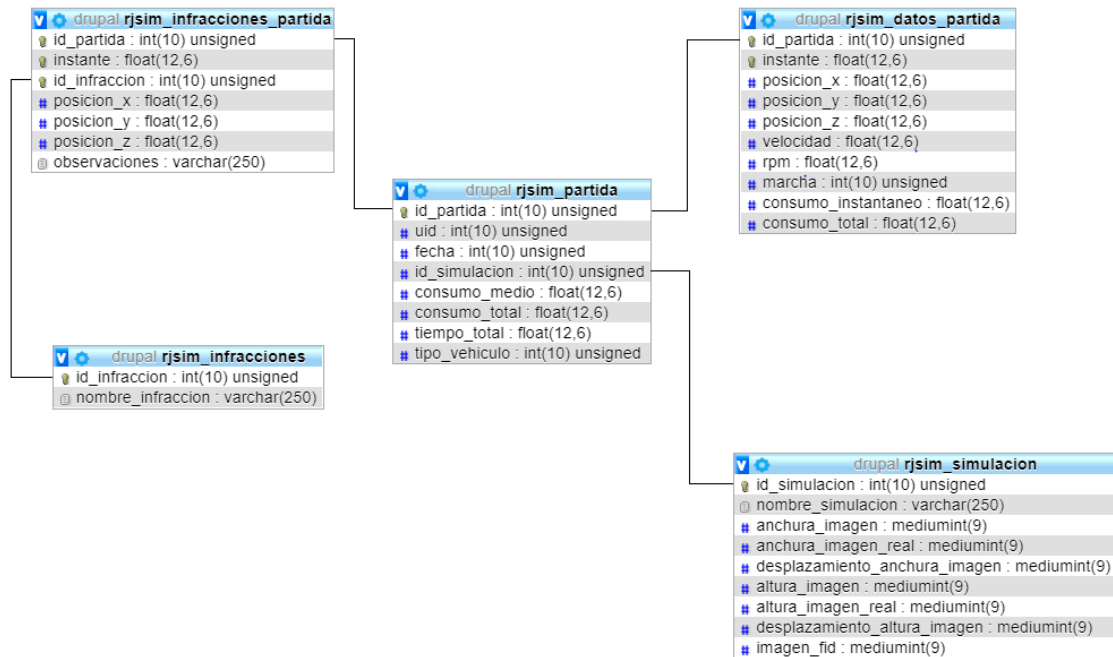


Figura 17: Estructura de la Base de datos después de la actualización.

Para que esta modificación sea permanente y se mantenga cuando se instale el módulo en un entorno nuevo y se cree una nueva base de datos, se tuvo que modificar el archivo `rjsimulador.install` y el `hook` `rjsimulador_schema()` que contiene. Este `hook` se encarga de generar la base de datos cuando se instala el módulo por primera vez. Las modificaciones realizadas en el archivo `rjsimulador.install` y en el `hook` `rjsimulador_schema()` se muestran en la figura 18. Dentro de este `hook` fue necesario incluir la nueva columna de la tabla `rjsim_partida` definiendo sus parámetros y su comportamiento dentro de la base de datos, como se muestra en la figura 18.

```

function rjsimulador_schema() {
    $schema['rjsim_partida'] = array(
        'description' => t( string: 'Listado de partidas jugadas'),
        'fields' => array(
            'id_partida' => array(
                'description' => 'ID de la partida',
                'type' => 'serial',
                'unsigned' => TRUE,
                'not null' => TRUE
            ),
            'uid' => array(
                'description' => 'UID del usuario que hizo al partida',
                'type' => 'int',
                'unsigned' => TRUE,
                'not null' => TRUE,
                'default' => 0
            ),
            'tipo_vehiculo' => array(
                'description' => 'Vehiculo con el que se realiza la simulacion',
                'type' => 'int',
                'unsigned' => TRUE,
                'not null' => TRUE,
                'default' => 1
            ),
            'fecha' => array(
                'description' => 'Timestamp de la fecha en la que se hizo la partida',
                'type' => 'int',
                'unsigned' => TRUE,
                'not null' => TRUE,
                'default' => 0
            ),
            'id_simulacion' => array(
                'description' => 'ID de la simulacion que se hizo en la partida',
                'type' => 'int',
                'unsigned' => TRUE,
                'not null' => TRUE,
                'default' => 1
            ),
            'consumo_medio' => array(
                'description' => 'Consumo medio del vehiculo durante la partida',
                'type' => 'float',
                'precision' => 12,
                'scale' => 6,
                'not null' => TRUE,
                'default' => 0
            ),
            'consumo_total' => array(
                'description' => 'Consumo total del vehiculo al finalizar la partida',
                'type' => 'float',
                'precision' => 12,
                'scale' => 6,
            )
        )
    );
}

```

Figura 18: Modificaciones realizadas en el archivo rjsimulador.install y en el hook rjsimulador_schema().

Una vez que se ha hablado de los cambios realizados en la base de datos para poder almacenar el tipo de vehículo y que se tomó como punto de partida en el desarrollo de la actualización, se analizará cómo se han tenido que modificar los accesos a la base de datos con la aparición del nuevo dato que enviará el simulador.

Al incluir el nuevo campo en la base de datos, ha sido necesario modificar el método que se encarga de hacer consultas a la tabla de la base de datos rjsim_partida. Este método, que se incluye en el archivo DBDataProvider.php, se encarga de recuperar una partida en función de un id de partida y se ha modificado para incluir el nuevo campo. Dentro del método loadPartidaById(\$id_partida), en el momento que se realiza la consulta a la base de datos apuntando a la tabla rjsim_partida, ha sido necesario incluir el campo tipo de vehículo para un funcionamiento óptimo, como se puede apreciar en la figura 19.

```

/**
 * @inheritdoc
 */
public function loadPartidaById($id_partida) {
    $partida = NULL;

    $query = db_select( table: 'rjsim_partida', alias: 'p');
    $query->fields( table_alias: 'p', array(
        'uid',
        'fecha',
        'id_simulacion',
        'tipo_vehiculo',
        'consumo_medio',
        'consumo_total',
        'tiempo_total'
    ))
    ->condition('id_partida', $id_partida, '=');
    $resultado = $query->execute();

    if ($resultado->rowCount() == 0) {
        throw new Exception( message: "No existe una partida con ese ID.");
    }

    while ($record = $resultado->fetchAssoc()) {
        $partida = new Partida($record['uid'], $record['fecha'], $record['id_simulacion'],$record['tipo_vehiculo']);
        $partida->setIdPartida($id_partida);
        $partida->setConsumoMedio($record['consumo_medio']);
        $partida->setConsumoTotal($record['consumo_total']);
        $partida->setTiempoTotal($record['tiempo_total']);
    }

    return $partida;
}

```

Figura 19: Modificaciones realizadas en el método loadPartidaById(\$id_partida) en el archivo DBDataProvider.php.

Por otro lado, también se ha modificado el método que se encarga de almacenar partidas en la base de datos savePartida(Partida \$partida). Este método se encuentra en el archivo DBDataSaver.php y en él se ha incluido el nuevo campo tipo de vehículo de la tabla rjsim_partida, como se muestra en la figura 20.

```

/**
 * @inheritdoc
 */
public function savePartida(Partida $partida) {
    $idPartidaNuevo = db_insert( table: 'rjsim_partida')
    ->fields(array(
        'uid' => $partida->getUserUid(),
        'fecha' => $partida->getFecha(),
        'tipo_vehiculo' => $partida->getTipoVehiculo(),
        'id_simulacion' => $partida->getIdSimulacion(),
        'consumo_medio' => $partida->getConsumoMedio(),
        'consumo_total' => $partida->getConsumoTotal(),
        'tiempo_total' => $partida->getTiempoTotal()
    ))
    ->execute();

    $partida->setIdPartida($idPartidaNuevo);

    // Solo insertamos infracciones si existen
    if ($partida->getListaInfracciones() != NULL && $partida->getListaInfracciones()->count() > 0) {
        foreach ($partida->getListaInfracciones() as $infraccion) {
            $infraccion->setIdPartida($partida->getIdPartida());
            $infraccion->save();
        }
    }

    // Solo insertamos datos si existen
    if ($partida->getListaDatos() != NULL && $partida->getListaDatos()->count() > 0) {
        foreach ($partida->getListaDatos() as $dato) {
            $dato->setIdPartida($partida->getIdPartida());
            $dato->save();
        }
    }

    return array("message" => "Partida OK");
}

```

Figura 20: Modificaciones realizadas en el método savePartida(Partida \$partida) en el archivo DBDataSaver.php.

Como se puede apreciar en las modificaciones realizadas en los métodos que se utilizan para realizar consultar y almacenar simulaciones en la base de datos, también ha sido necesario modificar el objeto partida incluyendo el tipo de vehículo. Este objeto partida es muy importante dentro de la plataforma porque en

él se almacenan los datos de las simulaciones de los usuarios que se extraen de la base de datos. Aunque estas modificaciones no están relacionadas directamente con la base de datos, influyen en los métodos que se utilizan para trabajar con ella. En el archivo Partida.php se ha definido una variable \$tipo_vehiculo y se ha modificado el constructor del objeto partida, incluyendo la variable \$tipo_vehiculo, para que sea un valor necesario para poder generar un objeto partida y almacenar los datos. Esta modificación del fichero se puede ver en la figura 21.

```

/**
 * Class Partida Representa una partida del Simulador.
 */
class Partida implements ServicesAdapterInterface {
    /* ***** PROPERTIES ***** */
    /* ***** */
    /* @var int $id_partida */
    private $id_partida;
    /* @var int $userId */
    private $userId;
    /* @var int $fecha */
    private $fecha;
    /* @var int $tipo_vehiculo */
    private $tipo_vehiculo;
    /* @var int $id_simulacion */
    private $id_simulacion;
    /* @var string $nombre_simulacion */
    private $nombre_simulacion;
    /* @var float $consumo_medio */
    private $consumo_medio;
    /* @var float $consumo_total */
    private $consumo_total;
    /* @var float $tiempo_total */
    private $tiempo_total;
    /* @var ListaInfracciones $listaInfracciones */
    private $listaInfracciones;
    /* @var ListaDatosInstantaneos $datos */
    private $listaDatos;

    /* ***** CONSTRUCTOR ***** */
    /**
     * Partida constructor.
     * @param int $uid
     * @param int $fecha La fecha en formato UNIX.
     * @param int $id_simulacion
     * @param int $tipo_vehiculo
     */
    function __construct($uid, $fecha, $id_simulacion, $tipo_vehiculo) {
        $this->setUserId($uid);
        $this->setFecha($fecha);
        $this->setIdSimulacion($id_simulacion);
        $this->setTipoVehiculo($tipo_vehiculo);
    }
}

```

Figura 21: Modificaciones realizadas en el archivo Partida.php.

Al modificar el constructor de la clase partida, también ha sido necesario crear los métodos get() y set() para el tipo de vehículo. En la figura 22 se puede ver el nuevo método getTipoVehiculo() creado que permite recuperar el tipo de vehículo de una simulación.

```

/**
 * Método que devuelve el tipo de vehículo de una partida en formato numérico
 * @return int tipo_vehiculo.
 */
public function getTipoVehiculo() {
    return $this->tipo_vehiculo;
}

```

Figura 22: Nuevo método getTipoVehiculo() creado en el archivo Partida.php.

Como el tipo de vehículo utilizado al realizar la simulación se almacena en la base de datos como un entero, se creyó conveniente crear un método que a partir de ese entero devolviera el tipo de vehículo en formato texto, como se aprecia en la figura 23. Este método resultó útil para aplicaciones en la interfaz de usuario.

```
/**
 * Método que devuelve el tipo de vehículo de una partida en formato texto
 * @return String tipo_vehiculo.
 */
public function getTipoVehiculoNombre() {
    switch ($this->tipo_vehiculo) {
        case 1:
            $vehiculo = 'Internal combustion engine passenger car';
            break;
        case 2:
            $vehiculo = 'Electric passenger car';
            break;
        case 3:
            $vehiculo = 'Internal combustion engine city bus';
            break;
        case 4:
            $vehiculo = 'Electric city bus';
            break;
        case 5:
            $vehiculo = 'Internal combustion engine truck';
            break;
        case 6:
            $vehiculo = 'Electric truck';
            break;
        default:
            $vehiculo = 'Internal combustion engine passenger car';
            break;
    }
    return $vehiculo;
}
```

Figura 23: Nuevo método `getTipoVehiculoNombre()` creado en el archivo `Partida.php`.

A parte de los métodos `get()` creados en la clase `Partida`, se ha creado un método `setTipoVehiculo($tipo_vehiculo)`, que es el que se muestra en la figura 24. Este método permite definir un tipo de vehículo para un objeto `partida`.

```
/**
 * @param int $tipo_vehiculo
 * @throws InvalidArgumentException
 */
public function setTipoVehiculo($tipo_vehiculo) {
    if (is_numeric($tipo_vehiculo)) {
        $this->tipo_vehiculo = $tipo_vehiculo;
    }
    else {
        throw new InvalidArgumentException( message: "The type of vehicle is not a number.");
    }
}
```

Figura 24: Nuevo método `setTipoVehiculo($tipo_vehiculo)` creado en el archivo `Partida.php`.

4.2 Modificaciones de la página de inicio y el core de Drupal

Después de hablar de los cambios realizados en la base de datos para poder almacenar el tipo de vehículo con el que se realizará cada simulación, se van a proceder a analizar las modificaciones que se han realizado en la interfaz de usuario de la plataforma web. Se analizará la plataforma web accediendo a ella con un perfil de administrador, que permitirá ver la interfaz de usuario del

administrador y de un usuario normal al mismo tiempo, ya que dispone de permisos para ambos perfiles.

Lo primero que se realizó es actualizar ciertos módulos que se encontraban desactualizados y Drupal aconsejaba actualizar por medio de mensajes. Estos mensajes se veían en la plataforma en la parte superior cuando se navegaba por ella. Se optó por actualizarlos porque supondrían una mejora para el funcionamiento y, por otro lado, desaparecerían los mensajes que no resultaban atractivos a la hora de navegar por la plataforma web.

La primera visión de la plataforma web que disponen los usuarios al acceder a ella es la que se muestra en la figura 25. Es la página donde tienen que iniciar sesión para acceder a la plataforma. Esta página no se ha modificado y mantiene su formato original antes de realizar la actualización.

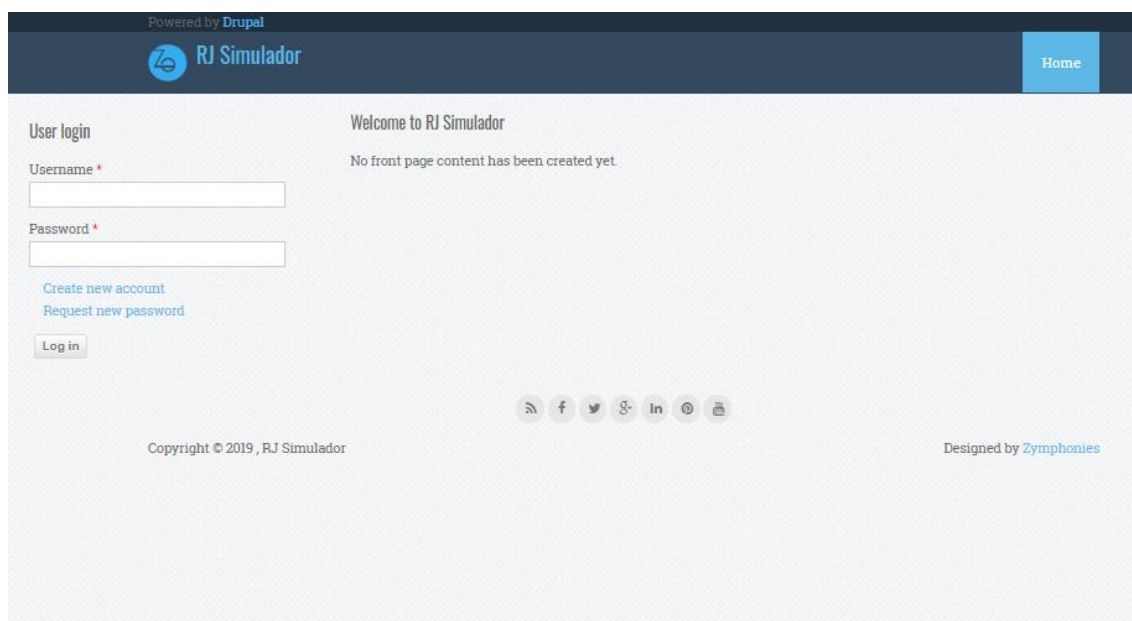


Figura 25: Página de inicio de sesión de la plataforma web.

Una vez que el usuario ha iniciado sesión en la plataforma web, lo primero que ve es lo que se muestra en la figura 26, donde accede a la página principal de la plataforma. Esta vista no ha sufrido modificaciones en la actualización para un perfil de usuario normal. En cambio, para el perfil de administrador, se han realizado unas pequeñas modificaciones.

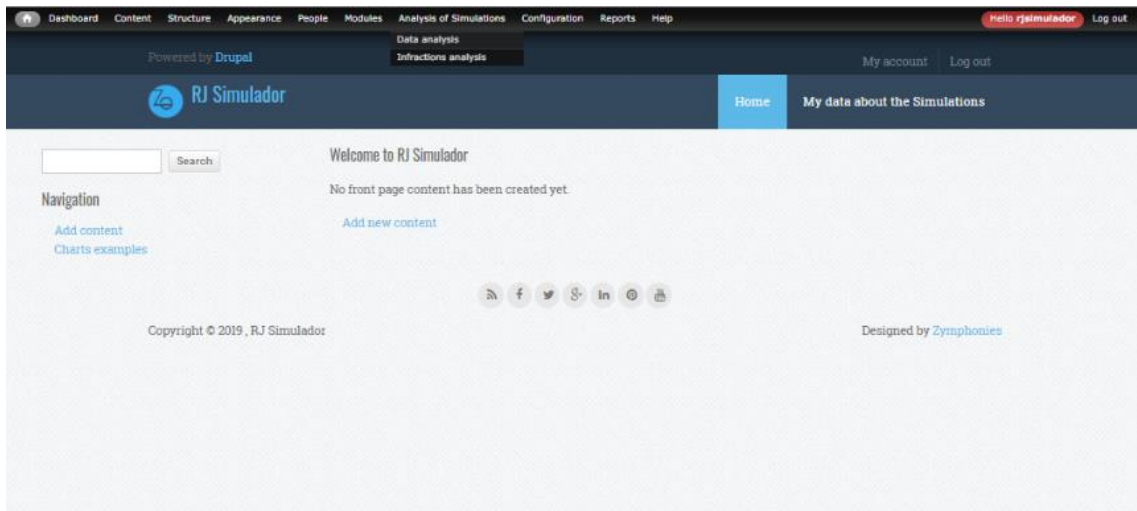


Figura 26: Página de principal de la plataforma web antes de la actualización.

Para el administrador, en la parte superior de esta página, aparece un menú de color negro que para el resto de usuarios no es visible. Dentro de ese menú, hay un apartado dedicado al módulo de la plataforma web, que redirige al administrador a páginas en las que se analizan los datos de las simulaciones de todos los usuarios.

En este desplegable se han modificado los títulos para que sean más acordes al contenido de las páginas a las que redirigen, como se puede apreciar si se comparan las figuras 26 y 27. Esta última es en la que se pueden apreciar los cambios realizados.

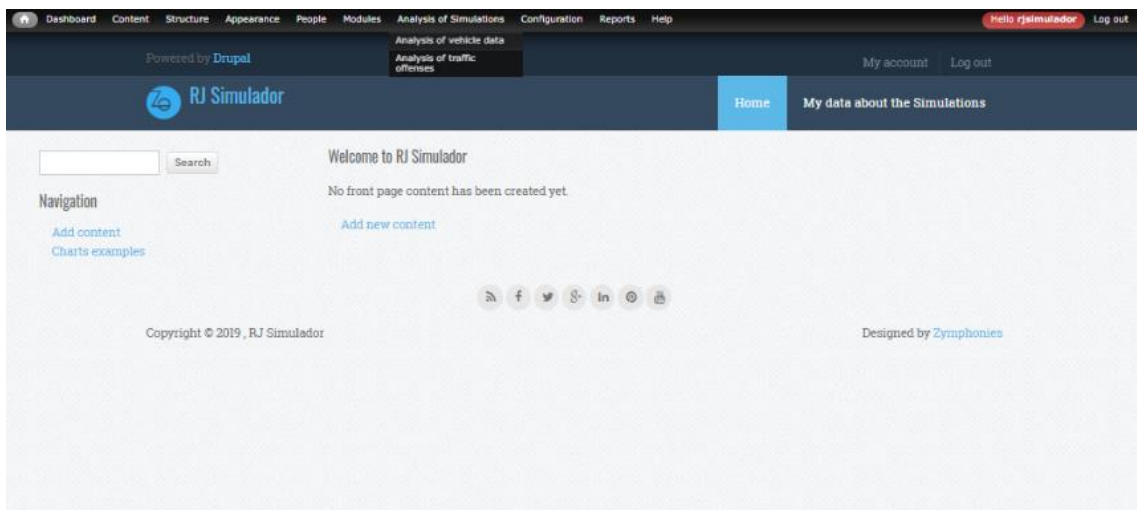


Figura 27: Página de principal de la plataforma web después de la actualización.

Los títulos del desplegable se han modificado para que se adecuen más al contenido de las páginas a las que llevan al administrador. De esta manera, “Data analysis” se ha reemplazado por “Analysis of vehicle data” e “Infractons analysis” se ha reemplazado por “Analysis of traffic offenses”.

A continuación, se empezarán a analizar las modificaciones de las páginas que son accesibles a través del desplegable y que son solo visibles para el administrador de la plataforma web.

4.3 Modificaciones del apartado Analysis of vehicle data

Se empezará analizando el apartado de la plataforma web que antes se denominaba “Data analysis” y tras la actualización se denomina “Analysis of vehicle data”. Este apartado de la plataforma ha sufrido numerosas modificaciones en esta actualización.

El apartado “Analysis of vehicle data” se subdivide en tres páginas. Estas páginas se analizarán por separado comentando las modificaciones realizadas. La primera modificación que se ha llevado a cabo en las tres páginas tiene que ver con los títulos de las mismas, que se han modificado de la siguiente manera: “General data analysis” ha pasado a ser “Simulation data”, “Data analysis by group” ahora se ha definido como “Simulation data per group” y “Data analysis by user” se ha reemplazado por “Simulation data per user”.

A continuación, se comenzarán a analizar las modificaciones realizadas en estas páginas, empezando por la principal.

4.3.1 Modificaciones de la página Simulation data

Esta es la página que aparece cuando se pulsa en el menú desplegable de la página principal “Analysis of vehicle data”. El título de esta página se ha modificado, como se comentó antes, ya que antes de la actualización se denominaba “General data analysis”, como se puede ver en la figura 28.

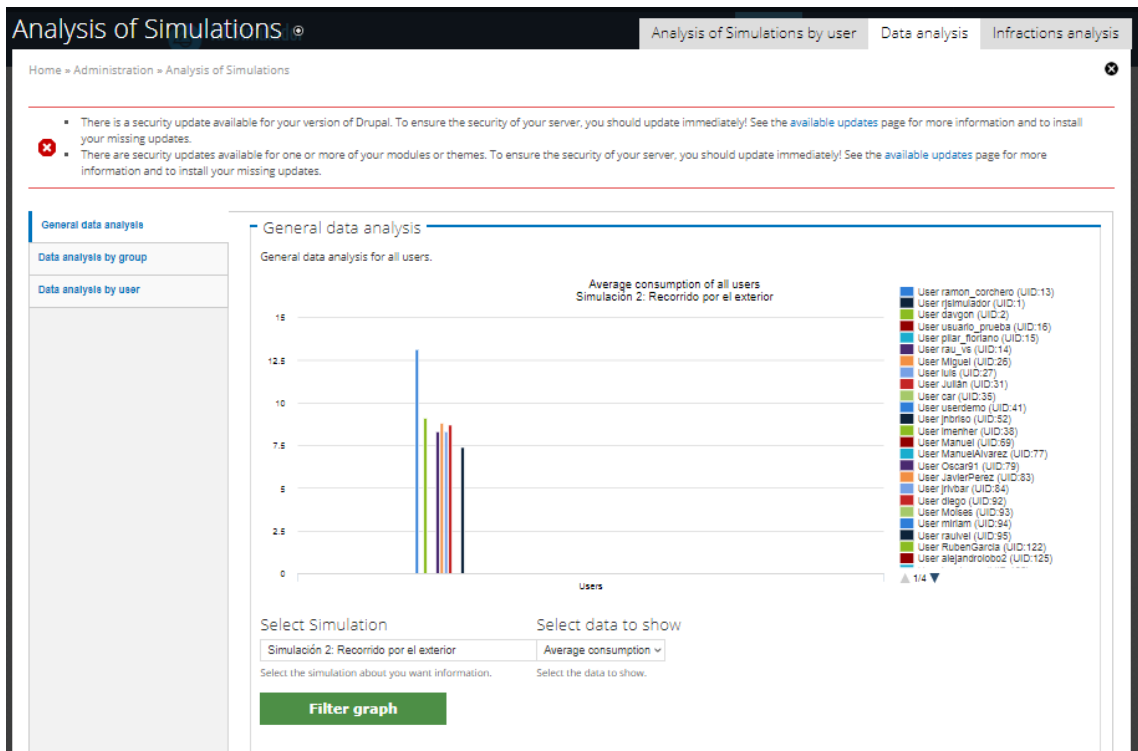


Figura 28: Página “Data analysis” de la plataforma web antigua, apartado “General data analysis”.

En esta página, se ha modificado el título de la gráfica, lo cual se comentará más adelante. También, se han modificado los selectores de datos que se sitúan en la parte inferior de la gráfica y que permiten filtrar los datos que se muestran en ella. Además, se ha añadido un nuevo selector para elegir el tipo de vehículo.

El primer selector es el que permite filtrar por el tipo de escenario en el que se ha realizado la simulación. Se ha modificado el título “Select Simulation” por “Select escenario”. También, se ha modificado el contenido del mismo poniendo todas las opciones en inglés y cambiando en cada una de ellas la palabra “simulación” por “escenario” que es la forma adecuada de referirse a estos datos. Los datos que contiene el selector de escenarios se extraen directamente desde la base de datos. Por eso, las modificaciones se han tenido que realizar directamente en ella y, para que estas sean permanentes, cuando se instale el módulo en cualquier entorno por primera vez, se ha tenido que modificar el fichero `rjsimulador.install`. Las modificaciones realizadas en el `hook rjsimulador_install()`, que es el encargado de cargar los datos en la base de datos cuando se instala el módulo por primera vez. Como se puede ver en

la figura 29, se ha incluido en este *hook* el nombre de todos los escenarios en inglés para tener un formato uniforme en la plataforma.

```
function rjsimulador_install() {
  db_insert( table: 'rjsim_infracciones')
  ->fields(array('id_infraccion','nombre_infraccion'))
  ->values(array(1, 'Striking a pedestrian'))
  ->values(array(2, 'Collision with other vehicle'))
  ->values(array(3, 'Drive off the road'))
  ->values(array(4, 'Exceed the speed limit'))
  ->values(array(5, 'Wrong exit in a roundabout'))
  ->values(array(6, 'Wrong path in a roundabout'))
  ->values(array(7, 'Opposite direction in the roundabout'))
  ->values(array(8, 'Wrong exit in a crossroads'))
  ->values(array(9, 'Failure to stop at a stop sign'))
  ->values(array(10, 'Failure to yield at a yield sign'))
  ->values(array(11, 'Failing to stop at a red traffic light'))
  ->values(array(12, 'Collision with street furniture'))
  ->values(array(13, 'Cross a solid line'))
  ->values(array(14, 'Low beams not switched driving under a tunnel'))
  ->values(array(15, 'Beams not switched at night'))
  ->values(array(16, 'Wrong turn in a crossroads'))
  ->values(array(17, 'Be closer than the minimum distance to the vehicle ahead'))
  ->values(array(18, 'Not switch on the corresponding turn light in a turn'))
  ->values(array(19, 'Not switch on the corresponding turn light in an overtaking'))
  ->values(array(20, 'Drive under the minimum speed'))
  ->values(array(21, 'Bus lane invasion'))
  ->values(array(22, 'Stop in yellow criss-cross lines'))
  ->values(array(23, 'Run over a cyclist'))
  ->values(array(24, 'Overtaking a cyclist without leaving the minimum safe distance'))
  ->values(array(25, 'Sudden breaking (not using engine braking)'))
  ->values(array(26, 'Not progressive acceleration (from 0 to 50km/h in less than 5 seconds)'))
  ->values(array(27, 'Sudden acceleration'))
  ->execute();
}

db_insert( table: 'rjsim_simulacion')
->fields(array('id_simulacion','nombre_simulacion','anchura_imagen','anchura_imagen_real',
| 'desplazamiento anchura_imagen','altura_imagen','altura_imagen_real','desplazamiento altura_imagen','imagen_fid'))
-> values (array (1, 'Scenario 1: Urban route',1200,1500,0,802,1000,0,1))
-> values (array (2, 'Scenario 2: Urban route outside',1200,1500,0,802,1000,0,2))
-> values (array (3, 'Scenario 3: Urban route from inside to outside',1200,1500,0,802,1000,0,3))
-> values (array (4, 'Scenario 4: Urban route from outside to inside',1200,1500,0,802,1000,0,4))
-> values (array (5, 'Scenario 5: Unguided Urban route',1200,1500,0,802,1000,0,5))
-> values (array (6, 'Scenario 6: M-30 ring road route',0,0,0,0,0,0))
-> values (array (7, 'Scenario 7: Through road route',1200,2170,0,1202,2170,0,6))
-> values (array (8, 'Scenario 8: Highway route',1200,24000,0,1205,24000,0,7))
-> values (array (9, 'Scenario 9: Urban route in Valladolid (Pilarica - San Juan)',1200,3000,325,1200,3000,800,20))
-> values (array (10, 'Scenario 10: Urban road, highway and through road in Valladolid',1200,7000,26,1330,8000,0,41))
-> values (array (11, 'Scenario 11: Urban route, highway, interurban road and through road in Valladolid',1080,6500,200,1600,9575,-150,43))
->execute();
}
```

Figura 29: Modificaciones realizadas en el hook *rjsimulador_install()* del fichero *rjsimulador.install*.

La descripción de este selector también se ha modificado. Antes, la descripción era “Select the simulation about you want information” y ahora es “Select the scenario”.

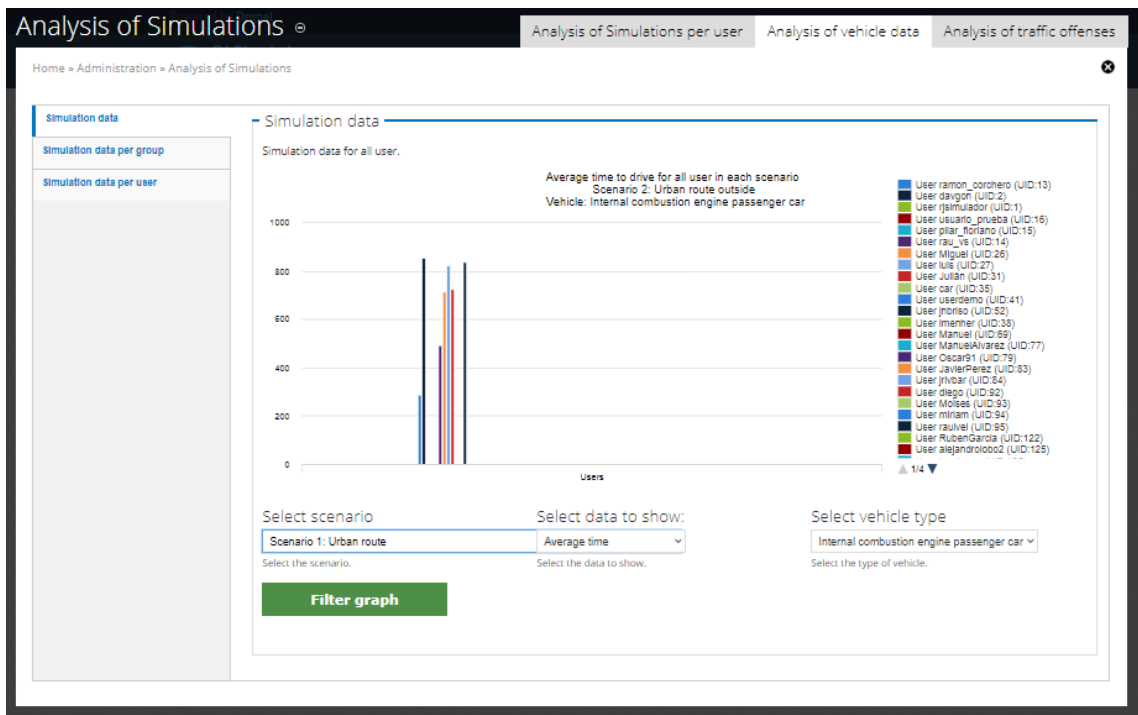


Figura 30: Página “Analysis of vehicle data” de la plataforma web, apartado “Simulation data”.

Con esta actualización, se ha añadido un nuevo selector que permite filtrar por el tipo vehículo con el que se han realizado las simulaciones como se puede apreciar en la figura 30, el título de este selector es “Select vehicle type” y la descripción que se ha incorporado a este selector es “Select the type of vehicle”. Para poder crear el nuevo selector en la página, se han tenido que incluir las líneas de código que se muestran en la figura 31, en la función `generate_form_data_general_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc`.

```
// Apartado del formulario que hace referencia al select tipo de vehiculo
$form['data_general_actions']['data_general_select_vehiculo'] = array (
  '#type' => 'select',
  '#title' => t( string: 'Select vehicle type'),
  '#options' =>
    array (
      t( string: 'Internal combustion engine passenger car'),
      t( string: 'Electric passenger car'),
      t( string: 'Internal combustion engine city bus'),
      t( string: 'Electric city bus'),
      t( string: 'Internal combustion engine truck'),
      t( string: 'Electric truck')
    ),
  '#default_value' => t( string: 'Tourism with combustion engine'),
  '#description' => t( string: 'Select the type of vehicle.'),
);
```

Figura 31: Código incluido en la función `generate_form_data_general_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para generar el selector de tipo de vehículo.

La modificación más relevante que se ha realizado en esta página se centra en el propio gráfico. Se ha modificado el título de la misma añadiendo las modificaciones de los nombres de los escenarios realizadas en el selector, los datos que se van a calcular y el tipo de vehículo con el que se han realizado

las simulaciones. Así, al analizar la gráfica, quedarán claros los datos que se muestran y que estamos analizando. Estos cambios se han llevado a cabo en la función `generate_form_data_general_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc`, como se muestra en las figuras 32 y 33. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector, ya que las otras modificaciones que influyen sobre el título de la gráfica relacionada con los datos a calcular y el escenario son textuales.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['data_general_select_vehiculo']) ?
    $form_state['values']['data_general_select_vehiculo'] : 0;
```

Figura 32: Código incluido en la función `generate_form_data_general_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para obtener los datos del selector de tipo de vehículo.

En función del valor que se recupera en la variable `$vehiculo`, se genera el título para la gráfica, gracias al código que se muestra en la figura 33.

```
// Generamos el título relacionado con el tipo de vehículo
$tituloGrafico .="<br>";
switch ($vehiculo) {
    case 0:
        $tituloGrafico .= ' Vehicle: Internal combustion engine passenger car';
        $vehi = 1;
        break;
    case 1:
        $tituloGrafico .= ' Vehicle: Electric passenger car';
        $vehi = 2;
        break;
    case 2:
        $tituloGrafico .= ' Vehicle: Internal combustion engine city bus';
        $vehi = 3;
        break;
    case 3:
        $tituloGrafico .= ' Vehicle: Electric city bus';
        $vehi = 4;
        break;
    case 4:
        $tituloGrafico .= ' Vehicle: Internal combustion engine truck';
        $vehi = 5;
        break;
    case 5:
        $tituloGrafico .= ' Vehicle: Electric truck';
        $vehi = 6;
        break;
}
```

Figura 33: Código incluido en la función `generate_form_data_general_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para generar el título en función del selector de tipo de vehículo.

Para poder aplicar los cambios del título en la gráfica que muestra los datos, se ha modificado la parte del formulario que hace referencia al título de la gráfica en la función `generate_form_data_general_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc`, incluyendo como título de la gráfica la variable `$tituloGrafico` como se muestra en la figura 34. Esta variable que se ha ido cargando con anterioridad de datos referentes al tipo de simulación y vehículo.

```

// Generamos el gráfico
$form['data_general_chart'] = array(
    '#type' => 'chart',
    '#chart_type' => 'column',
    '#chart_library' => 'highcharts',
    '#title' => $tituloGrafico,
    '#prefix' => '<div id="div-data-general-chart">',
    '#suffix' => '</div>',
    '#xaxis' =>
        array(
            '#type' => 'chart_xaxis',
            '#labels' => $labelAxisX,
        ),
);

```

Figura 34: Código incluido en la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el título de la gráfica.

Para que el funcionamiento de este gráfico a la hora de analizar los datos sea correcto, se ha tenido que modificar la codificación de la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc, como se muestra en la figura 35. Se ha añadido una nueva función de la que se hablará a continuación.

```

// Recuperamos el cálculo del dato para todos los usuarios que tienen partidas en el simulador para este tipo vehiculo
foreach ($gestorSimulaciones->getListatodosUsuarios() as $usuario) {
    // Datos medios por simulación
    $dataUsuario = round($usuario->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi)
        ->calculateData(new CalculateAverageData($tipoDatoCalcular)), precision: 2);

    $form['data_general_chart']['data_general_user_' . $usuario->getUId()] =
        array(
            '#type' => 'chart_data',
            '#title' => t( string: 'User @userName (UID:@userUID)', array(
                '@userName' => $usuario->getName(),
                '@userUID' => $usuario->getUId()
            )),
            '#data' => array($dataUsuario),
        );
}

```

Figura 35: Código modificado en la función generate_form_data_general_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos para el gráfico y para pintar los datos en el gráfico.

Como se puede apreciar en la figura 35 y como se ha comentado anteriormente, ha sido necesario crear una nueva función en el fichero UsuarioSimulacion.php para recuperar los datos necesarios para la gráfica en función del vehículo seleccionado en el selector. Esta nueva función filtra las simulaciones en función del id del escenario seleccionado y el id del vehículo seleccionado. Se puede ver la codificación de la misma en la figura 36.

```

/**
 * Método que devuelve todas las partidas del usuario para una simulación en concreto y para un tipo de vehículo.
 * @param int $idSimulation El id del escenario para el que recuperar las partidas del usuario.
 * @param int $vehiculo El id del tipo vehículo.
 * @return ListaPartidas Lista de todas las partidas de ese escenario realizadas con un vehículo en concreto.
 */
public function retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo) {
    $listaPartidas = new ListaPartidas();
    $listaPartidasVehi = new ListaPartidas();

    foreach ($this->getListaSimulaciones() as $simulacion) {
        if ($simulacion->getIdSimulacion() == $idSimulation) {
            $listaPartidas->mergeList($simulacion->getListaPartidas());
            break;
        }
    }
    foreach ($listaPartidas as $partida) {
        if ($partida->getTipoVehiculo() == $vehiculo) {
            $listaPartidasVehi->add($partida);
        }
    }

    return $listaPartidasVehi;
}

```

Figura 36: Nueva función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) dentro del fichero UsuarioSimulacion.php.

Con todas estas modificaciones, se ha conseguido un funcionamiento óptimo de esta página de la plataforma consiguiendo los resultados que se buscaban con esta actualización.

4.3.2 Modificaciones de la página Simulation data per group

La segunda de estas páginas de la cual se van a analizar las modificaciones es la que se titulaba “Data analysis by group” antes de la actualización, como se puede ver en la figura 37 y que tras la actualización se denomina “Simulation data per group”. Este cambio se puede apreciar en la figura 38.

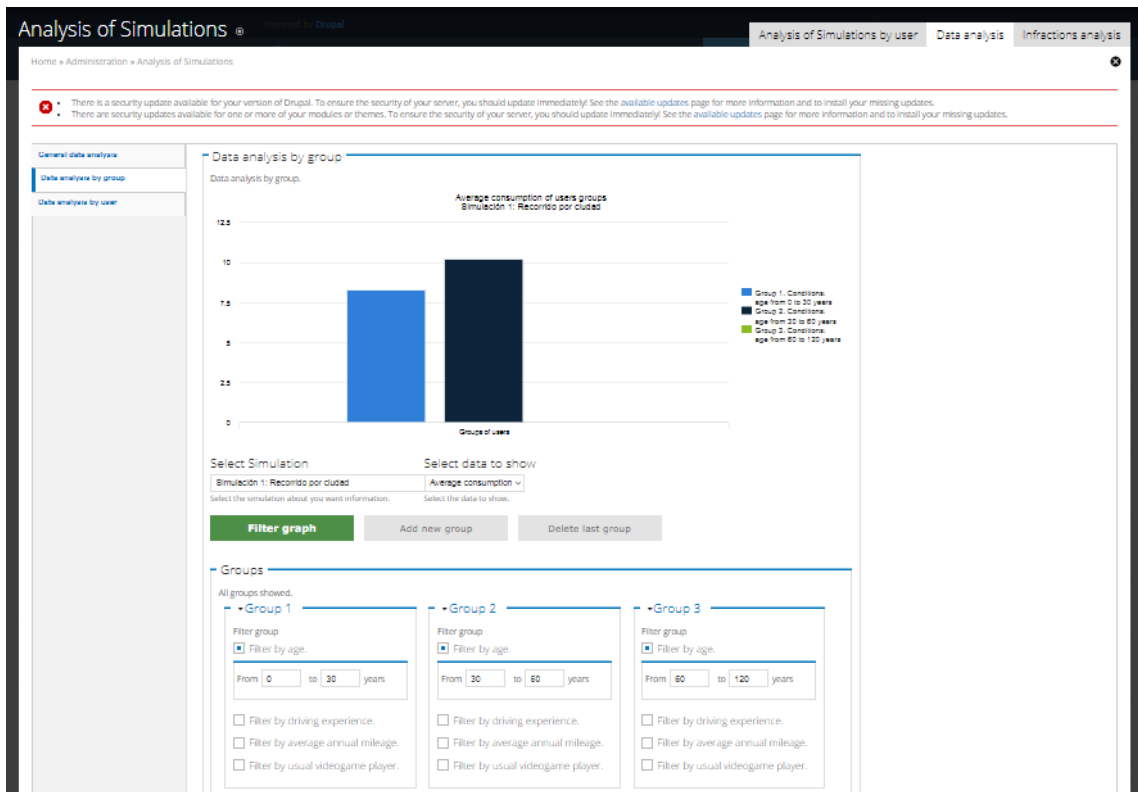


Figura 37: Página “Data analysis” de la plataforma web antigua, apartado “Data analysis by group”.

Como se ha comentado sobre las modificaciones de la página anterior, se han modificado los selectores de datos que se sitúan en la parte inferior de la gráfica y que permiten filtrar los datos que va a mostrar la gráfica. Además, se ha añadido un nuevo selector para elegir el tipo de vehículo.

El primero de los selectores permite filtrar por el tipo de escenario en el que se ha realizado la simulación. En él, se ha modificado el título “Select Simulation” por “Select escenario”. También, se ha modificado el contenido del mismo poniendo todas las opciones en inglés y cambiando en cada una de ellas “simulación” por “escenario”, que es la forma adecuada de referirse a estos datos. Como se comentó en el apartado anterior, el contenido del selector de escenarios se extrae directamente desde la base de datos. Por eso, las modificaciones se han tenido que realizar directamente en ella y, para que estas sean permanentes, cuando se instale el módulo en cualquier entorno por primera vez, se ha tenido que modificar el fichero `rjsimulador.install`. Las modificaciones realizadas en el `hook rjsimulador_install()`, que es el encargado de cargar los datos en la base de datos cuando se instala el módulo

por primera vez, se muestran en la figura 29 del apartado anterior e influyen a toda la plataforma. La descripción de este selector también se ha modificado como en la página que se analizó en el apartado anterior.

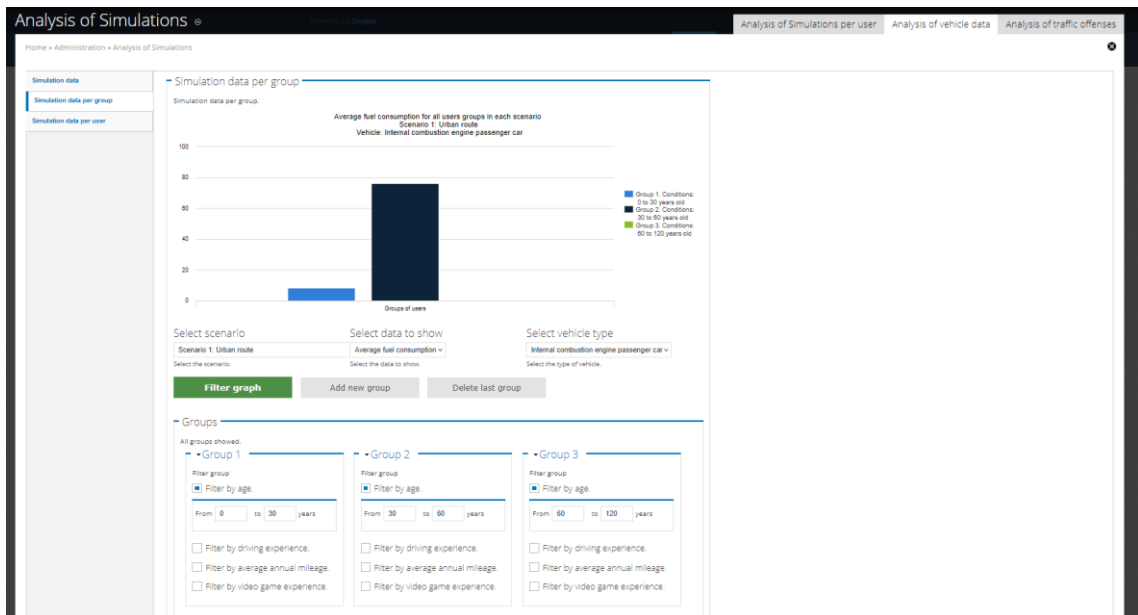


Figura 38: Página “Analysis of vehicle data” de la plataforma web, apartado “Simulation data per group”.

En esta página también se ha añadido el nuevo selector que permite filtrar por el tipo de vehículo con el que se han realizado las simulaciones, que era uno de los objetivos de esta actualización. El título de este selector es “Select vehicle type” y la descripción que se ha incorporado a este selector es “Select the type of vehicle”. Para poder crear el selector en esta página, se han tenido que incluir las líneas de código que se muestran en la figura 39 en la función generate_form_data_group_analysis(), dentro del fichero rjsimulador.data_analysis.forms.inc.

```
//Apartado del formulario que hace referencia al selector tipo de vehiculo
$form['data_group_container']['data_group_actions']['data_group_select_vehiculo'] = array(
  '#type' => 'select',
  '#title' => t( string: 'Select vehicle type'),
  '#options' =>
    array(
      t( string: 'Internal combustion engine passenger car'),
      t( string: 'Electric passenger car'),
      t( string: 'Internal combustion engine city bus'),
      t( string: 'Electric city bus'),
      t( string: 'Internal combustion engine truck'),
      t( string: 'Electric truck')
    ),
  '#default_value' => t( string: 'Tourism with combustion engine'),
  '#description' => t( string: 'Select the type of vehicle.'),
);
```

Figura 39: Código incluido en la función generate_form_data_group_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el selector de tipo de vehículo.

La modificación más relevante que se ha realizado en esta página, al igual que en la anterior, se centra en el propio gráfico. Se ha modificado el título de la misma añadiendo los cambios de los nombres de los escenarios realizados en el selector, los datos que se van a calcular y el tipo de vehículo con el que

se han realizado las simulaciones. Así, al analizar la gráfica, quedarán claro los datos que se muestran y que estamos analizando. Estos cambios se han llevado a cabo en la función `generate_form_data_group_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc`, como se muestra en las figuras 40 y 41. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector, ya que las otras modificaciones que influyen sobre el título de la gráfica, relacionadas con los datos a calcular y el escenario, son textuales.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['data_group_select_vehiculo']) ?
    $form_state['values']['data_group_select_vehiculo'] : 0;
```

Figura 40: Código incluido en la función `generate_form_data_group_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para obtener los datos del selector de vehículos.

En función del valor que se recupera en la variable `$vehiculo`, se genera el título para la gráfica dependiendo id del vehículo seleccionado en el selector, gracias al código que se muestra en la figura 41.

```
// Generamos el título relacionado con el tipo de vehículo
$titleGrafico .= "<br>";
switch ($vehiculo) {
    case 0:
        $titleGrafico .= ' Vehicle: Internal combustion engine passenger car';
        $vehi = 1;
        break;
    case 1:
        $titleGrafico .= ' Vehicle: Electric passenger car';
        $vehi = 2;
        break;
    case 2:
        $titleGrafico .= ' Vehicle: Internal combustion engine city bus';
        $vehi = 3;
        break;
    case 3:
        $titleGrafico .= ' Vehicle: Electric city bus';
        $vehi = 4;
        break;
    case 4:
        $titleGrafico .= ' Vehicle: Internal combustion engine truck';
        $vehi = 5;
        break;
    case 5:
        $titleGrafico .= ' Vehicle: Electric truck';
        $vehi = 6;
        break;
}
```

Figura 41: Código incluido en la función `generate_form_data_group_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para generar el título en función del selector.

Para poder aplicar los cambios del título en la gráfica que muestra los datos, se ha modificado la parte del formulario que hace referencia al título de la gráfica en la función `generate_form_data_group_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc`. Como se muestra en la figura 42, el procedimiento es similar al apartado anterior.

```

//Generamos el grafico
$form['data_group_container']['data_group_chart'] = array(
    '#type' => 'chart',
    '#chart_type' => 'column',
    '#chart_library' => 'highcharts',
    '#title' => $tituloGrafico,
    '#weight' => 0,
    '#xaxis' =>
        array(
            '#type' => 'chart_xaxis',
            '#labels' => $labelAxisX,
        ),
);

```

Figura 42: Código incluido en la función `generate_form_data_group_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para generar el título de la gráfica.

Para que el funcionamiento a la hora de analizar los datos sea correcto, se ha tenido que modificar la codificación de la función `generate_form_data_group_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc`, como se muestra en la figuras 43 y 44, siendo este cambio uno de los más relevantes.

```

// Recuperamos los datos de la lista de usuarios filtrada
$retriever = new ListaUsuariosDataRetriever($listaUsuarios);

// Datos medios por simulación del grupo filtrado
$dataGroup = round($retriever->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi)
->calculateData(new CalculateAverageData($tipoDatoCalcular), precision: 2);

```

Figura 43: Código modificado en la función `generate_form_data_general_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para obtener los datos para el gráfico.

```

$form['data_group_container']['data_group_chart']['data_group_' . $index] =
    array(
        '#type' => 'chart_data',
        '#title' => t($tituloSustitucion, $arraySustitucion),
        '#data' => array($dataGroup),
    );

```

Figura 44: Código modificado en la función `generate_form_data_general_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para pintar los datos en el gráfico.

Como se puede apreciar en la figura 43, ha sido necesario crear una nueva función en el fichero `ListaUsuariosDataRetriever.php` para recuperar los datos necesarios para la gráfica para un grupo concreto de usuarios. Esta nueva función se ha creado como se muestra en la figura 45 y es muy similar a la que se utilizó en el apartado anterior. En la parte inferior de esta página es donde se crean los grupos para comparar los datos que se recuperan con la función de la figura 45. Se ha modificado el título de la casilla de verificación “Filter by usual videogame player” por “Filter by video game experience”, para que se ajuste más a la funcionalidad que aporta a la hora de crear los grupos de usuarios.

```

/**
 * Método que devuelve todas las partidas de los usuarios de la lista para una simulación en concreto y
 * un tipo de vehículo.
 * @param int $idSimulacion El id de la simulación para la que recuperar todas las partidas.
 * @param int $vehiculo El id del vehículo.
 * @return ListaPartidas Lista de todas las partidas de la simulación para los usuarios de la lista.
 */
public function retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo) {
    $listaPartidas = new ListaPartidas();
    $listaPartidasGrupo = new ListaPartidas();

    foreach ($this->getListaUsuarios() as $usuario) {
        $listaPartidas->mergeList($usuario->retrieveAllPartidasByIdSimulacion($idSimulation));
    }
    foreach ($listaPartidas as $partida) {
        if ($partida->getTipoVehiculo() == $vehiculo) {
            $listaPartidasGrupo->add($partida);
        }
    }
    return $listaPartidasGrupo;
}

```

Figura 45: Nueva función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) dentro del fichero ListaUsuariosDataRetriever.php.

Con todas estas modificaciones, se ha conseguido un funcionamiento óptimo de esta página de la plataforma que se utiliza para comparar grupos de usuarios.

4.3.3 Modificaciones de la página Simulation data per user

Otra de las páginas de la que se van a analizar las modificaciones es la que se titulaba “Data analysis by user” antes de la actualización, como se puede observar en la figura 46, y que, tras la actualización, se denomina “Simulation data per user”, como se puede observar en la figura 47.

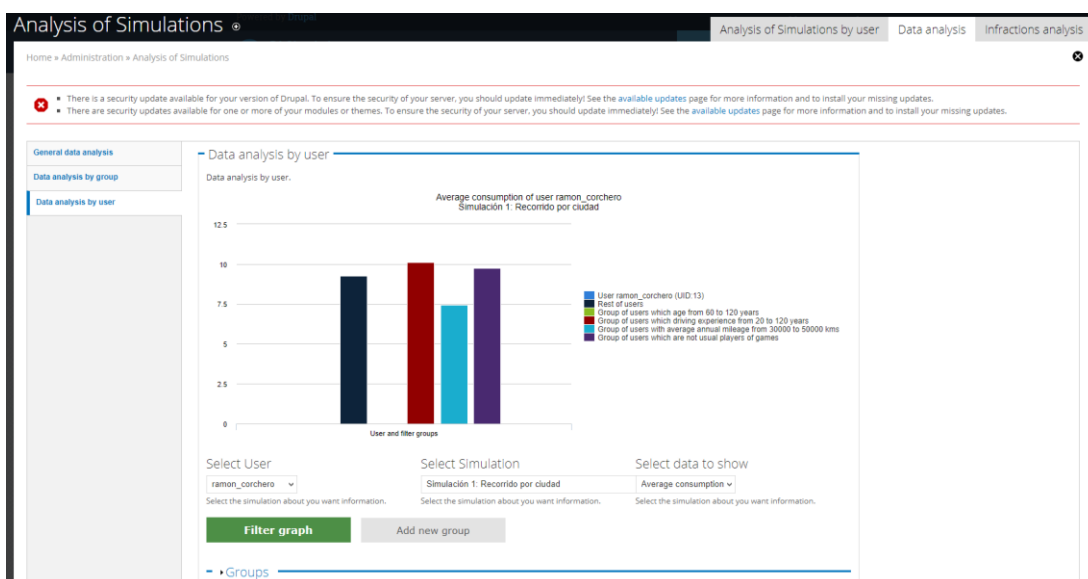


Figura 46: Página “Data analysis” de la plataforma web antigua, apartado “Data analysis by user”.

Como se ha comentado sobre las modificaciones de las páginas anteriores, se han modificado los selectores de datos que se sitúan en la parte inferior de la gráfica y que permiten filtrar los datos que se van mostrar en ella. Además, se ha añadido un nuevo selector para elegir el tipo de vehículo.

Como en los apartados anteriores ha sido necesario modificar los selectores siguiendo el mismo criterio, el primero de los selectores que permite filtrar por el tipo de escenario en el que se ha realizado la simulación. Se ha modificado el título como anteriormente “Select Simulation” por “Select scenario”. Como en otros casos también, se ha modificado el contenido del mismo poniendo todas las opciones en inglés y cambiando en cada una de ellas “simulación” por “scenario”, que es la forma adecuada de referirse a estos datos. Como se ha comentado en los apartados anteriores, el contenido del selector de escenarios se extrae directamente desde la base de datos. El cambio sobre ella, y que afecta a este tipo de selectores, es el que se ha comentado en apartados anteriores y que se puede apreciar en la figura 29. La descripción de este selector también se ha modificado como en las páginas que se analizaron en los apartados anteriores.

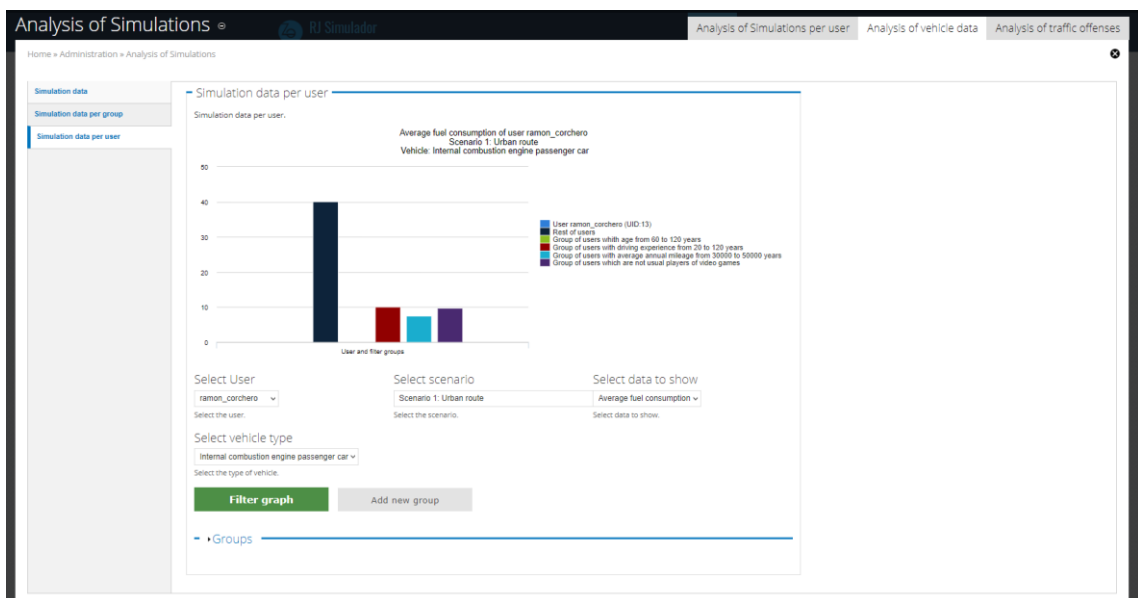


Figura 47: Página “Analysis of vehicle data” de la plataforma web, apartado “Simulation data per user”.

En esta página también se ha añadido el nuevo selector que permite filtrar las simulaciones por el tipo vehículo con el que se han realizado, el título de este selector es “Select vehicle type” y la descripción que se ha incorporado a este selector es “Select the type of vehicle”. Para poder crear el selector en esta página, se han tenido que incluir las líneas de código que se muestran en la figura 48 en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc.

```

//Apartado del formulario que hace referencia al selector tipo de vehiculo
$form['data_user_container']['data_user_actions']['data_user_select_vehiculo'] = array(
    '#type' => 'select',
    '#title' => t( string: 'Select vehicle type'),
    '#options' =>
        array(
            t( string: 'Internal combustion engine passenger car'),
            t( string: 'Electric passenger car'),
            t( string: 'Internal combustion engine city bus'),
            t( string: 'Electric city bus'),
            t( string: 'Internal combustion engine truck'),
            t( string: 'Electric truck')
        ),
    '#default_value' => t( string: 'Tourism with combustion engine'),
    '#description' => t( string: 'Select the type of vehicle.'),
);

```

Figura 48: Código incluido en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el selector.

La modificación más relevante que se ha realizado en esta página, al igual que en los anteriores apartados, se centra en el propio gráfico. Se ha modificado el título de la misma añadiendo las modificaciones de los nombres de los escenarios realizadas en el selector, los datos que se van a calcular y el tipo de vehículo con el que se han realizado las simulaciones. Así, al analizar la gráfica, quedarán claro los datos que se muestran y que estamos analizando. Estos cambios se han llevado a cabo en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc, como se muestra en las figuras 49 y 50. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector, ya que las otras modificaciones, que influyen sobre el título de la gráfica, relacionadas con los datos a calcular y el escenario son textuales.

```

// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['data_user_select_vehiculo']) ?
    $form_state['values']['data_user_select_vehiculo'] : 0;

```

Figura 49: Código incluido en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos del selector de vehículos.

En función del valor que se recupera en la variable \$vehiculo se genera el título para la gráfica, gracias al código que se muestra en la figura 50.

```

// Generamos el titulo relacionado con el tipo de vehiculo
$tituloGrafico .="<br>";
switch ($vehiculo) {
case 0:
    $tituloGrafico .= ' Vehicle: Internal combustion engine passenger car';
    $vehi = 1;
    break;
case 1:
    $tituloGrafico .= ' Vehicle: Electric passenger car';
    $vehi = 2;
    break;
case 2:
    $tituloGrafico .= ' Vehicle: Internal combustion engine city bus';
    $vehi = 3;
    break;
case 3:
    $tituloGrafico .= ' Vehicle: Electric city bus';
    $vehi = 4;
    break;
case 4:
    $tituloGrafico .= ' Vehicle: Internal combustion engine truck';
    $vehi = 5;
    break;
case 5:
    $tituloGrafico .= ' Vehicle: Electric truck';
    $vehi = 6;
    break;
}

```

Figura 50: Código incluido en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el título en función del selector.

Para poder aplicar los cambios del título en la gráfica que muestra los datos, se ha modificado la parte del formulario que hace referencia al título de la gráfica en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc. Como se muestra en la figura 51, el procedimiento es similar a los apartados anteriores.

```

//Generamos el grafico
$form['data_user_container']['data_user_chart'] = array (
    '#type' => 'chart',
    '#chart_type' => 'column',
    '#chart_library' => 'highcharts',
    '#title' => $tituloGrafico,
    '#weight' => 0,
    'xaxis' =>
        array (
            '#type' => 'chart_xaxis',
            '#labels' => $labelAxisX,
        ),
);

```

Figura 51: Código incluido en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para generar el título de la gráfica.

Para que el funcionamiento a la hora de analizar los datos sea correcto, se ha tenido que modificar la codificación de la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc, como se muestra en las siguientes figuras. En la figura 52 se muestra el código con el que se recuperan los datos de las simulaciones en función del usuario, que se elige en el selector de usuarios, el escenario seleccionado en el selector de escenarios y el tipo de vehículo seleccionado.

```

// Seteamos el usuario actual en el Gestor de Simulaciones
$gestorSimulaciones->setUsuarioActual($usuarioActual);
// Recuperamos los datos del usuario actual
$dataUser = round($usuarioActual->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi)
->calculateData(new CalculateAverageData($tipoDatoCalcular), precision: 2);

```

Figura 52: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos del usuario seleccionado.

Como se puede apreciar en la figura 52, se utiliza la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo)` creada en el fichero `UsuarioSimulacion.php` para recuperar los datos necesarios para la gráfica en función del vehículo seleccionado en el selector. De esta nueva función se habló en apartados anteriores y se mostró en la figura 36. Por otro lado, en la figura 53 se puede ver que los datos que se han recuperado en la figura 52 para el usuario seleccionado, se utilizan para representarlos en el gráfico.

```
$form['data_user_container']['data_user_chart']['data_user_actual'] =
    array(
        '#type' => 'chart_data',
        '#title' => t( string: 'User @userName (UID:@userUID)', array(
            '@userName' => $usuarioActual->getName(),
            '@userUID' => $usuarioActual->getUId()
        )),
        '#data' => array($dataUser),
    );
```

Figura 53: Código modificado en la función `generate_form_data_user_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para pintar los datos del usuario seleccionado.

En la figura 54 se ven las líneas de código modificadas para recuperar los datos de las simulaciones del grupo de resto de los usuarios, que son todos menos el usuario que se seleccionó en el selector de usuarios. Como se ha comentado anteriormente, estos datos se recuperan en función del vehículo y el escenario seleccionado.

```
// Recuperamos los datos del grupo del resto de usuarios
$listadoRestoUsuarios = $gestorSimulaciones->getListadoTodosUsuariosExceptoActual();
// Instanciamos el recuperador de datos
$retreiver = new ListaUsuariosDataRetriever($listadoRestoUsuarios);
// Recuperamos los datos de todos los usuarios
$dataRestoUsuarios = round($retreiver->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi
->calculateData(new CalculateAverageData($tipoDatoCalcular)), precision: 2);
```

Figura 54: Código modificado en la función `generate_form_data_user_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para obtener los datos del resto de usuarios.

En el código de la figura 54 ha sido necesario utilizar la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo)` del fichero `ListaUsuariosDataRetriever.php` para recuperar los datos necesarios para la gráfica de un grupo concreto de usuarios. De esta nueva función ya se habló en apartados anteriores y se puede ver en la figura 45. Si se analiza la figura 55, se aprecia que los datos que se han recuperado para el grupo resto de usuarios, se utilizan para representar los datos en el gráfico para este grupo.

```
$form['data_user_container']['data_user_chart']['data_user_rest_of_user'] =
    array(
        '#type' => 'chart_data',
        '#title' => t( string: 'Rest of users'),
        '#data' => array($dataRestoUsuarios),
    );
```

Figura 55: Código modificado en la función `generate_form_data_user_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para pintar los datos del resto de usuarios.

La figura 56 muestra las líneas de código modificadas para recuperar los datos de las simulaciones del grupo de usuarios que tienen la misma edad que el usuario analizado. Como se ha comentado anteriormente, estos datos se recuperan en función del vehículo y el escenario seleccionado.

```
// Recuperamos los datos del grupo del resto de usuarios que está en el mismo grupo de edad que el usuario analizado
$listaRestoUsuariosMismoGrupoEdad = $listaRestoUsuarios->filterBy(new FilterByInterval($usuarioActual->getGrupoEdad(),
    paramField: FilterByInterval::AGE));
$retreiver->setListaUsuarios($listaRestoUsuariosMismoGrupoEdad);
// Recuperamos los datos de todos los usuarios
$dataRestoUsuarioMismoGrupoEdad = round($retreiver->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi)
->calculateData(new CalculateAverageData($tipoDatoCalcular)), precision: 2);
```

Figura 56: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos de los usuarios con la misma edad que el usuario a analizar.

Se utiliza la función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) del fichero ListaUsuariosDataRetriever.php para recuperar los datos necesarios para la gráfica de un grupo concreto de usuarios, como se puede ver en la figura 56. De esta nueva función ya se habló con anterioridad y se puede ver en la figura 45. Observando la figura 57, se aprecia que los datos que se han recuperado para el grupo de usuarios que tienen la misma edad que el usuario a analizar, se utilizan para representar los datos en el gráfico para este grupo.

```
$form['data_user_container']['data_user_chart']['data_user_same_age_group'] =
array(
    '#type' => 'chart_data',
    //Modificado
    '#title' => t('string: Group of users with @userGroup', array('@userGroup' => $usuarioActual->getGrupoEdad( asText: TRUE))),
    '#data' => array($dataRestoUsuarioMismoGrupoEdad),
);
```

Figura 57: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para pintar los datos de los usuarios con la misma edad que el usuario a analizar.

En la figura 58 se observan las líneas de código modificadas para recuperar los datos de las simulaciones del grupo de usuarios que tienen la misma experiencia de conducción que el usuario analizado. Estos datos se recuperan en función del id del vehículo seleccionado en el selector y el id del escenario seleccionado en el selector gracias a la nueva función creada retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo), como se comentará a continuación.

```
// Recuperamos los datos del grupo del resto de usuarios que está en el mismo grupo de experiencia conductora que el usuario analizado
$listaRestoUsuariosMismoGrupoExperiencia = $listaRestoUsuarios->filterBy(new FilterByInterval($usuarioActual->getGrupoExperiencia(), paramField: FilterByInterval::DRIVING_EXPERIENCE));
$retreiver->setListaUsuarios($listaRestoUsuariosMismoGrupoExperiencia);
// Recuperamos los datos de todos los usuarios
$dataRestoUsuarioMismoGrupoExperiencia = round($retreiver->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi)
->calculateData(new CalculateAverageData($tipoDatoCalcular)), precision: 2);
```

Figura 58: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms, para obtener los datos de los usuarios con la misma experiencia conduciendo que el usuario a analizar.

La nueva función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) creada

en el fichero ListaUsuariosDataRetriever.php ha sido necesaria para recuperar los datos para la gráfica de un grupo concreto de usuarios, como se puede ver en la figura 58. El código de esta función se puede ver en la figura 45, como se ha hablado en apartados anteriores. La figura 59 muestra que los datos que se han recuperado para el grupo de usuarios que tienen la misma experiencia conduciendo que el usuario a analizar, se utilizan para pintar el gráfico para este grupo.

```
$form['data_user_container']['data_user_chart']['data_user_same_driving_experience_group'] =
array(
    '#type' => 'chart_data',
    //Modificado
    '#title' => t( string: 'Group of users with @userGroup', array('@userGroup' => $usuarioActual->getGrupoExperiencia( asText: TRUE))),
    '#data' => array($dataRestoUsuarioMismoGrupoExperiencia),
);
```

Figura 59: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms, para representar los datos de los usuarios con la misma experiencia conduciendo que el usuario a analizar.

La figura 60 muestra las nuevas líneas de código utilizadas para recuperar los datos de las simulaciones del grupo de usuarios que tienen el mismo número de kilómetros recorridos que el usuario analizado. Estos datos se recuperan en función del vehículo y el escenario seleccionado, como se viene haciendo para el resto de grupos anteriores.

```
// Recuperamos los datos del grupo del resto de usuarios que está en el mismo grupo de kilometraje medio que el usuario analizado
$listRestoUsuariosMismoGrupoKilometraje = $listaRestoUsuarios->filterBy(new FilterByInterval($usuarioActual->getGrupoKilometrajeMedioAnual(), $paramField: FilterByInterval::AVERAGE_ANNUAL_MILEAGE));
$retriever->setListaUsuarios($listaRestoUsuariosMismoGrupoKilometraje);
// Recuperamos los datos de todos los usuarios
$dataRestoUsuarioMismoGrupoKilometraje = round($retriever->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi
->calculateData(new CalculateAverageData($tipoDataCalculator), $precision: 2);
```

Figura 60: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos de los usuarios con el mismo número de kilómetros recorridos que el usuario a analizar.

Como para los grupos creados anteriores es necesario utilizar la función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) del fichero ListaUsuariosDataRetriever.php, esta función se utiliza para recuperar los datos para la gráfica de un grupo concreto de usuarios, como se puede ver en la figura 60. El código de esta función se puede ver en la figura 45. En la figura 61 se aprecia que los datos que se han recuperado para el grupo de usuarios con el mismo número de kilómetros recorridos que el usuario a analizar, se utilizan para representar el gráfico para este grupo, asignando el valor recuperado a la etiqueta #data de este gráfico.

```
$form['data_user_container']['data_user_chart']['data_user_same_average_annual_mileage_group'] =
array(
    '#type' => 'chart_data',
    //Modificado
    '#title' => t( string: 'Group of users with @userGroup', array('@userGroup' => $usuarioActual->getGrupoKilometrajeMedioAnual( asText: true))),
    '#data' => array($dataRestoUsuarioMismoGrupoKilometraje),
);
```

Figura 61: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para pintar los datos de los usuarios con el mismo número de kilómetros recorridos que el usuario a analizar.

En la figura 62 se observa el código utilizado para recuperar los datos de las simulaciones del grupo de usuarios que tienen la misma experiencia en

videojuegos que el usuario analizado. Los datos se obtienen filtrando las simulaciones por el vehículo y el escenario seleccionado.

```
// Recuperamos los datos del grupo del resto de usuarios que está en es del mismo tipo de jugador habitual que el usuario analizado
$listaRestoUsuariosMismoTipoJugador = $listaRestoUsuarios
->filterBy(new FilterByEquality(array($usuarioActual->isUsualVideogamePlayer()), paramField: FilterByEquality::USUARIO_USUAL_PLAYER));
$retriever->setListaUsuarios($listaRestoUsuariosMismoTipoJugador);
// Recuperamos los datos de todos los usuarios
$dataRestoUsuarioMismoTipoJugador = round($retriever->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi)
->calculateData(new CalculateAverageData($tipoDatoCalcular)), precision: 2);
```

Figura 62: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos de los usuarios con la misma experiencia en videojuegos que el usuario a analizar.

Para el grupo creado con la misma experiencia en videojuegos que el usuario a analizar es necesario utilizar la función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) del fichero ListaUsuariosDataRetriever.php. Esta función se utiliza para recuperar los datos para la gráfica de un grupo concreto de usuarios, como se puede ver en la figura 62. El código de esta función se puede ver en la figura 45. En la figura 63 se aprecia que los datos que se han recuperado para el grupo de usuarios con la misma experiencia en videojuegos que el usuario a analizar, se utilizan para pintar el gráfico para este grupo.

```
$form['data_user_container']['data_user_chart']['data_user_same_usual_videogame_player_group'] =
array (
  '#type' => 'chart_data',
  //Modificado
  '#title' => t( string: 'Group of users which @userGroup usual players of video games',
  array('@userGroup' => $usuarioActual->isUsualVideogamePlayer( asBoolean: TRUE) ? t( string: "are") : t( string: "are not")),
  '#data' => array($dataRestoUsuarioMismoTipoJugador),
);
```

Figura 63: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para pintar los datos de los usuarios con la misma experiencia en videojuegos que el usuario a analizar.

En la parte inferior de esta página se pueden crear grupos para comparar sus datos con los ya existentes y con el usuario a analizar. En esta parte se ha modificado el título de la casilla “Filter by usual videogame player” por “Filter by video game experience”, para que se ajuste más a la funcionalidad que aporta a la hora de crear los grupos de usuarios. Una vez que se han utilizado los selectores para crear un nuevo grupo, en la figura 64 se observa el código utilizado para recuperar los datos de las simulaciones del grupo de usuarios creado. Los datos se obtienen filtrando las simulaciones por el vehículo y el escenario seleccionado en la parte inferior de la gráfica.

```
$retriever = new ListaUsuariosDataRetriever($listaUsuarios);
// Datos medios por simulación
$dataGroup = round($retriever->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehi)
->calculateData(new CalculateAverageData($tipoDatoCalcular)), precision: 2);
```

Figura 64: Código modificado en la función generate_form_data_user_analysis() dentro del fichero rjsimulador.data_analysis.forms.inc para obtener los datos para el grupo creado.

Como en todos los grupos anteriores, se utiliza la función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) del

fichero `ListaUsuariosDataRetriever.php`. Esta función se utiliza para recuperar los datos para la gráfica de un grupo concreto de usuarios, como se puede ver en la figura 64. El código de esta función se puede ver en la figura 45. En la figura 65 se aprecia que los datos que se han recuperado para el grupo creado, se utilizan para pintar el gráfico para este grupo.

```
$form['data_user_container']['data_user_chart']['data_user_' . $index] =  
    array(  
        '#type' => 'chart_data',  
        '#title' => t($tituloSustitucion, $arraySustitucion),  
        '#data' => array($dataGroup),  
    );
```

Figura 65: Código modificado en la función `generate_form_data_user_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc` para representar los datos en el gráfico del grupo creado.

Todas estas modificaciones han conseguido un funcionamiento correcto de esta página de la plataforma que se utiliza para comparar un usuario concreto con varios grupos de usuarios.

4.4 Modificaciones del apartado Analysis of traffic offenses

Otro de los apartados que solo es accesible para el administrador de la plataforma web es el que antes se denominaba “Infractions analysis” y tras la actualización, se denomina “Analysis of traffic offenses”. Este apartado de la plataforma ha sufrido numerosas modificaciones en esta actualización.

Este apartado se divide en tres subpáginas que se han modificado también. En estas páginas se analizarán las modificaciones por separado. La primera modificación que se ha llevado a cabo en las tres páginas es el título de las mismas, estos títulos se han modificado de la siguiente manera: “General infraction analysis” por “Analysis of traffic offenses”, “Infraction analysis by group” por “Analysis of traffic offenses per group” y “Infraction analysis by user” por “Analysis of traffic offenses per user”.

A continuación, se comenzarán a analizar las modificaciones de estas páginas por separado empezando por la principal.

4.4.1 Modificaciones de la página Analysis of traffic offenses

Esta página aparece cuando se pulsa en el menú desplegable de la página principal “Analysis of simulations” y después en el apartado “Analysis of traffic offenses”. El título de esta página se ha modificado ya que antes de la actualización se denominaba “General infraction analysis”, como se aprecia en la figura 66.

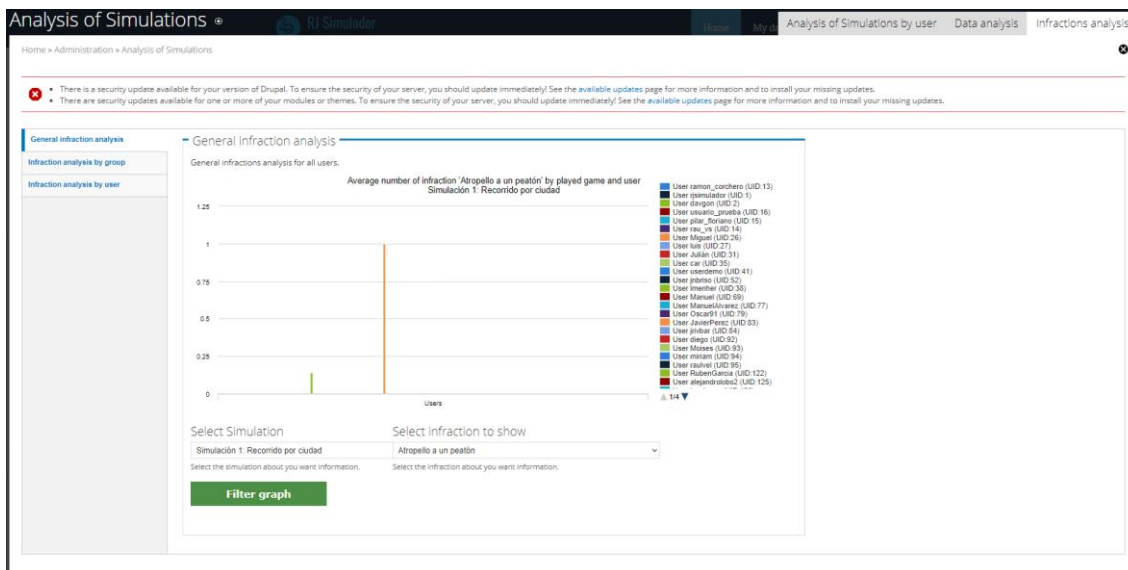


Figura 66: Página “Infractions analysis” de la plataforma web antigua, apartado “General infraction analysis”.

En esta página se ha modificado el título de la gráfica, de lo que se hablará más adelante. También, se han modificado los selectores de datos que se sitúan en la parte inferior de la gráfica y que permiten filtrar los datos que muestran en la gráfica. Además, se ha añadido un nuevo selector para elegir el tipo de vehículo.

El primer selector permite filtrar por el tipo de escenario en el que se ha realizado la simulación. Se ha modificado el título “Select Simulation” por “Select escenario”. También, como en otros apartados el contenido del mismo se ve afectado por las modificaciones realizadas en la base de datos de las que se habló en apartados anteriores, estas modificaciones se pueden ver en la figura 29.

Por otro lado, la descripción de este selector también se ha modificado. Antes la descripción era “Select the simulation about you want information” y ahora es “Select the escenario”, como se puede observar en la figura 67.

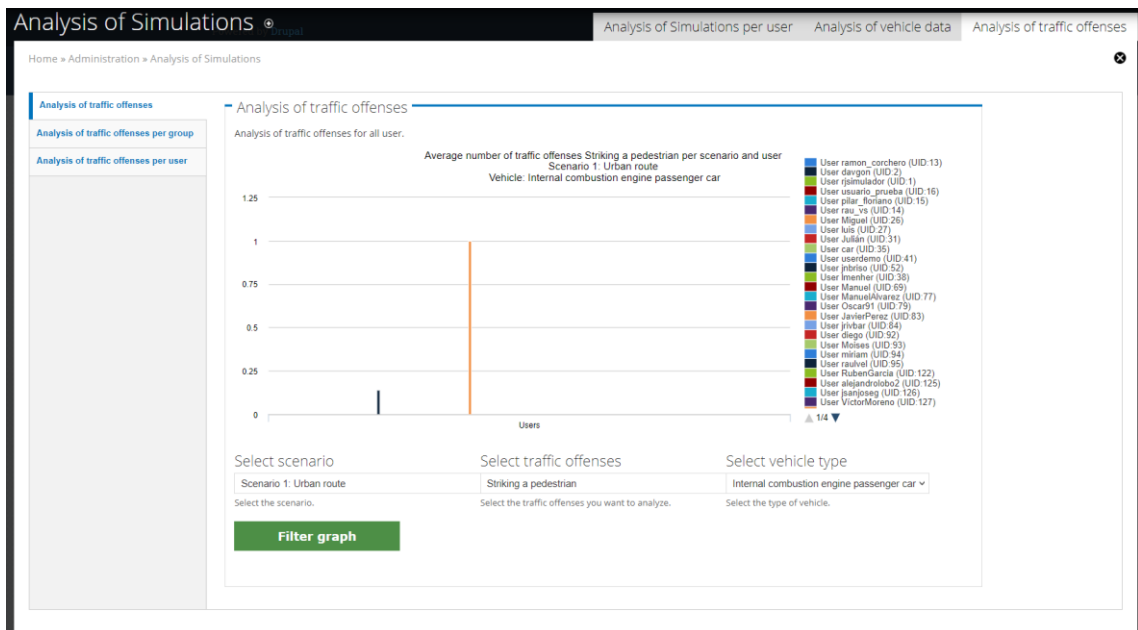


Figura 67: Página “Analysis of traffic offenses” de la plataforma web, apartado “Analysis of traffic offenses”.

El selector que permite elegir el tipo de infracción cometida también ha sufrido modificaciones, que se han realizado en el título del selector, que antes de la actualización se llamaba “Select infraction to show” como se puede apreciar en la figura 66 y que actualmente ha pasado a ser “Select traffic offenses” como se observa en la figura 67. También, se ha modificado la descripción de este selector, que ha pasado a ser “Select the traffic offenses you want to analyze” como se ve en la figura 67, cuando antes era “Select the infraction about you want information” como se aprecia en la figura 66. La última modificación relacionada con este selector es el propio contenido del mismo, que se ha pasado a inglés y se ha adecuado más su contenido a los casos que pueden darse. Los datos que contiene este selector se extraen directamente desde la base de datos, como los datos del selector de escenarios. Para que las modificaciones sean permanentes cuando se instale el módulo en cualquier entorno por primera vez, ha sido necesario modificar el fichero `rjsimulador.install`. Las modificaciones se han realizado en el `hook rjsimulador_install()` que es el encargado de cargar los datos en la base de datos cuando se instala el módulo por primera vez. Estas modificaciones se muestran en la figura 68 y son muy similares a las que se han realizado para el selector de escenarios.

```

function rjsimulador_install() {
  db_insert(table: 'rjsim_infracciones')
  ->fields(array('nombre_infraccion'))
  ->values(array(1, 'Striking a pedestrian'))
  ->values(array(2, 'Collision with other vehicle'))
  ->values(array(3, 'Drive off the road'))
  ->values(array(4, 'Exceed the speed limit'))
  ->values(array(5, 'Wrong exit in a roundabout'))
  ->values(array(6, 'Wrong path in a roundabout'))
  ->values(array(7, 'Opposite direction in the roundabout'))
  ->values(array(8, 'Wrong exit in a crossroads'))
  ->values(array(9, 'Failure to stop at a stop sign'))
  ->values(array(10, 'Failure to yield at a yield sign'))
  ->values(array(11, 'Failing to stop at a red traffic light'))
  ->values(array(12, 'Collision with street furniture'))
  ->values(array(13, 'Cross a solid line'))
  ->values(array(14, 'Low beams not switched driving under a tunnel'))
  ->values(array(15, 'Beams not switched at night'))
  ->values(array(16, 'Wrong turn in a crossroads'))
  ->values(array(17, 'Be closer than the minimum distance to the vehicle ahead'))
  ->values(array(18, 'Not switch on the corresponding turn light in a turn'))
  ->values(array(19, 'Not switch on the corresponding turn light in an overtaking'))
  ->values(array(20, 'Drive under the minimum speed'))
  ->values(array(21, 'Bus lane invasion'))
  ->values(array(22, 'Stop in yellow criss-cross lines'))
  ->values(array(23, 'Run over a cyclist'))
  ->values(array(24, 'Overtaking a cyclist without leaving the minimum safe distance'))
  ->values(array(25, 'Sudden breaking (not using engine braking)'))
  ->values(array(26, 'Not progressive acceleration (from 0 to 50km/h in less than 5 seconds)'))
  ->values(array(27, 'Sudden acceleration'))
  ->execute();

  db_insert(table: 'rjsim_simulacion')
  ->fields(array('nombre_simulacion'))
  ->values(array(1, 'Scenario 1: Urban route'))
  ->values(array(2, 'Scenario 2: Urban route outside'))
  ->values(array(3, 'Scenario 3: Urban route from inside to outside'))
  ->values(array(4, 'Scenario 4: Urban route from outside to inside'))
  ->values(array(5, 'Scenario 5: Unguided Urban route'))
  ->values(array(6, 'Scenario 6: M-30 ring road route'))
  ->values(array(7, 'Scenario 7: Through road route'))
  ->values(array(8, 'Scenario 8: Highway route'))
  ->values(array(9, 'Scenario 9: Urban route in Valladolid (Pilarica - San Juan)'))
  ->values(array(10, 'Scenario 10: Urban road, highway and through road in Valladolid'))
  ->values(array(11, 'Scenario 11: Urban route, highway, interurban road and through road in Valladolid'))
  ->execute();
}

```

Figura 68: Modificaciones realizadas en el hook rjsimulador_install() del fichero rjsimulador.install.

En esta actualización se ha añadido un nuevo selector en esta página, como en otras, que permite filtrar por el tipo vehículo con el que se han realizado las simulaciones, como se puede ver en la figura 67. El título de este selector es “Select vehicle” type y la descripción que se ha incorporado a este selector es “Select the type of vehicle”. Para crear este selector en la página se han tenido que incluir las líneas de código que se muestran en la figura 69 en la función generate_form_infractions_general_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc.

```

// Apartado del formulario que hace referencia al select tipo de vehículo
$form['infracciones_general_actions']['data_general_select_vehiculo'] = array(
  '#type' => 'select',
  '#title' => t(string: 'Select vehicle type'),
  '#options' =>
    array(
      t(string: 'Internal combustion engine passenger car'),
      t(string: 'Electric passenger car'),
      t(string: 'Internal combustion engine city bus'),
      t(string: 'Electric city bus'),
      t(string: 'Internal combustion engine truck'),
      t(string: 'Electric truck')
    ),
  '#default_value' => t(string: 'Tourism with combustion engine'),
  '#description' => t(string: 'Select the type of vehicle.'),
);

```

Figura 69: Código incluido en la función generate_form_infracciones_general_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para generar el selector.

En la actualización, la modificación más importante que se ha realizado en esta página se centra en el propio gráfico. Se ha modificado el título de la misma añadiendo las modificaciones de los nombres de los escenarios relacionadas con el selector, el tipo de infracción que se va a analizar y el tipo de vehículo con el que se ha realizado las simulaciones. Así, al analizar la gráfica, quedarán claros los datos que se muestran y que estamos analizando. Estos cambios se han llevado a cabo en la función `generate_form_infractions_general_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`, como se muestra en las figuras 70 y 71. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector, ya que las otras modificaciones que influyen sobre el título de la gráfica relacionadas con las infracciones cometidas y el escenario son textuales.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['data_general_select_vehiculo']) ?
    $form_state['values']['data_general_select_vehiculo'] : 0;
```

Figura 70: Código incluido en la función `generate_form_infractions_general_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para obtener los datos del selector de vehículos.

En función del valor que se recupera en la variable `$vehiculo` se genera el título para la gráfica, gracias al código que se muestra en la figura 71.

```
// Generamos el título relacionado con el tipo de vehículo
$titleGrafico .= "<br>";
switch ($vehiculo) {
    case 0:
        $titleGrafico .= ' Vehicle: Internal combustion engine passenger car';
        $vehi = 1;
        break;
    case 1:
        $titleGrafico .= ' Vehicle: Electric passenger car';
        $vehi = 2;
        break;
    case 2:
        $titleGrafico .= ' Vehicle: Internal combustion engine city bus';
        $vehi = 3;
        break;
    case 3:
        $titleGrafico .= ' Vehicle: Electric city bus';
        $vehi = 4;
        break;
    case 4:
        $titleGrafico .= ' Vehicle: Internal combustion engine truck';
        $vehi = 5;
        break;
    case 5:
        $titleGrafico .= ' Vehicle: Electric truck';
        $vehi = 6;
        break;
}
```

Figura 71: Código incluido en la función `generate_form_infractions_general_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para generar el título en función del selector de vehículos.

Para poder aplicar los cambios del título en la gráfica que muestra los datos, se ha modificado la parte del formulario que hace referencia al título de la gráfica en la función `generate_form_infractions_general_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`, como se muestra en la figura 72.

```

// Generamos el gráfico
$form['infracciones_general_chart'] = array(
    '#type' => 'chart',
    '#chart_type' => 'column',
    '#chart_library' => 'highcharts',
    '#title' => $tituloGrafico,
    '#prefix' => '<div id="div-infracciones-general-chart">',
    '#suffix' => '</div>',
    'xaxis' =>
        array(
            '#type' => 'chart_xaxis',
            '#labels' => $labelAxisX,
        ),
);

```

Figura 72: Código incluido en la función `generate_form_infracciones_general_analysis()` dentro del fichero `rjsimulador.infracciones_analysis.forms.inc` para generar el título del gráfico.

Para que el funcionamiento de este gráfico a la hora de analizar las infracciones continúe funcionando correctamente, se ha tenido que modificar la codificación de la función `generate_form_infracciones_general_analysis()` dentro del `rjsimulador.infracciones_analysis.forms.inc`, como se muestra en la figura 73.

```

// Recuperamos las infracciones para todos los usuarios que tienen partidas en el simulador
foreach ($gestorSimulaciones->getListaTodosUsuarios() as $usuario) {
    // Media de infracciones por partida, del tipo infracción, la simulación pasadas y el tipo vehículo
    $dataUsuario = round($usuario->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);

    $form['infracciones_general_chart']['infracciones_general_user' . $usuario->getUid()] =
        array(
            '#type' => 'chart_data',
            '#title' => t( string: 'User @userName (UID:@userID)', array(
                '@userName' => $usuario->getName(),
                '@userID' => $usuario->getUid()
            )),
            '#data' => array($dataUsuario),
        );
}

```

Figura 73: Código modificado en la función `generate_form_infracciones_general_analysis()` dentro del fichero `rjsimulador.infracciones_analysis.forms.inc` para obtener los datos para el gráfico y para pintar los datos en el gráfico.

Como se puede apreciar en la figura 73, ha sido necesario crear la función `getAverageInfraccionesByPartidaVehiculo()` en el fichero `UsuarioSimulacion.php` para recuperar los datos necesarios para la gráfica en función del vehículo, el escenario y la infracción seleccionada. Esta nueva función se ha creado como se muestra en la figura 74.

```

/**
 * Devuelve la media de infracciones de un cierto tipo cometidas por partida del usuario para la simulación pasada y para un tipo de vehículo.
 * @param int $idInfraccion El ID de la infracción.
 * @param int $idSimulacion El ID de la simulación.
 * @param int $vehiculo El ID de vehículo.
 * @return float|int Media de infracciones por partida.
 */
public function getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion, $vehiculo) {
    // Recuperamos total de partidas de una simulación y un vehículo
    $totalPartidasSimulacion = $this->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion, $vehiculo)->count();
    // Recuperamos total de infracciones de un cierto tipo por simulación y vehículo
    $totalInfraccionesSimulacion = $this->retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo($idInfraccion, $idSimulacion,$vehiculo)->count();

    // Si el total de partidas es 0 devolvemos 0 evitando la división
    $mediaInfraccionesPorPartida = $totalPartidasSimulacion > 0 ? $totalInfraccionesSimulacion/$totalPartidasSimulacion : 0;

    return $mediaInfraccionesPorPartida;
}

```

Figura 74: Nueva función `getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion, $vehiculo)` dentro del fichero `UsuarioSimulacion.php`.

Como se puede apreciar en la figura 74, la función `getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion, $vehiculo)` a su vez depende de otras dos funciones. Una es `retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion, $vehiculo)`, de la cual se habló en apartados anteriores, y de la que se puede ver el código en la figura 36. La otra función de la que depende es `retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo($idInfraccion, $idSimulacion,$vehiculo)`, que también se ha creado en el fichero `UsuarioSimulacion.php`. Su código se puede ver en la figura 75.

```

/**
 * Método que devuelve todas las infracciones de un cierto tipo de todas las partidas del usuario para la simulación pasada y para un tipo vehículo.
 * @param int $idInfraccion El ID de la infracción a recuperar.
 * @param int $idSimulacion ID de la simulación para la que recuperar las infracciones.
 * @param int $vehiculo ID del vehículo.
 * @return ListaInfracciones Lista de infracciones del tipo pasado para la simulación.
 */
public function retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo($idInfraccion, $idSimulacion, $vehiculo) {
    $listaInfracciones = $this->retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion,$vehiculo);
    $arrayIdsInfracciones = array($idInfraccion);
    // Devolvemos la lista de infracciones filtradas por el IdInfraccion
    return $listaInfracciones->filterBy(new FilterByEquality($arrayIdsInfracciones, paramField: FilterByEquality::INFRACCION_ID));
}

```

Figura 75: Nueva función `retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo($idInfraccion, $idSimulacion, $vehiculo)` dentro del fichero `UsuarioSimulacion.php`.

En la figura 75 se puede ver que la función `retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo ($idInfraccion, $idSimulacion, $vehiculo)` depende de la nueva función `retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion, $vehiculo)`. El código de esta función se puede ver en la figura 76.

```

/**
 * Método que devuelve todas las infracciones de todas las partidas del usuario para una simulación y un tipo de vehículo en concreto.
 * @param int $idSimulacion ID de la Simulación para la que recuperar las infracciones.
 * @param int $vehiculo ID del vehículo.
 * @return ListaInfracciones Lista de infracciones para la simulación pasada.
 */
public function retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion, $vehiculo) {
    $listaInfracciones = new ListaInfracciones();
    // Recuperamos todas las infracciones de una simulación en concreto
    foreach ($this->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehiculo) as $partida) {
        $listaInfracciones->mergeList($partida->getListaInfracciones());
    }
    return $listaInfracciones;
}

```

Figura 76: Nueva función `retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion, $vehiculo)` dentro del fichero `UsuarioSimulacion.php`.

En la figura 76 se observa que la función `retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion, $vehiculo)` depende de la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulation, $vehiculo)`, de la cual se habló en apartados anteriores. Se puede ver su código en la figura 36. Con todas estas modificaciones, se ha conseguido un funcionamiento óptimo de esta página.

4.4.2 Modificaciones de la página Analysis of traffic offenses per group

La segunda de estas páginas de la cual se van a analizar las modificaciones es la que se titulaba “Infraction analysis by group” antes de la actualización como se puede ver en la figura 77 y que, tras la actualización, se denomina “Analysis of traffic offenses per group” como se observa en la figura 78.

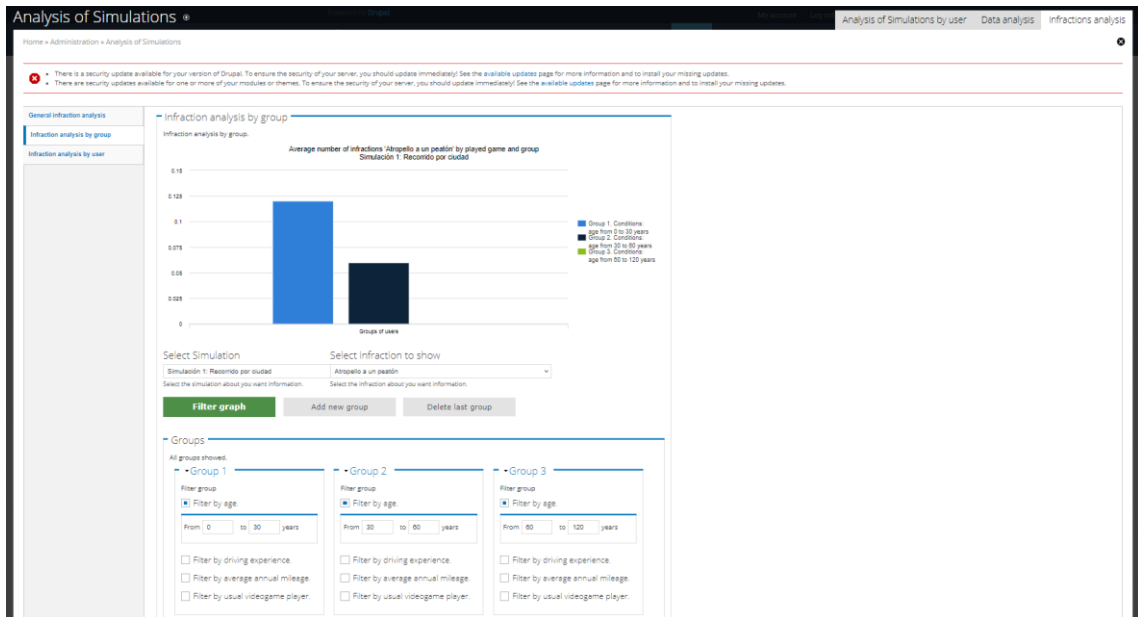


Figura 77: Página “Infractions analysis” de la plataforma web antigua, apartado “Infraction analysis by group”.

Como se ha comentado sobre las modificaciones de la página anterior, se han modificado los selectores de datos que se sitúan en la parte inferior de la gráfica y que permiten filtrar los datos que va a mostrar la gráfica, además, se ha añadido un nuevo selector para elegir el tipo de vehículo.

El primero de los selectores facilita al usuario filtrar por el tipo de escenario en el que se ha realizado la simulación. En este selector de escenarios se ha modificado el título “Select Simulation” por “Select escenario”. También, se ha modificado el contenido del mismo poniendo todas las opciones en inglés y cambiando en cada una de ellas “simulación” por “escenario”. Como se comentó en el apartado anterior, el contenido del selector de escenarios se extrae directamente desde la base de datos. Por eso las modificaciones se han

tenido que realizar directamente en ella y, para que estas sean permanentes cuando se instale el módulo en cualquier entorno por primera vez, se ha tenido que modificar el fichero `rjsimulador.install`. Las modificaciones realizadas en el `hook rjsimulador_install()` que es el encargado de cargar los datos en la base de datos cuando se instala el módulo por primera vez se muestran en la figura 29 del apartado anterior.

La descripción de este selector también se ha modificado como se comentó en el apartado anterior y como se puede ver en la figura 78.

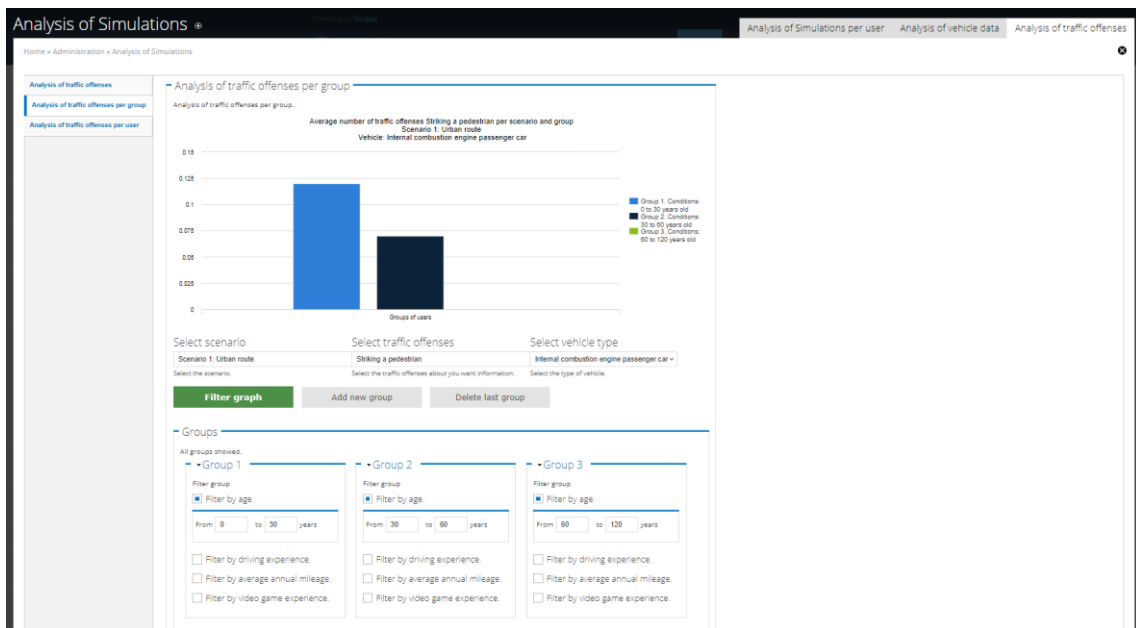


Figura 78: Página “Analysis of traffic offenses” de la plataforma web, apartado “Analysis of traffic offenses per group”.

También se ha modificado el título del selector que antes de la actualización, como se puede ver en la figura 77, se llamaba “Select infraction to show” y como se observa en la figura 78, actualmente ha pasado a ser “Select traffic offenses”. También, se ha modificado la descripción de este selector, que ha pasado a ser “Select the traffic offenses you want to analyze”, como se observa en la figura 78, cuando antes era “Select the infraction about you want information”, como se puede apreciar en la figura 77. El propio contenido de este selector también se ha modificado, se ha pasado a inglés y se ha adecuado más su contenido a los casos que pueden darse. Los datos que contiene este selector se extraen directamente desde la base de datos como los datos del selector de escenarios. Para que las modificaciones sean permanentes cuando se instale el módulo en cualquier entorno por primera vez, ha sido necesario modificar el fichero `rjsimulador.install` como se ha comentado anteriormente. Las modificaciones realizadas en el `hook rjsimulador_install()`, que es el encargado de cargar los datos en la base de datos cuando se instala el módulo por primera vez, se muestran en la figura 68.

En esta página también se ha añadido un nuevo selector que permite filtrar por el tipo vehículo con el que se han realizado las simulaciones, como se

puede observar en la figura 78. El título de este selector es “Select vehicle type” y la descripción que se ha incorporado a este selector es “Select the type of vehicle”. Para generar el selector de la página, se han tenido que incluir las líneas de código que se muestran en la figura 79 que son necesarias para implementar un selector con sus distintas opciones, en la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`.

```
// Apartado del formulario que hace referencia al select tipo de vehículo
$form['infractions_group_container']['infractions_group_actions']['data_group_select_vehiculo'] = array(
  '#type' => 'select',
  '#title' => t( string: 'Select vehicle type'),
  '#options' =>
  array(
    t( string: 'Internal combustion engine passenger car'),
    t( string: 'Electric passenger car'),
    t( string: 'Internal combustion engine city bus'),
    t( string: 'Electric city bus'),
    t( string: 'Internal combustion engine truck'),
    t( string: 'Electric truck')
  ),
  '#default_value' => t( string: 'Tourism with combustion engine'),
  '#description' => t( string: 'Select the type of vehicle.'),
);
```

Figura 79: Código incluido en la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para generar el selector.

Como se ha comentado en otros apartados referentes a otras páginas, se han realizado modificaciones en el título y el contenido de la gráfica de esta página. En el título de la misma se han añadido las modificaciones de los nombres de los escenarios relacionadas con el selector, el tipo de infracción que se va a analizar y el tipo de vehículo con el que se han realizado las simulaciones. Así, al analizar la gráfica quedarán claros los datos que se muestran y que estamos analizando. Estos cambios se han llevado a cabo en la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`, como se muestra en las figuras 80 y 81. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector, ya que las otras modificaciones que influyen sobre el título de la gráfica relacionadas con las infracciones cometidas y el escenario son textuales.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['data_group_select_vehiculo']) ?
  $form_state['values']['data_group_select_vehiculo'] : 0;
```

Figura 80: Código incluido en la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para obtener los datos del selector de vehículos.

En función del valor que se recupera en la variable `$vehiculo`, se genera el título para la gráfica, gracias al código que se muestra en la figura 81.

```

// Generamos el título relacionado con el tipo de vehículo
$tituloGrafico .="<br>";
switch ($vehiculo) {
  case 0:
    $tituloGrafico .= ' Vehicle: Internal combustion engine passenger car';
    $vehi = 1;
    break;
  case 1:
    $tituloGrafico .= ' Vehicle: Electric passenger car';
    $vehi = 2;
    break;
  case 2:
    $tituloGrafico .= ' Vehicle: Internal combustion engine city bus';
    $vehi = 3;
    break;
  case 3:
    $tituloGrafico .= ' Vehicle: Electric city bus';
    $vehi = 4;
    break;
  case 4:
    $tituloGrafico .= ' Vehicle: Internal combustion engine truck';
    $vehi = 5;
    break;
  case 5:
    $tituloGrafico .= ' Vehicle: Electric truck';
    $vehi = 6;
    break;
}

```

Figura 81: Código incluido en la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para generar el título en función del selector.

Para que estos cambios se apliquen al título del gráfico que muestra los datos, se ha modificado la parte del formulario que hace referencia al mismo en la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`, como se muestra en la figura 82. El procedimiento es similar a apartados anteriores.

```

$form['infractions_group_container']['infractions_group_chart'] = array(
  '#type' => 'chart',
  '#chart_type' => 'column',
  '#chart_library' => 'highcharts',
  '#title' => $tituloGrafico,
  '#weight' => 0,
  '#axis' =>
  array(
    '#type' => 'chart_xaxis',
    '#labels' => $labelAxisX,
  ),
);

```

Figura 82: Código incluido en la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para generar el título de la gráfica.

Para que, a la hora de analizar los datos, el comportamiento sea correcto, se ha tenido que modificar la codificación de la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`, como se muestra en la figura 83 y 84.

```
// Recuperamos los datos de la lista de usuarios filtrada
$retriever = new ListaUsuariosDataRetriever($listaUsuarios);

// Media del número de infracciones cometidas por simulación del grupo filtrado
$dataGroup = round($retriever->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);
```

Figura 83: Código modificado en la función `generate_form_infractions_group_analysis()` dentro del fichero `rjsimulador.infracciones_analysis.forms.inc` para obtener los datos para el gráfico.

```
$form['infracciones_group_container']['infracciones_group_chart']['infracciones_group_' . $index] =
  array(
    '#type' => 'chart_data',
    '#title' => t($tituloSustitucion, $arraySustitucion),
    '#data' => array($dataGroup),
  );
```

Figura 84: Código modificado en la función `generate_form_infracciones_group_analysis()` dentro del fichero `rjsimulador.infracciones_analysis.forms.inc` para pintar los datos para el gráfico.

Como se puede apreciar en la figura 83, ha sido necesario crear la función `getAverageInfraccionesByPartidaVehiculo($idInfraccion,$idSimulation,$vehiculo)` en el fichero `ListaUsuariosDataRetriever.php` para recuperar los datos necesarios para la gráfica en función del vehículo, el escenario y la infracción seleccionada. Esta nueva función se ha creado como se muestra en la figura 85.

```
/**
 * Método que devuelve la media de infracciones de un cierto tipo cometidas por partida de la lista de usuarios para la simulación pasada para un tipo de vehículo.
 * @param int $idInfraccion El ID de la infracción.
 * @param int $idSimulation El ID de la simulación.
 * @param int $vehiculo El ID del vehículo.
 * @return float|int Media de infracciones por partida del grupo de usuarios.
 */
public function getAverageInfraccionesByPartidaVehiculo($idInfraccion,$idSimulation,$vehiculo) {
  // Recuperamos total de partidas de una simulación y de un vehículo
  $totalPartidasSimulacion = $this->retrieveAllPartidasByIdSimulationVehiculo($idSimulation,$vehiculo)
  ->count();
  // Recuperamos total de infracciones de un cierto tipo por simulación y cierto tipo de vehículo
  $totalInfraccionesSimulacion = $this->retrieveAllInfraccionesByTypeAndIdSimulationVehiculo($idInfraccion, $idSimulation, $vehiculo)
  ->count();

  // Si el total de partidas es 0 devolvemos 0 evitando la división
  $mediaInfraccionesPorPartida = $totalInfraccionesSimulacion > 0 ? $totalInfraccionesSimulacion / $totalPartidasSimulacion : 0;

  return $mediaInfraccionesPorPartida;
}
```

Figura 85: Nueva función `getAverageInfraccionesByPartidaVehiculo($idInfraccion,$idSimulation,$vehiculo)` dentro del fichero `ListaUsuariosDataRetriever.php`.

En la figura 85, se puede ver que la función `getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulation, $vehiculo)` a su vez depende de otras dos funciones. Una es `retrieveAllPartidasByIdSimulationVehiculo($idSimulation, $vehiculo)` de la cual se habló en apartados anteriores y de la que se puede ver el código en la figura 45. La otra función de la que depende es `retrieveAllInfraccionesByTypeAndIdSimulationVehiculo($idInfraccion, $idSimulation,$vehiculo)`, que también se ha creado en el fichero `ListaUsuariosDataRetriever.php`. Su código se puede ver en la figura 86.

```

/**
 * Método que devuelve todas las infracciones de un cierto tipo de la lista de usuarios para las simulación pasada y para un tipo de vehículo.
 * @param int $idInfraccion El ID de la infracción a recuperar.
 * @param int $idSimulacion ID de la simulación para la que recuperar las infracciones.
 * @param int $vehiculo ID del vehículo.
 * @return ListaInfracciones Lista de infracciones del tipo pasado para la simulación.
 */
public function retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo($idInfraccion, $idSimulacion,$vehiculo) {
    $listaInfracciones = $this->retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion,$vehiculo);
    $arrayIdsInfracciones = array($idInfraccion);
    // Devolvemos la lista de infracciones filtradas por el IdInfraccion
    return $listaInfracciones->filterBy(new FilterByEquality($arrayIdsInfracciones, paramField: FilterByEquality::INFRACCION_ID));
}

```

Figura 86: Nueva función retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo(\$idInfraccion, \$idSimulacion,\$vehiculo) dentro del fichero ListaUsuariosDataRetriever.php.

En la figura 86 se puede ver que la función `retrieveAllInfraccionesByTypeAndIdSimulacionVehiculo($idInfraccion, $idSimulacion, $vehiculo)` depende de la nueva función `retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion, $vehiculo)`. El código de esta función se puede ver en la figura 87. Esta función depende de la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion, $vehiculo)`, de la cual se habló en apartados anteriores y de la que se puede ver el código en la figura 45.

```

/**
 * Método que devuelve todas las infracciones de todas las partidas de la lista de usuarios para una simulación en concreto y para un tipo de vehículo.
 * @param int $idSimulacion ID de la Simulación para la que recuperar las infracciones.
 * @param int $vehiculo ID del vehículo.
 * @return ListaInfracciones Lista de infracciones para la simulación pasada.
 */
public function retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion,$vehiculo) {
    $listaInfracciones = new ListaInfracciones();
    // Recuperamos todas las infracciones de una simulación en concreto
    foreach ($this->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehiculo) as $partida) {
        $listaInfracciones->mergeList($partida->getListaInfracciones());
    }
    return $listaInfracciones;
}

```

Figura 87: Nueva función retrieveAllInfraccionesByIdSimulacionVehiculo(\$idSimulacion, \$vehiculo) dentro del fichero ListaUsuariosDataRetriever.php.

En la figura 87 se observa que la función `retrieveAllInfraccionesByIdSimulacionVehiculo($idSimulacion, $vehiculo)` depende de la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion, $vehiculo)` de la cual se habló en apartados anteriores y de la que se puede ver el código en la figura 45. Estos cambios en el código de la página son muy útiles ya que, cuando se crean los grupos en la parte inferior de esta página para comparar los datos entre ellos, permite que funcione correctamente. En esta parte se ha modificado el título de la casilla “Filter by usual videogame player” por “Filter by video game experience”, para que se ajuste más a la funcionalidad que aporta a la hora de crear los grupos de usuarios para los que se quieren analizar los datos.

4.4.3 Modificaciones de la página Analysis of traffic offenses per user

La siguiente de estas páginas de que se van a analizar las modificaciones es la que se titulaba “Infraction analysis by user” antes de la actualización, como se puede ver en la figura 88, y que, tras la actualización, se denomina “Analysis of traffic offenses per user” como se aprecia en la figura 89.

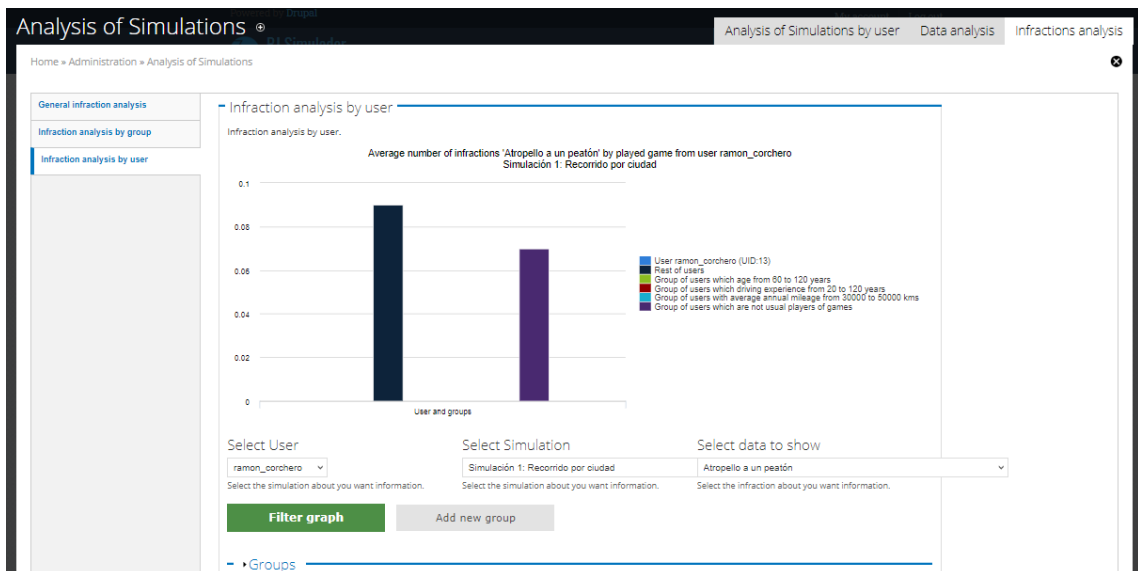


Figura 88: Página “Infractions analysis” de la plataforma web antigua, apartado “Infraction analysis by user”.

Al igual que en las modificaciones de las páginas anteriores, se han modificado los selectores de datos que se sitúan en la parte inferior del gráfico y que permiten filtrar los datos que se van a mostrar en ella. Además, se ha añadido un nuevo selector para elegir el tipo de vehículo.

En el selector que permite filtrar por el tipo de escenario en el que se han realizado las simulaciones, se ha modificado el título “Select Simulation” por “Select escenario”. También, se ha modificado el contenido del mismo poniendo todas las opciones en inglés y cambiando en cada una de ellas “simulación” por “escenario” que es la forma adecuada de referirse a estos datos. Al igual que en los apartados anteriores, el contenido del selector de escenarios se extrae directamente desde la base de datos, por eso las modificaciones se han tenido que realizar directamente en ella y para que estas sean permanentes cuando se instale el módulo en cualquier entorno por primera vez, se ha tenido que modificar el fichero `rjsimulador.install`. Las modificaciones realizadas en el `hook rjsimulador_install()`, que es el encargado de cargar los datos en la base de datos cuando se instala el módulo por primera vez, se muestran en la figura 29 de apartados anteriores e influyen a toda la plataforma. La descripción de este selector también se ha modificado, como se puede ver en la figura 89.

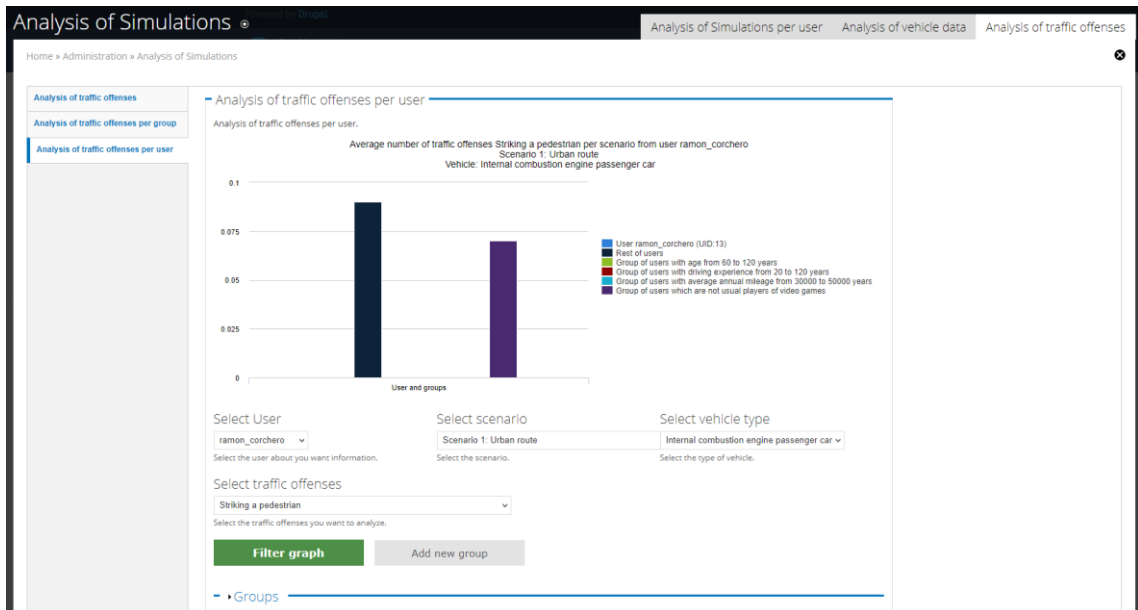


Figura 89: Página “Analysis of traffic offenses” de la plataforma web, apartado “Analysis of traffic offenses per user”.

El selector que antes de la actualización se llamaba “Select infraction to show”, como se muestra en la figura 88, después de la modificación ha pasado a ser “Select traffic offenses”, como se puede ver en la figura 89. Por otro lado, también se ha modificado la descripción de este selector, que ha pasado a ser “Select the traffic offenses you want to analyze”, cuando antes era “Select the infraction about you want information”. Este cambio se puede ver si se comparan las figuras 88 y 89. La última modificación relacionada con este selector es el propio contenido del mismo, que se ha pasado a inglés y se adecua más su contenido a los casos que pueden darse. El contenido del selector se extrae directamente desde la base de datos como los datos del selector de escenarios. Para que las modificaciones sean permanentes cuando se instale el módulo en cualquier entorno por primera vez, ha sido necesario modificar el fichero `rjsimulador.install`. Las modificaciones que se han realizado en el *hook* `rjsimulador_install()`, que es el encargado de cargar los datos en la base de datos cuando se instala el módulo por primera vez, como se comentó en apartados anteriores, se muestran en la figura 68.

Con la actualización de la plataforma, se ha añadido un nuevo selector que permite filtrar por el tipo vehículo con el que se han realizado las simulaciones. El título de este selector es “Select vehicle type” y la descripción que se ha incorporado a este selector es “Select the type of vehicle”. Estos cambios se pueden ver en la figura 89.

Al crear este selector en la página, se han tenido que incluir las líneas de código que se muestran en la figura 90, en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`.

```
// Apartado del formulario que hace referencia al select tipo de vehiculo
$form['infractions_user_container']['infractions_user_actions']['data_user_select_vehiculo'] = array(
  '#type' => 'select',
  '#title' => t( string: 'Select vehicle type'),
  '#options' =>
  array(
    t( string: 'Internal combustion engine passenger car'),
    t( string: 'Electric passenger car'),
    t( string: 'Internal combustion engine city bus'),
    t( string: 'Electric city bus'),
    t( string: 'Internal combustion engine truck'),
    t( string: 'Electric truck')
  ),
  '#default_value' => t( string: 'Tourism with combustion engine'),
  '#description' => t( string: 'Select the type of vehicle.'),
);
```

Figura 90: Código incluido en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para generar el selector.

Al igual que en los anteriores apartados se modificó el gráfico de la página. Se ha modificado el título añadiendo los cambios de los nombres de los escenarios realizadas en el selector, el tipo de infracción que se va a analizar y el tipo de vehículo con el que se ha realizado las simulaciones. Así, cuando se analice el gráfico quedarán claros los datos que se muestran en él. Estos cambios se han llevado a cabo en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`, como se muestra en las figuras 91 y 92. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector, ya que las otras modificaciones que influyen sobre el título de la gráfica relacionadas con los datos a calcular y el escenario son textuales.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['data_user_select_vehiculo']) ?
  $form_state['values']['data_user_select_vehiculo'] : 0;
```

Figura 91: Código incluido en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para obtener los datos del selector de vehículos.

En función del valor que se recupera en la variable `$vehiculo`, se genera el título para la gráfica, gracias al código que se muestra en la figura 92.

```
// Generamos el título relacionado con el tipo de vehículo
$tituloGrafico .= "<br>";
switch ($vehiculo) {
  case 0:
    $tituloGrafico .= ' Vehicle: Internal combustion engine passenger car';
    $vehi = 1;
    break;
  case 1:
    $tituloGrafico .= ' Vehicle: Electric passenger car';
    $vehi = 2;
    break;
  case 2:
    $tituloGrafico .= ' Vehicle: Internal combustion engine city bus';
    $vehi = 3;
    break;
  case 3:
    $tituloGrafico .= ' Vehicle: Electric city bus';
    $vehi = 4;
    break;
  case 4:
    $tituloGrafico .= ' Vehicle: Internal combustion engine truck';
    $vehi = 5;
    break;
  case 5:
    $tituloGrafico .= ' Vehicle: Electric truck';
    $vehi = 6;
    break;
}
```

Figura 92: Código incluido en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para generar el título en función del selector.

Para poder aplicar los cambios del título en la gráfica que muestra los datos, se ha modificado la parte del formulario que hace referencia al título de la gráfica en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc`, como se muestra en la figura 93. El procedimiento es similar a los apartados anteriores.

```
$form['infractions_user_container']['infractions_user_chart'] = array(
  '#type' => 'chart',
  '#chart_type' => 'column',
  '#chart_library' => 'highcharts',
  '#title' => $tituloGrafico,
  '#weight' => 0,
  'xaxis' =>
    array(
      '#type' => 'chart_xaxis',
      '#labels' => $labelAxisX,
    ),
);
```

Figura 93: Código incluido en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para generar el título de la gráfica.

Para que el funcionamiento con las nuevas características de la página sea correcto a la hora de analizar los datos, se ha tenido que modificar la codificación de la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.data_analysis.forms.inc`, como se muestra en las siguientes figuras. En la figura 94 se recuperan los datos de las simulaciones en función del usuario que se elige en el selector de usuarios, el escenario seleccionado en el selector de escenarios y el tipo de vehículo seleccionado.

```
// Seteamos el usuario actual en el Gestor de Simulaciones
$gestorSimulaciones->setUsuarioActual($usuarioActual);
// Recuperamos los datos del usuario actual
$dataUser = round($usuarioActual->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);
```

Figura 94: Código modificado en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para obtener los datos del usuario seleccionado.

Como se puede ver en la figura 94 se utiliza la función `getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi)` creada en el fichero `UsuarioSimulacion.php` para recuperar los datos necesarios de las infracciones para la gráfica en función del tipo de infracción, del escenario seleccionado y del vehículo seleccionado. De esta nueva función se habló en apartados anteriores y las dependencias que tiene. También se mostró en la figura 74 el código de esta función. Por otro lado, en la figura 95 se puede ver que los datos que se han recuperado en la figura 94 para el usuario seleccionado se utilizan para pintarlos en el gráfico.

```
$form['infractions_user_container']['infractions_user_chart']['infractions_user_actual'] =
array(
  '#type' => 'chart_data',
  '#title' => t(string: 'User @name (UID:@uid)', array('@name' => $usuarioActual->getName(), '@uid' => $usuarioActual->getUid())),
  '#data' => array($dataUser),
);
```

Figura 95: Código modificado en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para pintar los datos del usuario seleccionado.

En la figura 96 se pueden ver las líneas de código modificadas para recuperar los datos de las infracciones del grupo de resto de los usuarios, que son todos menos el usuario que se seleccionó en el selector de usuarios. Como se ha comentado anteriormente, estos datos se recuperan en función del tipo de infracción, el escenario seleccionado y del vehículo seleccionado.

```
// Recuperamos los datos del grupo del resto de usuarios
$listaRestoUsuarios = $gestorSimulaciones->getListaTodosUsuariosExceptoActual();
// Instanciamos el recuperador de datos
$retreiver = new ListaUsuariosDataRetriever($listaRestoUsuarios);
$dataRestoUsuarios = round($retreiver->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);
```

Figura 96: Código modificado en la función generate_form_infractions_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para obtener los datos del resto de los usuarios.

En el código que se muestra en la figura 96 ha sido necesario utilizar la función getAverageInfraccionesByPartidaVehiculo(\$idInfraccion, \$idSimulacion,\$vehi) del fichero ListaUsuariosDataRetriever.php para recuperar los datos necesarios de las infracciones para la gráfica en función del tipo de infracción, del escenario seleccionado y del vehículo seleccionado. De esta nueva función se habló en apartados anteriores y las dependencias que tiene, también se mostró en la figura 85 el código de esta función. Si se analiza la figura 97, se observa que los datos que se han recuperado para el grupo resto de usuarios, se asignan a la etiqueta #data del gráfico y gracias a esto se representan en él.

```
$form['infracciones_user_container']['infracciones_user_chart']['infracciones_user_rest_of_user'] =
array(
    '#type' => 'chart_data',
    '#title' => t('Rest of users'),
    '#data' => array($dataRestoUsuarios),
);
```

Figura 97: Código modificado en la función generate_form_infracciones_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para pintar los datos del resto de usuarios.

La figura 98 muestra las líneas de código que se han modificado para recuperar los datos de las simulaciones del grupo de usuarios que tienen la misma edad que el usuario analizado. Con esto se recuperan los datos en función del tipo de infracción a analizar, el escenario seleccionado y del vehículo con el que se realizó la simulación.

```
// Recuperamos los datos del grupo del resto de usuarios que está en el mismo grupo de edad que el usuario analizado
$listaRestoUsuariosMismoGrupoEdad = $listaRestoUsuarios->filterBy(new FilterByInterval($usuarioActual->getGrupoEdad(),
    paramField: FilterByInterval::AGE));
$retreiver->setListaUsuarios($listaRestoUsuariosMismoGrupoEdad);
$dataRestoUsuarioMismoGrupoEdad = round($retreiver->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);
```

Figura 98: Código modificado en la función generate_form_infracciones_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para obtener los datos de los usuarios con la misma edad que el usuario a analizar.

En la figura 98 se observa que se utiliza la función getAverageInfraccionesByPartidaVehiculo(\$idInfraccion, \$idSimulacion,\$vehi) del fichero ListaUsuariosDataRetriever.php para recuperar los datos de las infracciones necesarios para la gráfica de un grupo

concreto de usuarios. De esta nueva función ya se habló con anterioridad y se puede ver en la figura 85. Observando la figura 99 se aprecia que los datos que se han recuperado para el grupo de usuarios que tienen la misma edad que el usuario a analizar, se utilizan para pintar el gráfico para este grupo.

```
$form['infractions_user_container']['infractions_user_chart']['infractions_user_same_age_group'] =
array(
  '#type' => 'chart_data',
  '#title' => t('Group of users with @user_group', array('@user_group' => $usuarioActual->getGrupoEdad( asText: TRUE))),
  '#data' => array($dataRestoUsuarioMismoGrupoEdad),
);
```

Figura 99: Código modificado en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para pintar los datos de los usuarios con la misma edad que el usuario a analizar.

En la figura 100 se pueden ver las líneas de código modificadas para recuperar los datos de las simulaciones del grupo de usuarios que tienen la misma experiencia de conducción que el usuario analizado. Estos datos se recuperan en función del tipo de infracción, del escenario y del vehículo seleccionado.

```
// Recuperamos los datos del grupo del resto de usuarios que está en el mismo grupo de experiencia conductora que el usuario analizado
$listadoRestoUsuariosMismoGrupoExperiencia = $listaRestoUsuarios->filterBy(new FilterByInterval($usuarioActual->getGrupoExperiencia(), paramField: FilterByInterval::DRIVING_EXPERIENCE));
$retriever->setListaUsuarios($listadoRestoUsuariosMismoGrupoExperiencia);
$dataRestoUsuarioMismoGrupoExperiencia = round($retriever->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);
```

Figura 100: Código modificado en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para obtener los datos de los usuarios con la misma experiencia conduciendo que el usuario a analizar.

En la figura 100 se puede apreciar que, como en otros casos en los que se está trabajando con grupos de usuarios, se utiliza la función `getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi)` del fichero `ListaUsuariosDataRetriever.php` para recuperar los datos de las infracciones necesarios para el gráfico de un grupo concreto de usuarios. Se ha hablado con anterioridad de esta función y se puede ver el código de la misma en la figura 85. Observando la figura 101, se aprecia que los datos que se han recuperado para el grupo de usuarios con la misma experiencia conduciendo que el usuario a analizar, se utilizan para pintar el gráfico para este grupo.

```
$form['infractions_user_container']['infractions_user_chart']['infractions_user_same_driving_experience_group'] =
array(
  '#type' => 'chart_data',
  '#title' => t('Group of users with @user_group', array('@user_group' => $usuarioActual->getGrupoExperiencia( asText: TRUE))),
  '#data' => array($dataRestoUsuarioMismoGrupoExperiencia),
);
```

Figura 101: Código modificado en la función `generate_form_infractions_user_analysis()` dentro del fichero `rjsimulador.infractions_analysis.forms.inc` para pintar los datos de los usuarios con la misma experiencia conduciendo que el usuario a analizar.

En la figura 102 se muestran las nuevas líneas de código utilizadas para recuperar los datos de las infracciones del grupo de usuarios que tienen un kilometraje anual dentro del mismo intervalo que el usuario analizado. Para ello se obtendrá la lista de usuarios que cumple estos criterios y, a partir de estos, se recuperan los datos de las infracciones. Estos datos se recuperan en

función del tipo de infracción, el escenario seleccionado y el tipo de vehículo, como se viene haciendo para otros grupos.

```
// Recuperamos los datos del grupo del resto de usuarios que está en el mismo grupo de kilometraje medio que el usuario analizado
$listaRestoUsuariosMismoGrupoKilometraje = $listaRestoUsuarios->filterBy(new FilterByInterval($usuarioActual->getGrupoKilometrajeMedioAnual(),
    paramField: FilterByInterval::AVERAGE_ANNUAL_MILEAGE));
$retriever->setListaUsuarios($listaRestoUsuariosMismoGrupoKilometraje);
$dataRestoUsuarioMismoGrupoKilometraje = round($retriever->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);
```

Figura 102: Código modificado en la función generate_form_infractions_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para obtener los datos de los usuarios con el mismo número de kilómetros recorridos que el usuario a analizar.

Como para los otros grupos creados en esta página con anterioridad, es necesario utilizar la función getAverageInfraccionesByPartidaVehiculo(\$idInfraccion, \$idSimulacion,\$vehi) del fichero ListaUsuariosDataRetriever.php. Esta función se utiliza para recuperar los datos de las infracciones para el gráfico de un grupo concreto de usuarios como se puede ver en la figura 102. El código de esta función se puede ver en la figura 85, así como las dependencias con otras funciones. En la figura 103 se aprecia que los datos de las infracciones que se han recuperado para el grupo de usuarios con el mismo número de kilómetros recorridos que el usuario a analizar utilizando las líneas de código comentadas anteriormente, se asignan a la etiqueta #data del gráfico para poder ser representados en él los datos de este grupo de usuarios.

```
$form['infracciones_user_container']['infracciones_user_chart']['infracciones_user_same_average_annual_mileage_group'] =
    array(
        '#type' => 'chart_data',
        '#title' => t('Group of users with @user_group', array('@user_group' => $usuarioActual->getGrupoKilometrajeMedioAnual( asText: true))),
        '#data' => array($dataRestoUsuarioMismoGrupoKilometraje),
    );
```

Figura 103: Código modificado en la función generate_form_infracciones_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para pintar los datos de los usuarios con el mismo número de kilómetros recorridos que el usuario a analizar.

La figura 104 muestra el código utilizado para recuperar los datos de las simulaciones del grupo de usuarios que tienen la misma experiencia en videojuegos que el usuario analizado. Los datos se obtienen filtrando las simulaciones por el tipo de infracción, el escenario que se seleccione y el tipo de vehículo empleado en la simulación.

```
// Recuperamos los datos del grupo del resto de usuarios que está en es del mismo tipo de jugador habitual que el usuario analizado
$listaRestoUsuariosMismoTipoJugador = $listaRestoUsuarios->filterBy(new FilterByEquality(array($usuarioActual->isUsualVideogamePlayer()),
    paramField: FilterByEquality::USUARIO_USUAL_PLAYER));
$retriever->setListaUsuarios($listaRestoUsuariosMismoTipoJugador);
$dataRestoUsuarioMismoTipoJugador = round($retriever->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);
```

Figura 104: Código modificado en la función generate_form_infracciones_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para obtener los datos de los usuarios con la misma experiencia en videojuegos que el usuario a analizar.

Para el grupo creado con la misma experiencia en videojuegos que el usuario a analizar es necesario utilizar la función getAverageInfraccionesByPartidaVehiculo(\$idInfraccion, \$idSimulacion,\$vehi) del fichero ListaUsuariosDataRetriever.php, como para

otros grupos. Esta función se utiliza para recuperar los datos para la gráfica de un grupo concreto de usuarios, como se puede ver en la figura 104. El código de esta función se puede ver en la figura 85. En la figura 105 se puede ver que los datos que se han recuperado para el grupo de usuarios con la misma experiencia en videojuegos que el usuario a analizar, se utilizan para pintar el gráfico para este grupo.

```
$form['infracciones_user_container']['infracciones_user_chart']['infracciones_user_same_usual_videogame_player_group'] =
array(
  '#type' => 'chart_data',
  '#title' => t('Group of users which @user_group usual players of video games',
  array('@user_group' => $usuarioActual->isUsualVideogamePlayer( asBoolean: TRUE) ? t(' string: "are") : t(' string: "are not"')),
  '#data' => array($dataRestoUsuarioMismoTipoJugador),
);
```

Figura 105: Código modificado en la función generate_form_infracciones_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para pintar los datos de los usuarios con la misma experiencia en videojuegos que el usuario a analizar.

En la parte inferior de esta página se pueden crear grupos para comparar los datos con los grupos ya existentes y con el usuario a analizar. Se ha modificado el título de la casilla de verificación “Filter by usual videogame player” por “Filter by video game experience”, para que se ajuste más a la funcionalidad que aporta. Cuando el usuario utiliza la parte inferior de la página para crear un grupo de usuarios con ciertas características, se utiliza el código de la figura 106 para recuperar los datos de las infracciones de este grupo. Para ello, se filtran los datos por el tipo de infracción, el escenario donde se realizó la simulación y el tipo de vehículo con el que se realizó.

```
$retriever = new ListaUsuariosDataRetriever($listaUsuarios);
// Datos medios por simulación
$dataGroup = round($retriever->getAverageInfraccionesByPartidaVehiculo($idInfraccion, $idSimulacion,$vehi), precision: 2);
```

Figura 106: Código modificado en la función generate_form_infracciones_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para obtener los datos para el grupo creado.

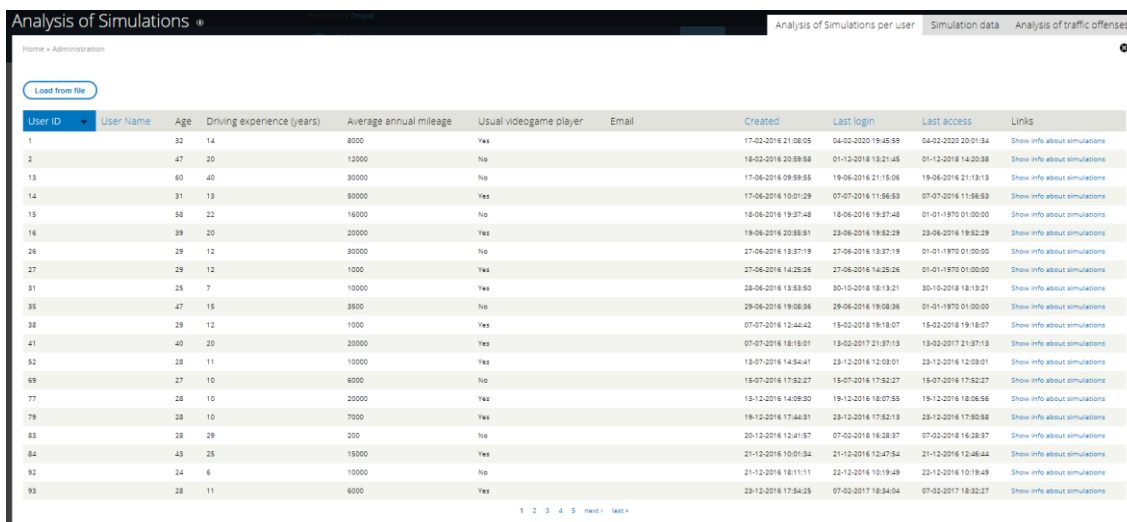
Como se ha venido dando para todos los grupos de esta página, es necesario utilizar la función getAverageInfraccionesByPartidaVehiculo(\$idInfraccion, \$idSimulacion,\$vehi) del fichero ListaUsuariosDataRetriever.php para recuperar los datos de las infracciones de este grupo. Una vez que se han recuperado los datos de las infracciones de este grupo, se pintan en el gráfico como se muestra en la figura 107.

```
$form['infracciones_user_container']['infracciones_user_chart']['infracciones_user_' . $index] =
array(
  '#type' => 'chart_data',
  '#title' => t($tituloSustitucion, $arraySustitucion),
  '#data' => array($dataGroup),
);
```

Figura 107: Código modificado en la función generate_form_infracciones_user_analysis() dentro del fichero rjsimulador.infracciones_analysis.forms.inc para pintar los datos para el grupo creado.

4.5 Apartado My data about the simulations y Analysis of Simulations of Simulations

Ahora se analizarán las páginas que son accesibles para cualquier usuario, incluso para el administrador, en el apartado “My data about the simulations” de la página principal. En estas páginas se pueden analizar los datos de las simulaciones que ha realizado el usuario que accede a este apartado. Por otro lado, el administrador puede acceder a las páginas de los distintos usuarios con los datos de sus simulaciones, como se comentará más adelante, simplemente pulsando en el apartado “Analysis of Simulations” del menú desplegable de la página principal. Esta página se puede ver en la figura 108.



The screenshot shows a web application interface titled "Analysis of Simulations". It features a navigation menu with options like "Analysis of Simulations per user", "Simulation data", and "Analysis of traffic offenses". Below the menu is a "Load from file" button. The main content is a table with the following columns: User ID, User Name, Age, Driving experience (years), Average annual mileage, Usual videogame player, Email, Created, Last login, Last access, and Links. The table contains 20 rows of data, each representing a user's simulation record. The "Links" column for each row contains a "Show info about simulation" link.

User ID	User Name	Age	Driving experience (years)	Average annual mileage	Usual videogame player	Email	Created	Last login	Last access	Links
1		32	14	8000	Yes		17-02-2016 21:08:05	04-02-2020 19:45:59	04-02-2020 20:01:34	Show info about simulation
2		47	20	12000	No		18-02-2016 20:59:58	01-12-2018 13:21:45	01-12-2018 14:20:38	Show info about simulation
13		60	40	30000	No		17-06-2016 09:59:55	19-06-2016 21:19:06	19-06-2016 21:13:13	Show info about simulation
14		31	13	80000	Yes		17-06-2016 10:01:29	07-07-2016 11:56:53	07-07-2016 11:56:53	Show info about simulation
15		58	22	14000	No		18-06-2016 19:37:48	18-06-2016 19:37:48	01-01-1970 01:00:00	Show info about simulation
16		39	20	20000	Yes		19-06-2016 20:59:51	23-06-2016 19:52:29	23-06-2016 19:52:29	Show info about simulation
26		29	12	30000	No		27-06-2016 13:37:19	27-06-2016 13:37:19	01-01-1970 01:00:00	Show info about simulation
27		29	12	1000	Yes		27-06-2016 14:25:26	27-06-2016 14:25:26	01-01-1970 01:00:00	Show info about simulation
31		25	7	10000	Yes		28-06-2016 13:53:50	30-10-2018 18:13:21	30-10-2018 18:13:21	Show info about simulation
35		47	15	3500	No		29-06-2016 19:08:36	29-06-2016 19:08:36	01-01-1970 01:00:00	Show info about simulation
38		29	12	1000	Yes		07-07-2016 12:44:42	15-02-2018 19:18:07	15-02-2018 19:18:07	Show info about simulation
41		40	20	20000	Yes		07-07-2016 18:15:01	13-02-2017 21:37:13	13-02-2017 21:37:13	Show info about simulation
52		28	11	10000	Yes		13-07-2016 14:54:41	23-12-2016 12:09:01	23-12-2016 12:09:01	Show info about simulation
69		27	10	6000	No		19-07-2016 17:52:27	19-07-2016 17:52:27	19-07-2016 17:52:27	Show info about simulation
77		28	10	20000	Yes		13-12-2016 14:09:30	19-12-2016 18:07:55	19-12-2016 18:06:56	Show info about simulation
79		28	10	7000	Yes		19-12-2016 17:44:31	23-12-2016 17:52:13	23-12-2016 17:50:58	Show info about simulation
83		28	29	200	No		20-12-2016 12:41:57	07-02-2018 16:28:37	07-02-2018 16:28:37	Show info about simulation
84		43	25	15000	Yes		21-12-2016 10:01:34	21-12-2016 12:47:54	21-12-2016 12:46:44	Show info about simulation
92		24	6	10000	No		21-12-2016 18:11:11	22-12-2016 10:19:49	22-12-2016 10:19:49	Show info about simulation
93		28	11	6000	Yes		23-12-2016 17:54:25	07-02-2017 18:34:04	07-02-2017 18:32:27	Show info about simulation

Figura 108: Página desde la que el administrador tiene acceso a los datos de las simulaciones de cada usuario.

Dentro de esta página, el administrador tiene la posibilidad de acceder a los datos de las simulaciones de cada usuario, pulsando en el hipertexto “Show info about simulation” de la fila que representa a cada usuario en la tabla.

Una vez explicado cómo se accede a la página “My data about the Simulation” utilizando los distintos modos posibles en función del usuario, procederemos a analizar las modificaciones que ha sufrido este apartado y las subpáginas del mismo con la actualización de la plataforma.

4.5.1 Modificaciones de la página My data about the Simulation

Para poder analizar con detalle la página “My data about the Simulation” la separaremos en tres partes.

4.5.1.1 Modificaciones de la parte superior de la página My data about the Simulation

La primera de ellas abarca las dos primeras gráficas que aparecen en la parte superior de la pantalla. En la figura 109 aparecen estas dos gráficas antes de la actualización y en la figura 110 se pueden observar las dos gráficas después de la actualización.

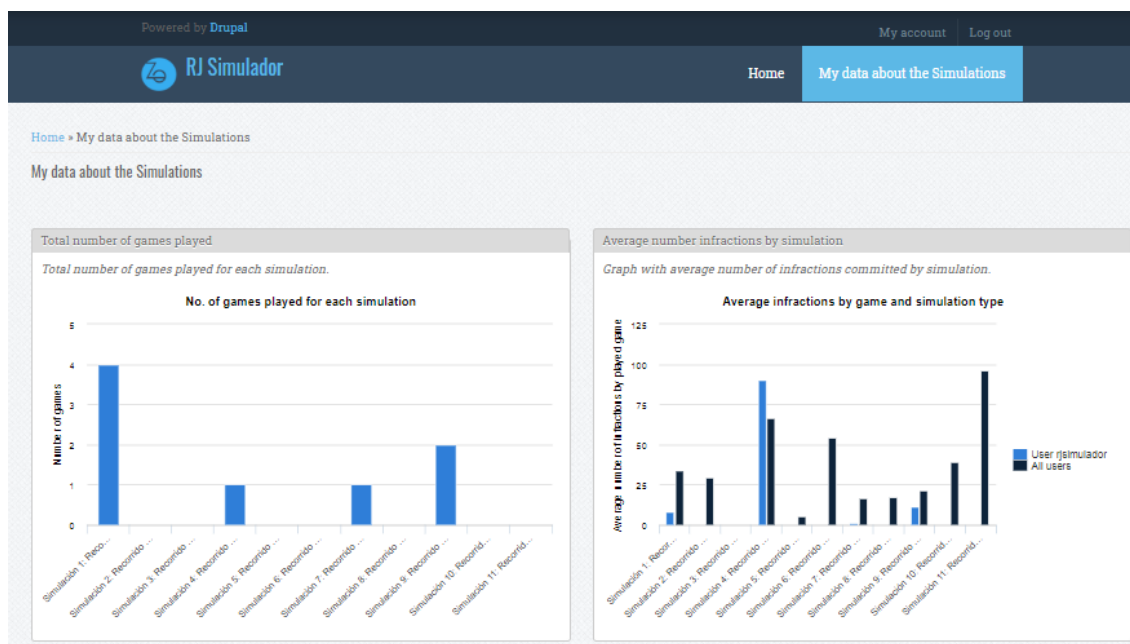


Figura 109: Parte superior de la página “My data about the Simulation” antes de la actualización de la plataforma.

En esta página de la plataforma, se han realizado varias modificaciones. En la parte superior de la misma existen dos gráficas. En estas dos gráficas se han cambiado los títulos principales, de la siguiente manera: “Total number of games played” por “Total number of simulations fulfilled by the user” y “Average number infractions by simulation” por “Average number of traffic offenses per scenario”.

En estas gráficas se ha eliminado la descripción de la misma porque resultaba innecesaria, ya que no aportaba ningún tipo de información adicional. En las gráficas también se ha modificado el título de la propia gráfica para adecuarlo más a los datos que contienen. Esto ayuda a los usuarios a la hora de analizar los datos y saber qué están viendo. El cambio realizado es el siguiente “No. of games played for each simulation” por “Total number of simulations fulfilled by the user” y “Average infractions by game and simulation type” por “Average number of traffic offenses”.

Estos cambios son meramente visuales y hacen la plataforma más fácil de utilizar para los usuarios. Se puede ver cómo ha cambiado la página de la plataforma si se comparan las figuras 109 y 110.

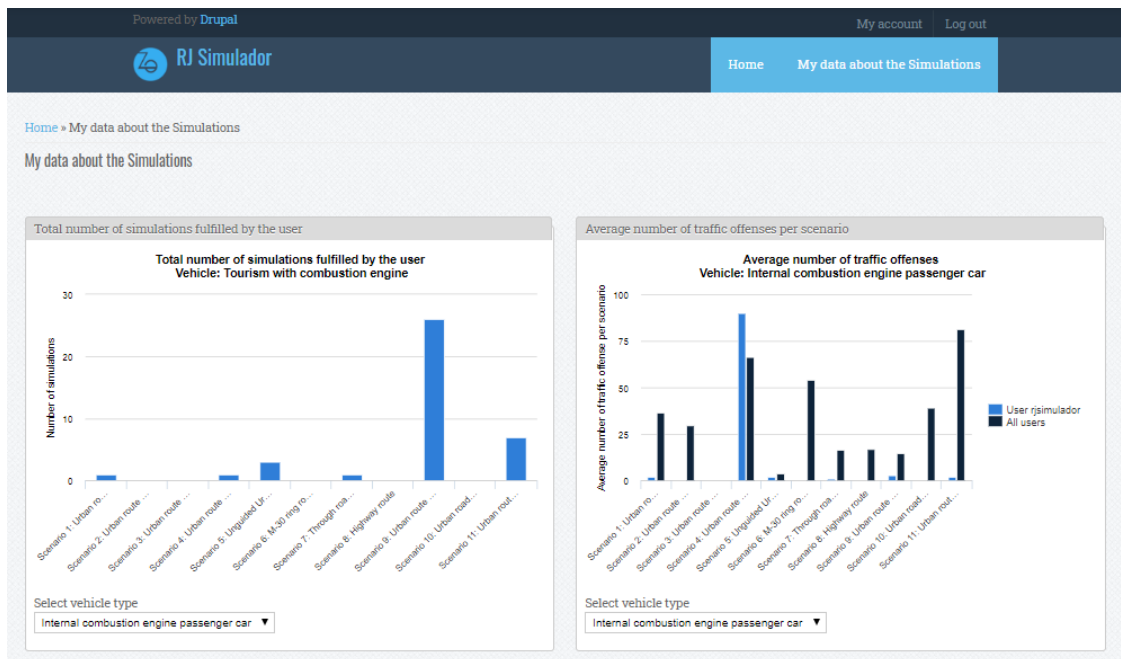


Figura 110: Parte superior de la página “My data about the Simulation” después de la actualización de la plataforma.

En esta parte de la página se ha incluido un selector bajo las gráficas. Este selector permite a los usuarios seleccionar el tipo de vehículo de cada simulación. Esta nueva característica facilita el análisis de los datos de las simulaciones y ayuda a saber a los usuarios con qué tipo de vehículo han realizado cada simulación. Para poder incluir el selector en la gráfica “Total number of simulations fulfilled by the user” se ha tenido que modificar la función `rjsim_simulaciones_upper_left_form()` del fichero `rjsimulador.forms.inc`. En esta función se ha tenido que incluir el código de la figura 111 para implementar el selector.

```
// Apartado del formulario que hace referencia al select tipo de vehículo
$form['rjsim_fieldset_partidas_simulacion']['select_vehiculo'] = array(
  '#type' => 'select',
  '#title' => t( string: 'Select vehicle type'),
  '#options' => array(
    1 => t( string: 'Internal combustion engine passenger car'),
    2 => t( string: 'Electric passenger car'),
    3 => t( string: 'Internal combustion engine city bus'),
    4 => t( string: 'Electric city bus'),
    5 => t( string: 'Internal combustion engine truck'),
    6 => t( string: 'Electric truck')
  ),
  '#default_value' => 1,
  '#ajax' => array(
    'callback' => 'ajax_partidas_vehiculo_callback',
    'wrapper' => 'div-chart-partidas-simulacion',
  ),
);
```

Figura 111: Código incluido en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` para generar el selector de la gráfica “Total number of simulations fulfilled by the user”.

El título de esta gráfica se ha modificado en función del selector añadiendo el tipo de vehículo seleccionado por el usuario. Así, al analizar la gráfica, quedarán claros los datos que se muestran y que estamos analizando. La

elección del selector filtra los datos de las simulaciones por el tipo de vehículo con el que se realizaron. Estos cambios se han llevado a cabo en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` como se muestra en las figuras 112 y 113. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector que influye sobre el título de la gráfica.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehicul = isset($form_state['values']['select_vehiculo']) ? $form_state['values']['select_vehiculo'] : 1;
```

Figura 112: Código incluido en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del selector de vehículos.

En función del valor que se recupera en la variable `$vehicul`, se genera el título para la gráfica, gracias al código que se muestra en la figura 113.

```
// Generamos el título relacionado con el tipo de vehículo
$titleGrafic = "<br>";
switch ($vehicul) {
  case 1:
    $titleGrafic .= ' Vehicle: Tourism with combustion engine';
    break;
  case 2:
    $titleGrafic .= ' Vehicle: Tourism with electric engine';
    break;
  case 3:
    $titleGrafic .= ' Vehicle: Bus with combustion engine';
    break;
  case 4:
    $titleGrafic .= ' Vehicle: Bus with electric engine';
    break;
  case 5:
    $titleGrafic .= ' Vehicle: Truck with combustion engine';
    break;
  case 6:
    $titleGrafic .= ' Vehicle: Truck with electric engine';
    break;
}
```

Figura 113: Código incluido en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` para generar el título en función del selector.

Para poder aplicar los cambios del título en la gráfica, se ha modificado la parte del formulario que hace referencia a la propia gráfica en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en la figura 114.


```

$form['rjsim_fieldset_partidas_simulacion'] = [
    '#type' => 'fieldset',
    '#title' => t( string: 'Total number of simulations fulfilled by the user'),
    '#collapsible' => FALSE,
    '#attributes' => ['class' => ['rjsim-margin']]
];

// Modificamos el apartado del formulario que hace referencia a la gráfica para que interactúe con el select tipo de vehículo
$form['rjsim_fieldset_partidas_simulacion']['chart_partidas_simulacion'] = [
    '#type' => 'chart',
    '#chart_type' => 'column',
    '#chart_library' => 'highcharts',
    '#legend' => FALSE,
    '#title' => t( string: 'Total number of simulations fulfilled by the user'.stituloGrafic),
    '#title_font_weight' => 'bold',
    '#prefix' => '<div id="div-chart-partidas-simulacion">',
    '#suffix' => '</div>',
    'simulation' => [
        '#type' => 'chart_data',
        '#title' => t( string: 'Number of simulations'),
        '#data' => $arrayData,
    ],
    'xaxis' =>
    [
        [
            '#type' => 'chart_xaxis',
            '#labels' => $xLabels
        ],
    ],
    'yaxis' =>
    [
        [
            '#type' => 'chart_yaxis',
            '#title' => 'Number of simulations',
        ],
    ],
];

```

Figura 114: Código de la función rjsim_simulaciones_upper_left_form() dentro del fichero rjsimulador.forms.inc que genera la gráfica.

Para que el funcionamiento a la hora de analizar los datos sea correcto, se ha tenido que modificar la codificación de la función rjsim_simulaciones_upper_left_form() dentro del fichero rjsimulador.forms.inc como se muestra en la figura 115. En ella, se puede ver que se recuperan los datos de todas las simulaciones en función del tipo de vehículo que el usuario elige en el selector de vehículos, recorriendo todas las simulaciones del usuario y filtrándolas por el tipo de vehículo.

```

// Recorremos la lista de simulaciones del usuario actual
foreach ($gestorSimulaciones->getUsuarioActual()->getListaSimulaciones() as $simulacion) {
    // Obtenemos el número de partidas de una simulación en función de un tipo de vehículo
    $arrayData[] = $simulacion->getListaPartidasVehiculo($vehicul)->count();
    $xLabels[] = $simulacion->getNombreSimulacion();
}

```

Figura 115: Código modificado de la función rjsim_simulaciones_upper_left_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del usuario actual.

Al observar la figura 115 se puede apreciar que se utiliza una nueva función getListaPartidasVehiculo(\$vehiculo) creada en el fichero Simulacion.php para recuperar la lista de simulaciones de un escenario en función del tipo de vehículo. Estos datos son necesarios para la gráfica que analiza el número de simulaciones. El código de la función se puede ver en la figura 116. Por otro lado, en la figura 114 se puede ver que los datos del usuario que se han recuperado en la figura 115 en función del tipo de vehículo seleccionado se utilizan para pintarlos en el gráfico.

```

/**
 * Método que devuelve la lista de partidas de una simulación en función del tipo de vehículo
 * @param int $vehiculo El id del vehículo.
 * @return ListaPartidas
 * @throws Exception Error recuperando la lista de partidas.
 */
public function getListaPartidasVehiculo($vehiculo) {
    $listapartidasVehi = new ListaPartidas();
    $provider = FactoryDataManager::createDataProvider();
    $listapartidas = $provider->loadListaPartidasBySimulation($this);
    foreach ($listapartidas as $partida) {
        if ($partida->getTipoVehiculo() == $vehiculo) {
            $listapartidasVehi->add($partida);
        }
    }
    return $listapartidasVehi;
}

```

Figura 116: Nueva función getListaPartidasVehiculo(\$vehiculo) dentro del fichero Simulacion.php.

También ha sido necesario crear una nueva función AJAX que se ejecuta cuando cambia el valor del selector, esta función se puede ver figura 117. La llamada a esta función AJAX se puede ver en la figura 111. La función ajax_partidas_vehiculo_callback() recarga la parte del formulario que hace referencia a la gráfica actualizando los datos en función del vehículo seleccionado.

```

// Función Ajax que actualiza el valor del tipo de vehículo para la gráfica
function ajax_partidas_vehiculo_callback($form, $form_state) {
    return $form['rjsim_fieldset_partidas_simulacion']['chart_partidas_simulacion'];
}

```

Figura 117: Nueva función ajax_partidas_vehiculo_callback() dentro del fichero rjsimulador.forms.inc.

Para incluir el selector de tipo de vehículo en la gráfica “Average number of traffic offenses”, se ha tenido que modificar la función rjsim_simulaciones_upper_right_form() del fichero rjsimulador.forms.inc. En esta función se ha tenido que incluir el código de la figura 118 para implementar el selector.

```

// Apartado del formulario que hace referencia al select tipo de vehículo
$form['rjsim_fieldset_infracciones_simulacion']['select_vehiculo_infracciones_simulacion'] = array(
    '#type' => 'select',
    '#title' => t('Select vehicle type'),
    '#options' => array(
        1 => t('Internal combustion engine passenger car'),
        2 => t('Electric passenger car'),
        3 => t('Internal combustion engine city bus'),
        4 => t('Electric city bus'),
        5 => t('Internal combustion engine truck'),
        6 => t('Electric truck')
    ),
    '#default_value' => 1,
    '#ajax' => array(
        'callback' => 'ajax_infracciones_vehiculo_callback',
        'wrapper' => 'div-chart-infracciones-simulacion',
    ),
);

```

Figura 118: Código incluido en la función rjsim_simulaciones_upper_right_form() dentro del fichero rjsimulador.forms.inc para generar el selector de la gráfica “Average number of traffic offenses”.

En el título de esta gráfica se han realizado modificaciones en función del selector, añadiendo el tipo de vehículo seleccionado por el usuario. Como

en la gráfica que se analizó antes, la elección del selector filtra los datos de las simulaciones por el tipo de vehículo con el que se realizaron. Estos cambios se han llevado a cabo en la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en las figuras 119 y 120. En estas figuras solo se muestran las modificaciones relacionadas con el nuevo selector que influye sobre el título.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['select_vehiculo_infracciones_simulacion']) ? $form_state['values']['select_vehiculo_infracciones_simulacion'] : 1;
```

Figura 119: Código incluido en la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del selector de vehículos.

En función del valor que se recupera en la variable `$vehiculo` se genera el título para la gráfica, gracias al código que se muestra en la figura 120.

```
// Generamos el título relacionado con el tipo de vehículo
$tituloGrfico = "<br>";
switch ($vehiculo) {
  case 1:
    $tituloGrfico .= ' Vehicle: Internal combustion engine passenger car';
    break;
  case 2:
    $tituloGrfico .= ' Vehicle: Electric passenger car';
    break;
  case 3:
    $tituloGrfico .= ' Vehicle: Internal combustion engine city bus';
    break;
  case 4:
    $tituloGrfico .= ' Vehicle: Electric city bus';
    break;
  case 5:
    $tituloGrfico .= ' Vehicle: Internal combustion engine truck';
    break;
  case 6:
    $tituloGrfico .= ' Vehicle: Electric truck';
    break;
}
```

Figura 120: Código incluido en la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc` para generar el título en función del selector.

Para que los cambios surtan efecto en el título en la gráfica, se ha modificado la parte del formulario que hace referencia a la gráfica en la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc`, añadiendo la variable `$tituloGrfico` que contiene los datos del tipo de vehículo que el usuario eligió en el selector al contenido de la etiqueta `#title` de la gráfica, como se muestra en la figura 121.

```

$form['rjsim_fieldset_infracciones_simulacion'] = array(
    '#type' => 'fieldset',
    '#title' => t( string: 'Average number of traffic offenses per scenario'),
    '#collapsible' => FALSE,
    '#attributes' => array('class' => array('rjsim-margin'))
);

// Modificamos el apartado del formulario que hace referencia a la gráfica para que interactúe con el select tipo de vehículo
$form['rjsim_fieldset_infracciones_simulacion']['chart_infracciones_simulacion_1'] = array(
    '#type' => 'chart',
    '#chart_type' => 'column',
    '#chart_library' => 'highcharts',
    '#title' => t( string: 'Average number of traffic offenses'. $tituloGrfico),
    '#title_font_weight' => 'bold',
    '#prefix' => '<div id="div-chart-infracciones-simulacion">',
    '#suffix' => '</div>',
    'infracciones_usuario' => array(
        '#type' => 'chart_data',
        '#title' => t( string: 'User ' ) . $gestorSimulaciones->getUsuarioActual()
            ->getName(),
        '#data' => $dataUsuarioInfracciones,
    ),
    'infracciones_totales' => array(
        '#type' => 'chart_data',
        '#title' => t( string: 'All users'),
        '#data' => $dataTotalInfracciones,
    ),
    'xaxis' =>
    [
        [
            '#type' => 'chart_xaxis',
            '#labels' => $labelXAxis
        ],
    ],
    'yaxis' =>
    [
        [
            '#type' => 'chart_yaxis',
            '#title' => 'Average number of traffic offense per scenario'
        ],
    ],
);

```

Figura 121: Código de la función rjsim_simulaciones_upper_right_form() dentro del fichero rjsimulador.forms.inc que genera la gráfica.

Se ha tenido que modificar la codificación de la función rjsim_simulaciones_upper_right_form() dentro del fichero rjsimulador.forms.inc. Esta función ha sufrido numerosas modificaciones y resultaría excesivo incluir capturas de todo lo relevante, ya que ejecuta numerosos cálculos para obtener los datos para la gráfica. Por eso, se supondrá que las variables, que aparecen en la figura 121, \$dataUsuarioInfracciones y \$dataTotalInfracciones se encuentra cargadas con los datos necesarios para pintarlos en la gráfica.

Para que cuando el usuario cambie en el selector el valor del tipo de vehículo y este cambio tenga efecto sobre la gráfica y sus datos, ha sido necesario crear una nueva función AJAX. La llamada a esta función AJAX se puede ver en la figura 118. La función ajax_infracciones_vehiculo_callback () recarga la parte del formulario que hace referencia a la gráfica, actualizando los datos en función del vehículo seleccionado. El código de esta función se puede ver en la figura 122.

```

// Función Ajax que actualiza el valor del tipo de vehículo para la gráfica
function ajax_infracciones_vehiculo_callback($form, $form_state) {
    return $form['rjsim_fieldset_infracciones_simulacion']['chart_infracciones_simulacion_1'];
}

```

Figura 122: Nueva función ajax_partidas_vehiculo_callback() dentro del fichero rjsimulador.forms.inc.

La información de los ejes horizontales de las gráficas se ha cambiado, la palabra Simulación se ha cambiado por Escenario y se ha traducido la información al inglés. Esta información sobre los escenarios se extrae de la base de datos, como se comentó anteriormente. El título del eje vertical de las gráficas también se ha modificado de la siguiente manera: “Number of games” por “Number of simulations” y “Average number of infractions by played game” por “Average number of traffic offenses per scenario”.

4.5.1.2 Modificaciones de la parte intermedia de la página My data about the Simulation

Ahora se pasará a analizar la parte intermedia de la página, que abarca las dos graficas que se sitúan bajo las gráficas que aparecen en la parte superior de la pantalla.

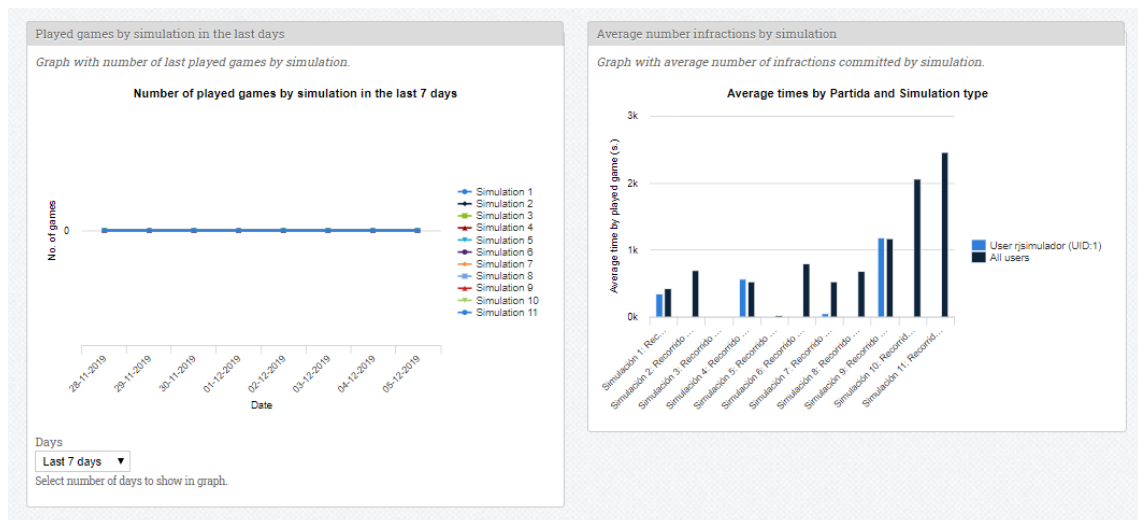


Figura 123: Parte intermedia de la página “My data about the Simulation” antes de la actualización de la plataforma.

En la parte intermedia de la página se han realizado varias modificaciones. En esta parte existen dos gráficas, en ambas se han cambiado los títulos principales, de la siguiente manera: “Played games by simulation in the last” por “Total number of simulations fulfilled by the user in the last days” y “Average number infractions by simulation” por “Average time to drive in each scenario”.

En estas gráficas se ha eliminado su descripción porque resultaba innecesaria, ya que no aportaba ningún tipo de información adicional. En las gráficas también se ha modificado su título. Esto ayuda a los usuarios a la hora de analizar los datos. El cambio realizado es el siguiente: “Number of played games by simulation in the last X days” por “Total number of simulations fulfilled by the user in the last X days” y “Average times by Partida and Simulation type” por “Average time to drive in each scenario”.

Estos cambios son meramente visuales y hacen la plataforma más fácil de utilizar para los usuarios. Se puede ver cómo ha cambiado la página de la plataforma si se comparan las figuras 123 y 124. El título de la primera gráfica de la parte intermedia se ve afectado por el valor que se seleccione en el selector “Days”, cambiando el valor de X por los días que elige el usuario en el selector.

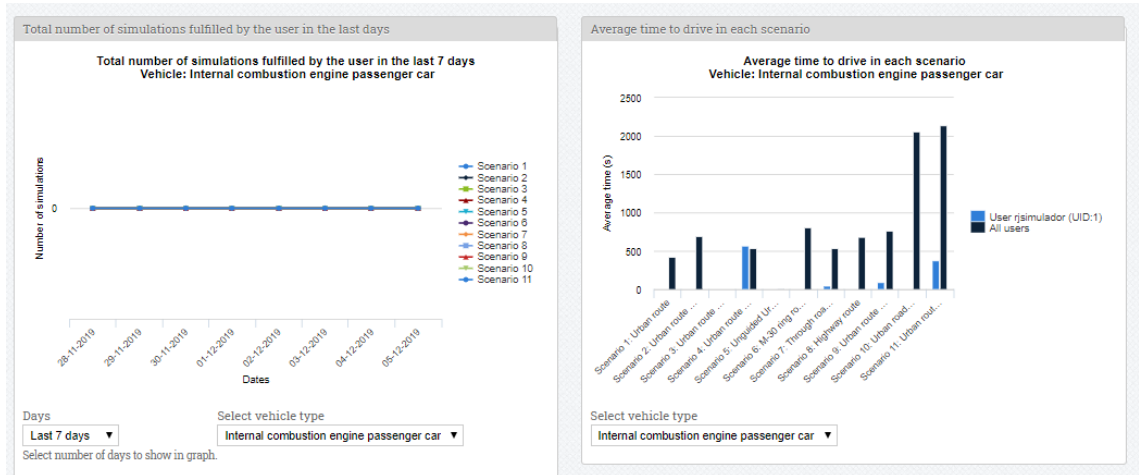


Figura 124: Parte intermedia de la página “My data about the Simulation” después de la actualización de la plataforma.

En la parte intermedia de la página se ha incluido un selector bajo las gráficas. Este selector permite a los usuarios filtrar las simulaciones por el tipo de vehículo con el que las han realizado. Esta nueva característica facilita el análisis de los datos. Para poder incluir el selector en la gráfica “Total number of simulations fulfilled by the user in the last X days” se ha tenido que modificar la función `rjsim_simulaciones_upper_left_form()` del fichero `rjsimulador.forms.inc`. En esta función se ha tenido que incluir el código de la figura 125 para implementar el selector.

```
// Apartado del formulario que hace referencia al select tipo de vehículo
$form['rjsim_fieldset_partidas_ultimos_dias']['select_vehiculo_dias'] = array(
  '#type' => 'select',
  '#title' => t('Select vehicle type'),
  '#options' => array(
    1 => t('Internal combustion engine passenger car'),
    2 => t('Electric passenger car'),
    3 => t('Internal combustion engine city bus'),
    4 => t('Electric city bus'),
    5 => t('Internal combustion engine truck'),
    6 => t('Electric truck')
  ),
  '#default_value' => 1,
  //'#description' => t('Select the type of vehicle.'),
  '#ajax' => array(
    'callback' => 'ajax_partidas_vehiculo_ultimos_dias_callback',
    'wrapper' => 'div-chart-partidas-ultimos-dias',
  ),
);
```

Figura 125: Código incluido en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` para generar el selector de la gráfica “Total number of simulations fulfilled by the user in the last X days”.

El título de la gráfica también se ha modificado en función del selector añadiendo el tipo de vehículo seleccionado por el usuario. Al analizar la

gráfica, los usuarios tendrán claros los datos que se muestran, la elección del selector filtra los datos de las simulaciones por el tipo de vehículo con el que se realizaron. Estos cambios se han llevado a cabo en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en las figuras 126 y empleando un codificación como las que se muestra en las figuras 113 y 120. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector que influye sobre el título de la gráfica.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['select_vehiculo_dias']) ? $form_state['values']['select_vehiculo_dias'] : 1;
```

Figura 126: Código incluido en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del selector de vehículos.

Gracias al valor de la variable `$vehiculo`, se genera el título para la gráfica, utilizando unas líneas código similares a las que se muestran en las figuras 113 y 120.

Para que los cambios se apliquen en el título de la gráfica, se ha modificado la parte del formulario que hace referencia a la misma en la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en la figura 127.

```
$form['rjsim_fieldset_partidas_ultimos_dias'] = array(
  '#type' => 'fieldset',
  '#title' => t('Total number of simulations fulfilled by the user in the last days'),
  '#collapsible' => FALSE,
  '#attributes' => array('class' => array('rjsim-margin'))
);

// Modificamos el apartado del formulario que hace referencia a la gráfica para que interactúe con el select tipo de vehículo
$form['rjsim_fieldset_partidas_ultimos_dias']['chart_partidas_ultimos_dias'] = array(
  '#type' => 'chart',
  '#chart_type' => 'line',
  '#chart_library' => 'highcharts',
  '#title' => t('Total number of simulations fulfilled by the user in the last @numberDays days'. $tituloGrafico, array('@numberDays' => $ultimosXDias)),
  '#title_font_weight' => 'bold',
  '#prefix' => '<div id="div-chart-partidas-ultimos-dias">',
  '#suffix' => '</div>',
  '#axis' =>
  array(
    array(
      '#type' => 'chart_xaxis',
      '#title' => t('Dates'),
      '#labels' => $xDayLabels,
    ),
  ),
  '#yaxis' =>
  array(
    array(
      '#type' => 'chart_yaxis',
      '#title' => t('Number of simulations'),
    ),
  ),
);

foreach ($arrayPartidasJugadasPorDia as $idSimulacion => $arrayPartidasPorSimulacion) {
  $arrayPartidasPorSimulacion = array_reverse($arrayPartidasPorSimulacion);
  $form['rjsim_fieldset_partidas_ultimos_dias']['chart_partidas_ultimos_dias'][$idSimulacion] = array(
    '#type' => 'chart_data',
    '#title' => t('Scenario @idSimulacion', array('@idSimulacion' => $idSimulacion)),
    '#data' => $arrayPartidasPorSimulacion,
  );
}
```

Figura 127: Código de la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` que genera la gráfica.

Se ha tenido que modificar la codificación de la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en figura 128. Como se comentará más adelante, se ha tenido que implementar código para que la elección del usuario en el selector de vehículos se vea reflejada en la gráfica. En

ella se puede ver que se recuperan los datos de las simulaciones en función del tipo de vehículo que el usuario que accede a la página elige en el selector.

```
// Recuperamos tipos de simulaciones
$tiposSimulacion = Constants::getTiposSimulacion();
$arrayPartidasJugadasPorDia = array();
foreach ($tiposSimulacion as $idSimulacion => $nombreSimulacion) {
    $arrayPartidasJugadasPorDia[$idSimulacion][] = $gestorSimulaciones->getUsuarioActual()
        ->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehiculo)
        ->filterBy(new FilterByFecha($fechas))
        ->count();
}

$xDaysLabels[] = $fechaInicio->format( format: 'd-m-Y');

for ($index = 0; $index < $ultimosXDias; $index++) {
    $fechaFin = new DateTime($fechaInicio->format( format: 'Y-m-d'));
    $fechaInicio->modify( modify: '-1 day');
    $fechas = array('fecha_inicio' => $fechaInicio, 'fecha_fin' => $fechaFin);
    // Recorremos la lista de tipos de escenario en función del id de un escenario y el nombre de un escenario
    foreach ($tiposSimulacion as $idSimulacion => $nombreSimulacion) {
        // Obtenemos el número de partidas del usuario actual para un escenario y un tipo de vehículo, filtrándolas por una fecha en concreto
        $arrayPartidasJugadasPorDia[$idSimulacion][] = $gestorSimulaciones->getUsuarioActual()
            ->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion,$vehiculo)
            ->filterBy(new FilterByFecha($fechas))
            ->count();
    }
}
$xDaysLabels[] = $fechaInicio->format( format: 'd-m-Y');
```

Figura 128: Código modificado de la función `rjsim_simulaciones_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del usuario actual.

Como se puede apreciar en la figura 128, se utiliza dos veces la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo)`, creada en el fichero `UsuarioSimulacion.php`, para recuperar los datos necesarios para la gráfica en función del vehículo seleccionado en el selector. De esta nueva función se habló en apartados anteriores y se mostró su código en la figura 36. Por otro lado, se puede ver que los datos que se han recuperado en la figura 128 para el usuario seleccionado se utilizan para pintarlos en el gráfico en la figura 129.

Para que el funcionamiento sea correcto ha sido necesario crear una nueva función AJAX que se ejecuta cuando cambia el valor del selector. La llamada a esta función AJAX se puede ver en la figura 125. La función `ajax_partidas_vehiculo_ultimos_dias_callback()` recarga la parte del formulario que hace referencia a la gráfica actualizando los datos en función del vehículo seleccionado. Se puede ver esta nueva función en la figura 129.

```
// Función Ajax que actualiza el valor del tipo de vehículo para la gráfica
function ajax_partidas_vehiculo_ultimos_dias_callback($form, $form_state) {
    return $form['rjsim_fieldset_partidas_ultimos_dias']['chart_partidas_ultimos_dias'];
}
```

Figura 129: Nueva función `ajax_partidas_vehiculo_callback()` dentro del fichero `rjsimulador.forms.inc`.

Para poder incluir el nuevo selector de tipo de vehículo en la gráfica “Average time to drive in each scenario”, se ha tenido que modificar la función `rjsim_simulaciones_upper_right_form()` del fichero

rjsimulador.forms.inc. Se ha tenido que incluir el código de la figura 130 para implementar el selector.

```
// Apartado del formulario que hace referencia al select tipo de vehículo
$form['rjsim_fieldset_tiempo_partida']['select_vehiculo_tiempo_simulacion'] = array(
  '#type' => 'select',
  '#title' => t( string: 'Select vehicle type'),
  '#options' => array(
    1 => t( string: 'Internal combustion engine passenger car'),
    2 => t( string: 'Electric passenger car'),
    3 => t( string: 'Internal combustion engine city bus'),
    4 => t( string: 'Electric city bus'),
    5 => t( string: 'Internal combustion engine truck'),
    6 => t( string: 'Electric truck')
  ),
  '#default_value' => 1,
  '#ajax' => array(
    'callback' => 'ajax_infracciones_vehiculo_tiempo_partida_callback',
    'wrapper' => 'div-chart-tiempo-por-partida-simulacion',
  ),
);
```

Figura 130: Código incluido en la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc` para generar el selector de la gráfica “Average time to drive in each scenario”.

El título de esta gráfica se ha modificado añadiendo el tipo de vehículo seleccionado por el usuario. Como en gráficas anteriores, la elección del selector filtra los datos de las simulaciones por el tipo de vehículo con el que se realizaron. Estos cambios se han llevado a cabo en la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en la figura 131 y utilizando unas líneas de código similares a las de la figura 113. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector que influye sobre el título.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehicul = isset($form_state['values']['select_vehiculo_tiempo_simulacion']) ? $form_state['values']['select_vehiculo_tiempo_simulacion'] : 1;
```

Figura 131: Código incluido en la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del selector de vehículos.

En función del valor que se recupera en la variable `$vehicul`, se genera el título para la gráfica, utilizando unas líneas código similares a las que se muestran en las figuras 113 y 120.

Se ha modificado la parte del formulario que hace referencia a la gráfica en la función `rjsim_simulaciones_upper_right_form()`, dentro del fichero `rjsimulador.forms.inc`, como se muestra en la figura 132.

```

$form['rjsim_fieldset_tiempo_partida'] = array(
    '#type' => 'fieldset',
    '#title' => t( string: 'Average time to drive in each scenario'),
    '#collapsible' => FALSE,
    '#attributes' => array('class' => array('rjsim-margin'))
);
// Modificamos el apartado del formulario que hace referencia a la gráfica para que interactúe con el select tipo de vehículo
$form['rjsim_fieldset_tiempo_partida']['chart_tiempo_por_partida_simulacion'] = array(
    '#type' => 'chart',
    '#chart_type' => 'column',
    '#chart_library' => 'highcharts',
    '#title' => t( string: 'Average time to drive in each scenario'. $tituloGrafic),
    '#title_font_weight' => 'bold',
    '#prefix' => '<div id="div-chart-tiempo-por-partida-simulacion">',
    '#suffix' => '</div>',
    'tiempo_usuario' => array(
        '#type' => 'chart_data',
        '#title' => t( string: 'User @userName (UID:@userID)', array(
            '@userName' => $gestorSimulaciones->getUsuarioActual()->getName(),
            '@userID' => $gestorSimulaciones->getUsuarioActual()->getUid()
        )),
        '#data' => $dataUsuarioTiempo,
    ),
    'tiempos_totales' => array(
        '#type' => 'chart_data',
        '#title' => t( string: 'All users'),
        '#data' => $dataTotalTiempo,
    ),
    'xaxis' =>
    array(
        '#type' => 'chart_xaxis',
        '#labels' => $labelXAxis
    ),
    'yaxis' =>
    array(
        '#type' => 'chart_yaxis',
        '#title' => 'Average time (s)'
    )
);

```

Figura 132: Código incluido en la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc` que genera la gráfica.

Se ha modificado la codificación de la función `rjsim_simulaciones_upper_right_form()` dentro del fichero `rjsimulador.forms.inc`. Esta función ha sufrido numerosas modificaciones y resultaría excesivo incluir capturas de todo lo relevante, como se comentó en el apartado anterior. Por eso, se supondrá que las variables que aparecen en la figura 132 `$dataUsuarioTiempo` y `$dataTotalTiempo` se encuentra cargadas con los datos necesarios para pintarlos en la gráfica. Para que los cambios en selector de vehículos tengan efecto sobre la gráfica y sus datos, ha sido necesario crear una nueva función AJAX. La llamada a esta función AJAX se puede ver en la figura 130. La función `ajax_infracciones_vehiculo_callback()` recarga la parte del formulario que hace referencia a la gráfica actualizando los datos en función del vehículo seleccionado. Esta función se puede ver en la figura 133.

```

// Función Ajax que actualiza el valor del tipo de vehículo para la gráfica
function ajax_infracciones_vehiculo_tiempo_partida_callback($form, $form_state) {
    return $form['rjsim_fieldset_tiempo_partida']['chart_tiempo_por_partida_simulacion'];
}

```

Figura 133: Nueva función `ajax_infracciones_vehiculo_tiempo_partida()` dentro del fichero `rjsimulador.forms.inc`.

La información del eje horizontal de la gráfica “Average time to drive in each scenario” se ha cambiado la palabra “Simulación” por “Escenario” y

se ha traducido la información a inglés. Como se ha comentado en numerosas ocasiones, estos datos se cambiaron directamente en la base de datos. En la gráfica “Total number of simulations fulfilled by the user in the last X days”, se ha cambiado el título del eje horizontal y ha pasado de ser “Date” a “Dates”. El título del eje vertical de las gráficas también se ha modificado de la siguiente manera: “No. of games” por “Number of simulations” y “Average time by played game (s)” por “Average time (s)”.

4.5.1.3 Modificaciones de la parte inferior de la página My data about the Simulation

En esta parte, se comentarán las modificaciones realizadas en la parte inferior de la página. En esta parte de la página, existe un gráfico circular con datos de las simulaciones y una tabla en la que los usuarios pueden seleccionar los datos a analizar en función del escenario.

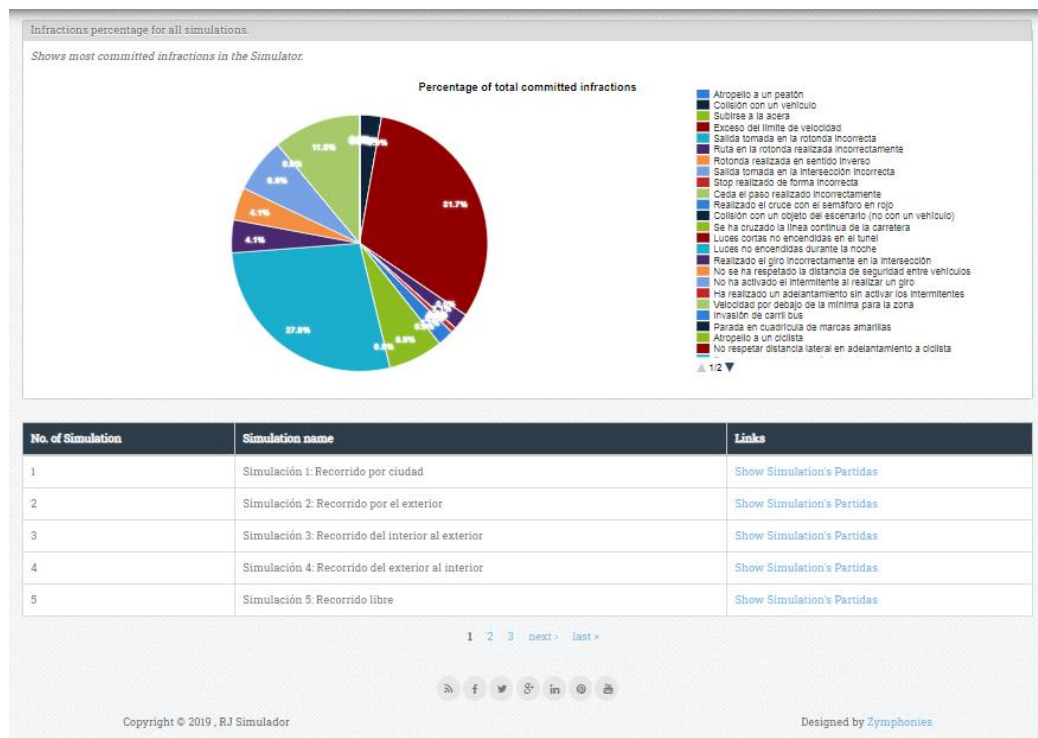


Figura 134: Parte inferior de la página “My data about the Simulation” antes de la actualización de la plataforma.

En la parte final de la página de la plataforma se han realizado varias modificaciones. En el gráfico circular se ha cambiado el título principal, de la siguiente manera: “Infractions percentage for all simulations” por “Percentages of traffic offences for all simulations”.

En este gráfico se ha eliminado su descripción porque resultaba innecesaria, ya que no aportaba ningún tipo de información adicional. En el gráfico también se ha modificado su título para adecuarlo más a los datos que contiene. El cambio realizado es el siguiente: “Percentage of

total committed infractions” por “Percentages of traffic offences for all simulations”.

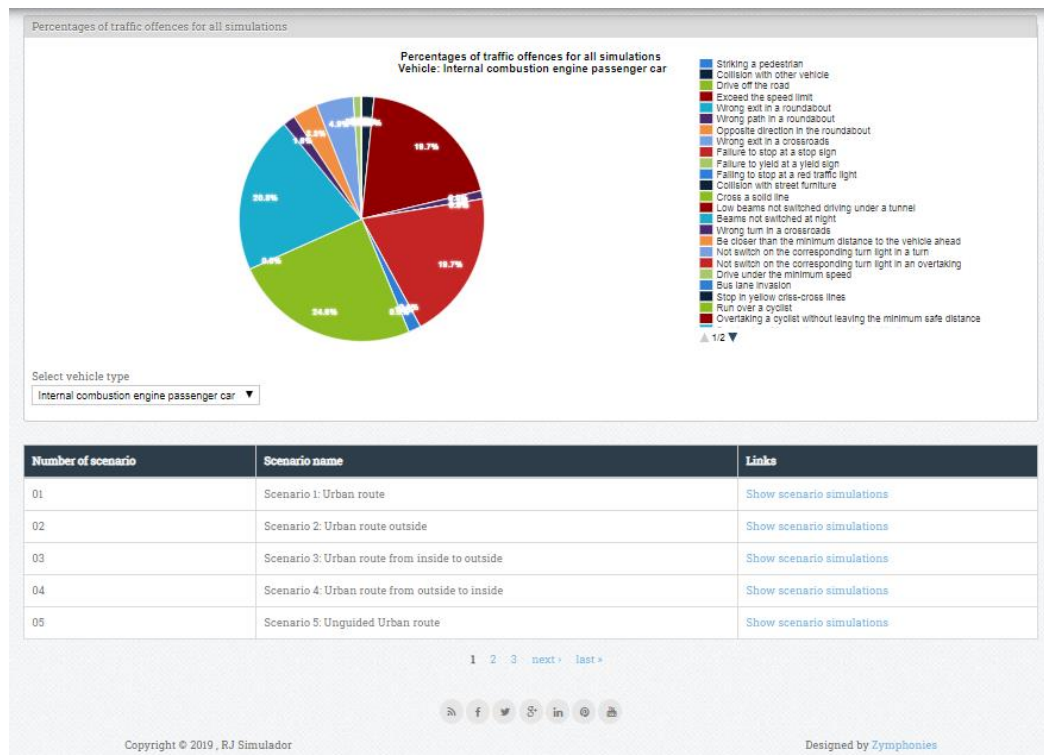


Figura 135: Parte inferior de la página “My data about the Simulation” después de la actualización de la plataforma.

En la tabla que existe en la parte inferior de esta página también se han realizado numerosas modificaciones, para adecuar más la información a los datos que se muestran en ella. Las modificaciones realizadas en los títulos de la tabla son las siguientes “No. Simulation” por “Number of scenario” y “Simulation name” por “Scenario name”.

En los datos que incluye la tabla, también se han realizado modificaciones, como los títulos de los escenarios que se han pasado a inglés y se ha cambiado la palabra “Simulación” por “Scenario”. Como se ha comentado ya numerosas veces, este cambio se ha hecho directamente en la base de datos y afecta a toda la plataforma. También, se ha realizado la siguiente modificación en los títulos de los hipertextos de la tabla “Show Simulation’s Partidas” por “Show scenario simulations”.

Todos estos cambios están dirigidos a mejorar la experiencia de usuario en esta página. Se pueden observar los cambios realizados si se comparan las figuras 134 y 135.

Por otro lado, se ha incluido un selector bajo el gráfico circular. Este selector permite a los usuarios filtrar las simulaciones por el tipo de vehículo con el que han realizado las simulaciones. Esta nueva característica facilita el análisis de los datos de las simulaciones y ayuda a saber a los usuarios con qué tipo de vehículo han realizado cada simulación. Para incluir este selector en la base del gráfico circular, se ha tenido que modificar la función `rjsim_simulaciones_main_form()` del

fichero `rjsimulador.forms.inc`. En esta función se ha tenido que incluir el código de la figura 136 para implementar el selector.

```
// Apartado del formulario que hace referencia al select tipo de vehículo
$form['rjsim_fieldset_infracciones']['select_vehiculo_infracciones_totales'] = array(
  '#type' => 'select',
  '#title' => t('Select vehicle type'),
  '#options' => array(
    1 => t('Internal combustion engine passenger car'),
    2 => t('Electric passenger car'),
    3 => t('Internal combustion engine city bus'),
    4 => t('Electric city bus'),
    5 => t('Internal combustion engine truck'),
    6 => t('Electric truck')
  ),
  '#default_value' => 1,
  /*#description' => t('Select the type of vehicle.'),
  '#ajax' => array(
    'callback' => 'ajax_infracciones_totales_vehiculo_callback',
    'wrapper' => 'div-chart-infracciones-totales-simulacion',
  ),
);
```

Figura 136: Código incluido en la función `rjsim_simulaciones_main_form()` dentro del fichero `rjsimulador.forms.inc` para generar el selector del gráfico “Percentages of traffic offences for all simulations”.

El título del gráfico se ha modificado en función del selector, añadiendo el tipo de vehículo seleccionado por el usuario. La elección del selector filtra los datos de las simulaciones por el tipo de vehículo con el que se realizaron. Estos cambios se han llevado a cabo en la función `rjsim_simulaciones_main_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en las figuras 137 y utilizando unas líneas código similares a las que se muestran en la figura 113. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector de vehículos incluido en esta gráfica.

```
// Recuperamos el tipo de vehículo
$vehiclo = isset($form_state['values']['select_vehiculo_infracciones_totales']) ? $form_state['values']['select_vehiculo_infracciones_totales'] : 1;
```

Figura 137: Código incluido en la función `rjsim_simulaciones_main_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del selector de vehículos.

Con el valor de la variable `$vehiclo` se genera el título para el gráfico, utilizando unas líneas de código parecidas a las que se muestran en las figuras 113 y 120.

Para que los cambios se apliquen en el título del gráfico, se ha modificado la parte del formulario que hace referencia al gráfico en la función `rjsim_simulaciones_main_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en la figura 138.

```

$form['rjsim_fieldset_infracciones'] = array(
    '#type' => 'fieldset',
    '#title' => t( string: 'Percentages of traffic offences for all simulations'),
    '#collapsible' => FALSE,
    '#attributes' => array('class' => array('rjsim-margin'))
);
// Modificamos el apartado del formulario que hace referencia a la gráfica para que interactúe con el select tipo de vehículo
$form['rjsim_fieldset_infracciones']['chart_infracciones'] = array(
    '#type' => 'chart',
    '#title' => t( string: 'Percentages of traffic offences for all simulations'. $titulGrfico),
    '#title_font_weight' => 'bold',
    '#prefix' => '<div id="div-chart-infracciones-totales-simulacion">',
    '#suffix' => '</div>',
    '#chart_type' => 'pie',
    '#chart_library' => 'highcharts',
    '#legend_position' => 'right',
    '#data_labels' => TRUE,
    '#tooltips' => TRUE,
    '#pie_data' =>
        array(
            '#type' => 'chart_data',
            '#title' => t( string: 'Infracciones'),
            '#labels' => $labels,
            '#data' => $data,
        )
    )
);

```

Figura 138: Código incluido en la función rjsim_simulaciones_main_form() dentro del fichero rjsimulador.forms.inc que genera el gráfico.

Se ha tenido que modificar la codificación de la función rjsim_simulaciones_main_form() dentro del fichero rjsimulador.forms.inc, como se muestra en la figura 139. En ella, se puede ver que se recuperan los datos de las simulaciones en función del tipo de vehículo que se elige en el selector.

```

// Obtenemos la lista de partidas en función del tipo de vehículo para el usuario actual
$listaPartidas = $gestorSimulaciones->getUsuarioActual()->retrieveAllPartidasVehiculo($vehciclo);

```

Figura 139: Código modificado de la función rjsim_simulaciones_main_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del usuario actual.

En la figura 139 se puede ver que se utiliza una nueva función retrieveAllPartidasVehiculo(\$vehciclo) creada en el fichero UsuarioSimulacion.php para recuperar la lista de simulaciones de un usuario en función de un tipo de vehículo. Estos datos son necesarios para el gráfico. El código de la función se puede ver en la figura 140. Por otro lado, en la figura 138 se puede ver que los datos que se han recuperado en la figura 139 del usuario en función del tipo de vehículo seleccionado se utilizan para pintarlos en el gráfico.

```

/**
 * Método que devuelve todas las partidas de un usuario para un tipo de vehículo.
 * @param int $vehiculo El id del vehículo.
 * @return ListaPartidas Lista de todas las partidas de ese usuario de todas las simulaciones.
 */
public function retrieveAllPartidasVehiculo($vehiculo) {
    $listaPartidas = new ListaPartidas();
    $listaPartidasVehi = new ListaPartidas();

    foreach ($this->getListaSimulaciones() as $simulacion) {
        $listaPartidas->mergeList($simulacion->getListaPartidas());
    }
    foreach ($listaPartidas as $partida) {
        if ($partida->getTipoVehiculo() == $vehiculo) {
            $listaPartidasVehi->add($partida);
            //break;
        }
    }
    return $listaPartidasVehi;
}

```

Figura 140: Nueva función retrieveAllPartidasVehiculo(\$vehiculo) dentro del fichero UsuarioSimulacion.php.

Se ha creado una nueva función AJAX que se ejecuta cuando cambia el valor del selector. La llamada a esta función AJAX se puede ver en la figura 136. La función ajax_infracciones_totales_vehiculo_callback () recarga la parte del formulario que hace referencia a la gráfica actualizando los datos en función del vehículo seleccionado, esta función se puede ver en la figura 141.

```

// Función Ajax que actualiza el valor del tipo de vehículo para la gráfica
function ajax_infracciones_totales_vehiculo_callback($form, &$form_state) {
    return $form['rjsim_fieldset_infracciones']['chart_infracciones'];
}

```

Figura 141: Nueva función ajax_partidas_vehiculo_callback() dentro del fichero rjsimulador.forms.inc.

En este gráfico, en la parte lateral derecha se aprecia que se ha modificado el título de las infracciones, traduciéndolas a inglés. Como se ha comentado con anterioridad, esta modificación se ha tenido que realizar sobre la base de datos e influye sobre toda la plataforma web. Todas estas modificaciones que se han realizado en esta página, se han realizado con la intención de mejorar la experiencia del usuario y facilitar el análisis de los datos a los usuarios.

Cuando, en la tabla de la parte inferior de la página “My data about the Simulation”, un usuario pulsa en uno de los enlaces que, después de la actualización, ha pasado a ser “Show scenario simulations”, accederá a la página que, antes de la actualización, se llamaba “Partidas by Simulation X” y que, ahora, ha pasado a llamarse “Simulations in scenario X” n, siendo X el número del escenario del que el usuario quiere analizar los datos.

4.5.2 Modificaciones de la página Simulations in escenario X

En la página “Partidas by Simulation X” y que ahora ha pasado a llamarse “Simulations in escenario X” después de la actualización, se han realizado numerosas modificaciones. Para analizar esta página de la plataforma con detalle se dividirá este análisis en dos partes. Empezaremos por la parte superior de la página.

4.5.2.1 Modificaciones de la parte superior de la página Simulations in escenario X

En la parte superior de esta página existen dos gráficas. En estas dos gráficas se han cambiado los títulos principales, de la siguiente manera: “Average consumption comparison” por “Comparison of average fuel consumption” y “Average time by Partida comparison” por “Average time to drive in scenario”.

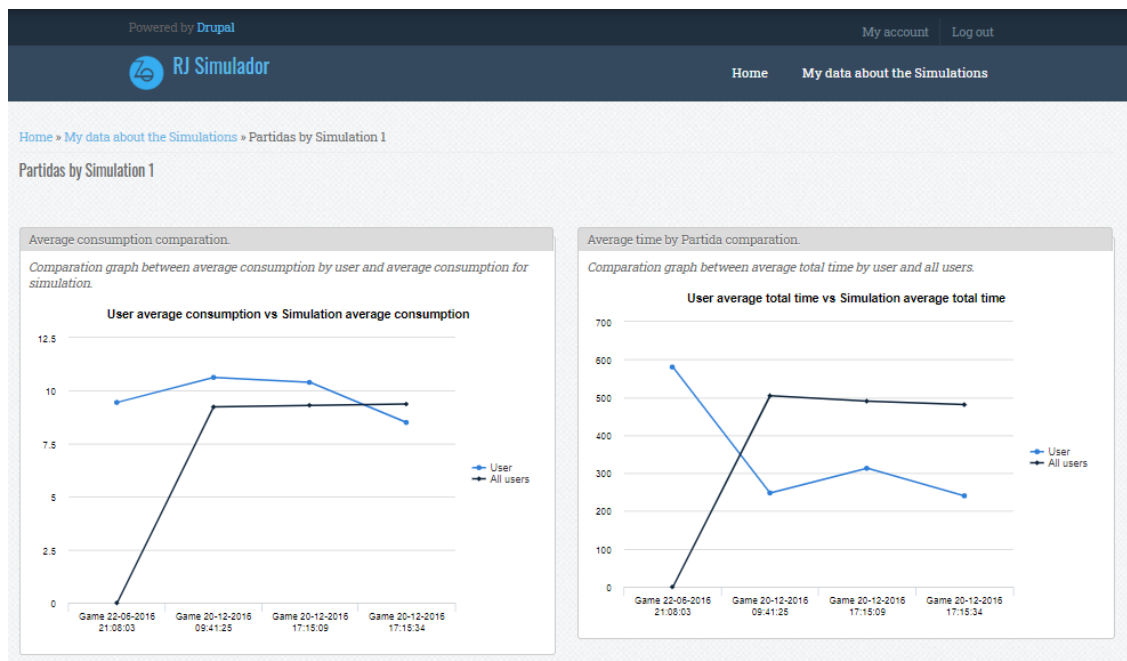


Figura 142: Parte superior de la página “Partidas by Simulation X” antes de la actualización de la plataforma.

En estas gráficas se han eliminado las descripciones porque resultaban innecesarias, ya que no aportaban ningún tipo de información adicional. En las gráficas también se ha modificado su título para adecuarlo más a los datos que contienen. Esto ayuda a los usuarios a la hora de analizar los datos y saber qué están viendo. El cambio realizado es el siguiente: “User average consumption vs Simulation average consumption” por “User average fuel consumption VS escenario average fuel consumption” y “User

average total time vs Simulation average total time” por “User average time VS scenario average time”.

Todas estas modificaciones, dirigidas a mejorar la información que se muestra a los usuarios de la plataforma, se pueden apreciar si se realiza una comparativa entre las figuras 142 y 143.

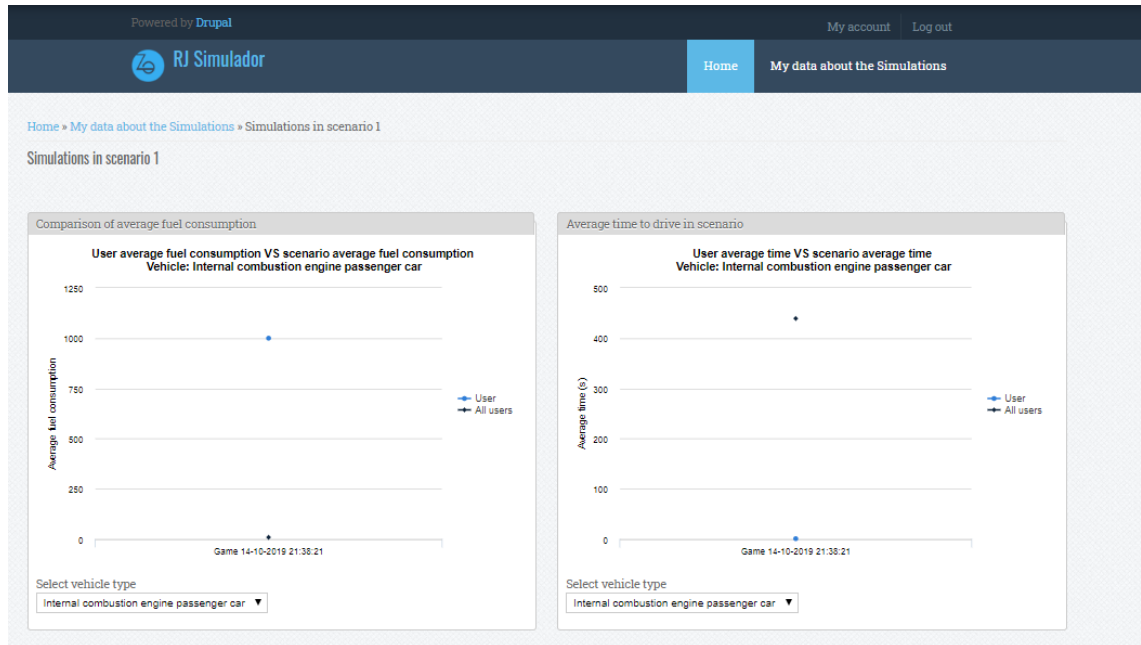


Figura 143: Parte superior de la página “Simulations in scenario X” después de la actualización de la plataforma.

También se ha añadido un selector bajo las gráficas. Este selector permite filtrar a los usuarios por el tipo de vehículo con el que han realizado las simulaciones.

Para poder incluir el selector en la gráfica “Comparison of average fuel consumption”, se ha tenido que modificar la función `rjsim_partidas_simulacion_upper_left_form()` del fichero `rjsimulador.forms.inc`. En esta función se ha tenido que incluir el código de la figura 144 para implementar el selector.

```
// Apartado del formulario que hace referencia al select tipo de vehículo
$form['rjsim_fieldset_consumo_medio']['select_vehiculo_consumo_medio'] = array(
  '#type' => 'select',
  '#title' => t('Select vehicle type'),
  '#options' => array(
    1 => t('Internal combustion engine passenger car'),
    2 => t('Electric passenger car'),
    3 => t('Internal combustion engine city bus'),
    4 => t('Electric city bus'),
    5 => t('Internal combustion engine truck'),
    6 => t('Electric truck')
  ),
  '#default_value' => 1,
  //'#description' => t('Select the type of vehicle.'),
  '#ajax' => array(
    'callback' => 'ajax_consumo_medio_vehiculo_callback',
    'wrapper' => 'div-chart-consumo-medio',
  ),
);
```

Figura 144: Código incluido en la función `rjsim_partidas_simulacion_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` para generar el selector de la gráfica “Comparison of average fuel consumption”.

En el título de esta gráfica se ha añadido el tipo de vehículo seleccionado por el usuario. La elección del selector filtra los datos de las simulaciones por el tipo de vehículo con el que se realizaron. Estos cambios se han llevado a cabo en la función `rjsim_partidas_simulacion_upper_left_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en las figuras 145 y empleando una codificación similar a la que se muestra en la figura 113. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['select_vehiculo_consumo_medio']) ? $form_state['values']['select_vehiculo_consumo_medio'] : 1;
```

Figura 145: Código incluido en la función `rjsim_partidas_simulacion_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del selector de vehículos.

Utilizando el valor de la variable `$vehiculo`, se genera el título para la gráfica, utilizando unas líneas de código similares a las de la figura 113. Para que los cambios en el título de la gráfica se apliquen en función del selector, se ha modificado la parte del formulario que hace referencia a la gráfica en la función `rjsim_partidas_simulacion_upper_left_form()` dentro del fichero `rjsimulador.forms.inc`, incluyendo la variable `$titulgrfico` que contiene los datos del tipo de vehículo dentro del contenido de la etiqueta `#title`, como se muestra en la figura 146.

```
$form['rjsim_fieldset_consumo_medio'] = array(
  '#type' => 'fieldset',
  '#title' => t( string: 'Comparison of average fuel consumption'),
  '#collapsible' => FALSE,
  '#attributes' => array('class' => array('rjsim-margin'))
);

// Modificamos el apartado del formulario que hace referencia a la gráfica para que interactúe con el select tipo de vehículo
$form['rjsim_fieldset_consumo_medio']['chart_consumo_medio'] = [
  '#type' => 'chart',
  '#chart_type' => 'line',
  '#chart_library' => 'highcharts',
  '#title' => t( string: 'User average fuel consumption VS scenario average fuel consumption' . $titulgrfico),
  '#title_font_weight' => 'bold',
  '#prefix' => '<div id="div-chart-consumo-medio">',
  '#suffix' => '</div>',
  '#usuario' =>
  [
    '#type' => 'chart_data',
    '#title' => t( string: 'User'),
    '#data' => $dataUsuario,
  ],
  '#todos' =>
  [
    '#type' => 'chart_data',
    '#title' => t( string: 'All users'),
    '#data' => $dataTodos,
  ],
  '#xaxis' =>
  [
    '#type' => 'chart_xaxis',
    '#labels' => $labelAxisX,
  ],
  '#yaxis' =>
  [
    '#type' => 'chart_yaxis',
    '#title' => 'Average fuel consumption'
  ]
];
```

Figura 146: Código de la función `rjsim_partidas_simulacion_upper_left_form()` dentro del fichero `rjsimulador.forms.inc` que genera la gráfica.

Se ha modificado el código de la función `rjsim_partidas_simulacion_upper_left_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en las siguientes figuras 147 y 148. En ellas se puede ver que se recuperan los datos de las simulaciones

en función del tipo de vehículo que el usuario que accede a la página elige en el selector.

```
// Obtenemos la lista de partidas del usuario actual en función del id de un escenario y el tipo de vehículo
$listaPartidasUsuario = $gestorSimulaciones->getUsuarioActual()->retrieveAllPartidasByIdSimulacionVehiculo($id_simulacion, $vehiculo);
$listaPartidasUsuario->sortBy( sortField: 'Fecha', sort: 'ASC');
```

Figura 147: Código modificado de la función risim_partidas_simulacion_upper_left_form() dentro del fichero rjsimulador.forms.inc para obtener los datos del usuario actual.

En la figura 147 se puede ver que se utiliza la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo)` creada en el fichero `UsuarioSimulacion.php` para recuperar los datos necesarios para la gráfica en función del escenario y el vehículo seleccionado. De esta función se viene hablando en apartados anteriores y se mostró en la figura 36.

```
$dataTodos=dataTodosConsumoMedio($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario);
```

Figura 148: Código modificado de la función risim_partidas_simulacion_upper_left_form() dentro del fichero rjsimulador.forms.inc para obtener los datos de todos los usuarios.

Como se puede ver en la figura 148 ha sido necesario crear una nueva función para obtener los datos de todos los usuarios, para que el desarrollo fuera más ordenado, el código de la función `dataTodosConsumoMedio($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario)` se puede ver en la figura 149.

```
function dataTodosConsumoMedio($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario) {
    $gestorSimulaciones = new GestorSimulaciones();
    // Lista de partidas de todos los usuarios para la simulación
    $dataRetriever = new ListaUsuariosDataRetriever($gestorSimulaciones->getListaTodosUsuarios());
    //Modificado
    // Obtenemos la lista de partidas de todos los usuarios en función del id de un escenario y el tipo de vehículo
    $listaPartidasTodos = $dataRetriever->retrieveAllPartidasByIdSimulacionVehiculo($id_simulacion, $vehiculo);

    foreach (new LimitIterator($listaPartidasUsuario, $ultimasPartidas) as $partida) {
        $listaResultado = $listaPartidasTodos->filterBy(new FilterByFecha(array (FilterByFecha::FECHA_FIN => $partida->getFechaAsObject())));
        $dataTodos[] = $listaResultado->calculateData(new CalculateAverageData( field: CalculateAverageData::CONSUMO_MEDIO));
    }
    return $dataTodos;
}
```

Figura 149: Nueva función dataTodosConsumoMedio(\$id_simulacion,\$vehiculo,\$ultimasPartidas,\$listaPartidasUsuario) dentro del fichero rjsimulador.forms.inc para obtener los datos de todos los usuarios

Por otro lado, en la nueva función `dataTodosConsumoMedio($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario)` también se ha utilizado la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo)` del fichero `ListaUsuariosDataRetriever.php` para recuperar los datos de todos los usuarios necesarios para la gráfica. De esta función ya se habló con anterioridad y se puede ver en la figura 45. Los datos que se recuperan en las figuras 147 y 148 se tratan para poder ser mostrados en la gráfica,

como se muestra en la figura 146. El tratamiento de los datos resulta excesivo mostrarlo en capturas del código, por lo que no se ha incluido.

En el fichero `rjsimulador.forms.inc` ha sido necesario crear una nueva función AJAX que se ejecuta cuando cambia el valor del selector. La llamada a esta función AJAX se puede ver en la figura 144. La función `ajax_consumo_medio_vehiculo_callback()` recarga la parte del formulario que hace referencia a la gráfica actualizando los datos en función del vehículo seleccionado. El código de esta función se puede ver en la figura 150.

```
// Función Ajax que actualiza el valor del tipo de vehículo para la gráfica
function ajax_consumo_medio_vehiculo_callback($form, $form_state) {
    return $form['rjsim_fieldset_consumo_medio']['chart_consumo_medio'];
}
```

Figura 150: Nueva función `ajax_consumo_medio_vehiculo_callback()` dentro del fichero `rjsimulador.forms.inc`.

Para añadir el selector de tipo de vehículo en la gráfica “Average time to drive in scenario” se ha tenido que modificar la función `rjsim_partidas_simulacion_upper_right_form()` del fichero `rjsimulador.forms.inc`. En la función se ha tenido que incluir el código de la figura 151 para implementar el selector.

```
// Apartado del formulario que hace referencia al select tipo de vehículo
$form['rjsim_fieldset_tiempo_medio']['select_vehiculo_tiempo_medio'] = array(
  '#type' => 'select',
  '#title' => t('Select vehicle type'),
  '#options' => array(
    1 => t('Internal combustion engine passenger car'),
    2 => t('Electric passenger car'),
    3 => t('Internal combustion engine city bus'),
    4 => t('Electric city bus'),
    5 => t('Internal combustion engine truck'),
    6 => t('Electric truck')
  ),
  '#default_value' => 1,
  '##description' => t('Select the type of vehicle.'),
  '#ajax' => array(
    'callback' => 'ajax_tiempo_medio_vehiculo_callback',
    'wrapper' => 'div-chart-tiempo-medio',
  ),
);
```

Figura 151: Código incluido en la función `rjsim_partidas_simulacion_upper_right_form()` dentro del fichero `rjsimulador.forms.inc` para generar el selector de la gráfica “Average time to drive in scenario”.

Se ha añadido el tipo de vehículo seleccionado por el usuario al título de la gráfica. Estos cambios se han llevado a cabo en la función `rjsim_partidas_simulacion_upper_right_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en las figuras 152 y con una codificación muy similar a la que se muestra en la figura 113. En las figuras solo se muestran las modificaciones relacionadas con el nuevo selector que influye sobre el título.

```
// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['select_vehiculo_tiempo_medio']) ? $form_state['values']['select_vehiculo_tiempo_medio'] : 1;
```

Figura 152: Código incluido en la función `rjsim_partidas_simulacion_upper_right_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del selector de vehículos.

En función de la variable \$vehículo se genera el título para la gráfica, utilizando unas líneas de código similares a las que se muestran en la figura 113.

Para que el título de la gráfica cambie en función de la elección del selector, se ha modificado la parte del formulario que hace referencia a la gráfica en la función rjsim_partidas_simulacion_upper_right_form() dentro del fichero rjsimulador.forms.inc, como se puede ver en la figura 153.

```

$form['rjsim_fieldset_tiempo_medio'] = array(
    '#type' => 'fieldset',
    '#title' => t( string: 'Average time to drive in scenario'),
    '#collapsible' => FALSE,
    '#attributes' => array('class' => array('rjsim-margin'))
);
// Modificamos el apartado del formulario que hace referencia a la gráfica para que interactúe con el select tipo de vehículo
$form['rjsim_fieldset_tiempo_medio']['chart_tiempo_medio'] = array(
    '#type' => 'chart',
    '#chart_type' => 'line',
    '#chart_library' => 'highcharts',
    '#title' => t( string: 'User average time VS scenario average time'. $titulgrafico),
    '#title_font_weight' => 'bold',
    '#prefix' => '<div id="div-chart-tiempo-medio">',
    '#suffix' => '</div>',
    '#usuario' =>
        array(
            '#type' => 'chart_data',
            '#title' => t( string: 'User'),
            '#data' => $dataUsuario,
        ),
    '#todos' =>
        array(
            '#type' => 'chart_data',
            '#title' => t( string: 'All users'),
            '#data' => $dataTodos,
        ),
    '#xaxis' =>
        array(
            '#type' => 'chart_xaxis',
            '#labels' => $labelAxisX,
        ),
    '#yaxis' =>
        array(
            '#type' => 'chart_yaxis',
            '#title' => 'Average time (s)'
        )
);

```

Figura 153: Código en la función rjsim_partidas_simulacion_upper_right_form() dentro del fichero rjsimulador.forms que genera la gráfica.

En la función rjsim_partidas_simulacion_upper_right_form() del fichero rjsimulador.forms.inc se han tenido que realizar modificaciones, para poder analizar los datos de manera correcta, como se muestra en las siguientes figuras 154 y 155. En ellas se puede ver que se recuperan los datos de las simulaciones en función del tipo de vehículo y el escenario.

```

// Obtenemos la lista de partidas del usuario actual en función del id de un escenario y el tipo de vehículo
$listaPartidasUsuario = $gestorSimulaciones->getUsuarioActual()->retrieveAllPartidasByIdSimulacionVehiculo($id_simulacion, $vehiculo);
$listaPartidasUsuario->sortBy( sortField: 'Fecha', sort: 'ASC');

```

Figura 154: Código en la función rjsim_partidas_simulacion_upper_right_form() dentro del fichero rjsimulador.forms. para obtener los datos del usuario actual.

En la figura 154 se puede ver que se utiliza la función retrieveAllPartidasByIdSimulacionVehiculo(\$idSimulation,\$vehiculo) creada en el fichero UsuarioSimulacion.php para recuperar los datos

necesarios para la gráfica. De esta función se habla en apartados anteriores y se mostró en la figura 36.

```
$dataTodos=dataTodosTiempoTotal($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario);
```

Figura 155: Código en la función `rjsim_partidas_simulacion_upper_right_form()` dentro del fichero `rjsimulador.forms`. para obtener los datos de todos los usuarios.

Como se puede ver en la figura 155 se ha creado una nueva función para obtener los datos de todos los usuarios de una manera más ordenada, el código de la función `dataTodosTiempoTotal($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario)` se puede ver en la figura 156.

```
function dataTodosTiempoTotal($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario){
    $gestorSimulaciones = new GestorSimulaciones();
    // Lista de partidas de todos los usuarios para la simulación
    $dataRetriever = new ListaUsuariosDataRetriever($gestorSimulaciones->getListaTodosUsuarios());
    //Modificado
    // Obtenemos la lista de partidas de todos los usuarios en función del id de un escenario y el tipo de vehículo
    $listaPartidasTodos = $dataRetriever->retrieveAllPartidasByIdSimulacionVehiculo($id_simulacion, $vehiculo);

    foreach (new LimitIterator($listaPartidasUsuario, $ultimasPartidas) as $partida) {
        $listaResultado = $listaPartidasTodos->filterBy(new FilterByFecha(array(FilterByFecha::FECHA_FIN => $partida->getFechaAsObject())));
        $dataTodos[] = $listaResultado->calculateData(new CalculateAverageData( field: CalculateAverageData::TIEMPO_TOTAL));
    }
    return $dataTodos;
}
```

Figura 156: Nueva función `dataTodosTiempoTotal($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario)` dentro del fichero `rjsimulador.forms.inc` para obtener los datos de todos los usuarios

También en la nueva función `dataTodosTiempoTotal($id_simulacion,$vehiculo,$ultimasPartidas,$listaPartidasUsuario)` se ha utilizado la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo)` del fichero `ListaUsuariosDataRetriever.php`. De esta función ya se habló con anterioridad y se puede ver en la figura 45. Los datos que se recuperan en las figuras 154 y 155 se tratan para poder ser representados en la gráfica, como se muestra en la figura 153. Como se ha comentado antes, este tratamiento de los datos resulta excesivo mostrarlo en capturas del código, por lo que no se ha incluido.

Ha sido necesario crear una nueva función AJAX que se ejecuta cuando cambia el valor del selector. La llamada a esta función AJAX se puede ver en la figura 151. La función `ajax_tiempo_medio_vehiculo_callback()` recarga la parte del formulario que hace referencia a la gráfica actualizando los datos en función del vehículo seleccionado. El código de esta nueva función del fichero `rjsimulador.forms.inc` se puede ver en la figura 157.

```
// Función Ajax que actualiza el valor del tipo de vehículo para la gráfica
function ajax_tiempo_medio_vehiculo_callback($form, $form_state) {
    return $form['rjsim_fieldset_tiempo_medio']['chart_tiempo_medio'];
}
```

Figura 157: Nueva función `ajax_consumo_medio_vehiculo_callback()` dentro del fichero `rjsimulador.forms.inc`.

También se han incluido títulos a los ejes verticales de las gráficas, para que el análisis de la información sea más fácil para los usuarios. Los títulos de los ejes verticales son los siguientes: “User average consumption vs Simulation average consumption” por “Average fuel consumption” y “User average total time vs Simulation average total time” por “Average time (s)”.

4.5.2.2 Modificaciones de la parte inferior de la página Simulations in scenario X

Por último, se comentarán las modificaciones realizadas en la parte inferior de la página. En esta parte de la página existe un gráfico de barras con datos de las infracciones cometidas en las simulaciones y una tabla en la que los usuarios pueden seleccionar los datos a analizar en función de la simulación.

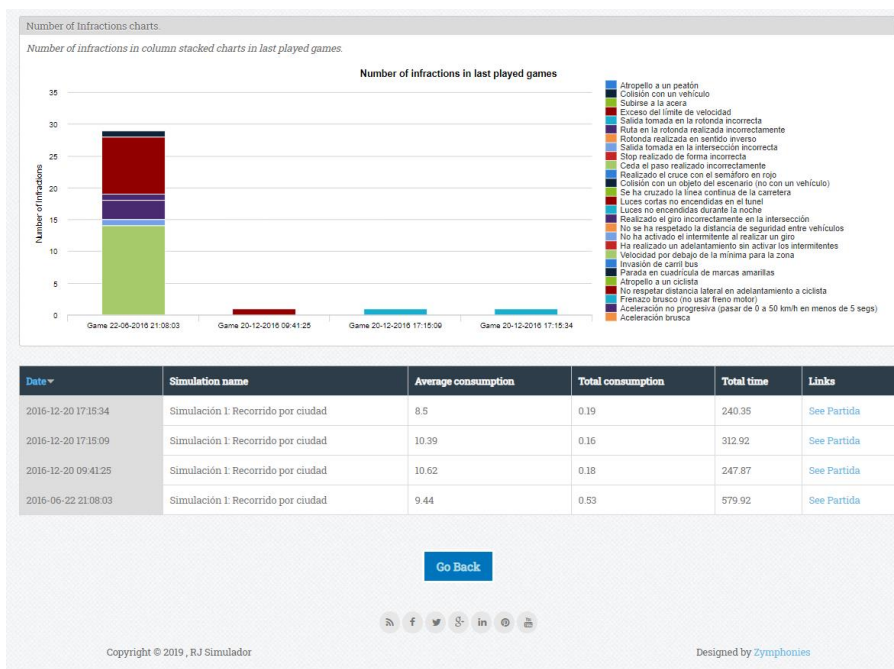


Figura 158: Parte inferior de la página “Partidas by Simulation X” antes de la actualización de la plataforma.

En esta parte de la página se han realizado varias modificaciones. En ella existe un gráfico de barras, del que se ha cambiado el título principal, de la siguiente manera: “Number of Infractions charts” por “Traffic offenses in last simulations in the scenario”.

En el gráfico se ha eliminado la descripción del mismo porque resultaba innecesaria, ya que no aportaba ningún tipo de información adicional. En él también se ha modificado su título del propio gráfico de barras para adecuarlo más a los datos que contienen. Esto ayuda a los usuarios a la hora de analizar los datos y saber qué están viendo. El cambio realizado es

el siguiente: “Number of Infractions in last played games” por “Traffic offenses in last Simulations”.

Si se comparan las figuras 158 y 159, se pueden ver las modificaciones realizadas, que se acaban de comentar, referentes al gráfico de barras.

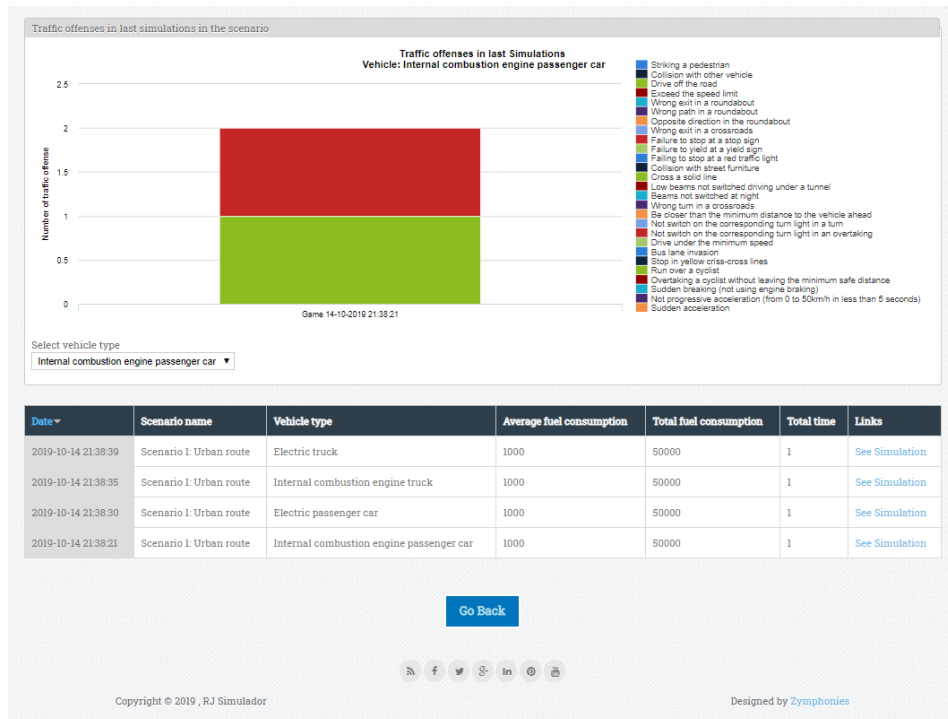


Figura 159: Parte inferior “Simulations in scenario X” después de la actualización de la plataforma.

En la tabla que existe en la parte inferior de esta página también se han realizado numerosas modificaciones, para adecuar más la información a los datos que se muestran en ella. También, se ha añadido una nueva columna en la tabla para determinar el tipo de vehículo con el que se realizó la simulación “Vehicle type”. Para poder implementar este cambio, se ha tenido que crear una nueva función en el fichero Partida.php, que devuelve a partir de una simulación el tipo de vehículo en formato texto. El código de esta nueva función getTipoVehiculoNombre() se puede ver en la figura 160.


```

/**
 * Método que devuelve el tipo de vehículo de una partida en formato texto
 * @return String tipo_vehiculo.
 */
public function getTipoVehiculoNombre() {

    switch ($this->tipo_vehiculo) {
        case 1:
            $vehiculo = 'Internal combustion engine passenger car';
            break;
        case 2:
            $vehiculo = 'Electric passenger car';
            break;
        case 3:
            $vehiculo = 'Internal combustion engine city bus';
            break;
        case 4:
            $vehiculo = 'Electric city bus';
            break;
        case 5:
            $vehiculo = 'Internal combustion engine truck';
            break;
        case 6:
            $vehiculo = 'Electric truck';
            break;
        default:
            $vehiculo = 'Internal combustion engine passenger car';
            break;
    }
    return $vehiculo;
}

```

Figura 160: Nueva función getTipoVehiculoNombre() dentro del fichero Partida.php.

Las modificaciones realizadas en los títulos de la tabla son las siguientes: “Simulation name” por “Scenario name”, “Average consumption” por “Average fuel consumption” y “Total consumption” por “Total fuel consumption”.

En los datos que incluye la tabla, también se han realizado modificaciones, como los títulos de los escenarios que se han pasado a inglés y se ha cambiado la palabra “Simulación” por “Scenario”. Como ya se ha comentado numerosas veces, esta modificación se ha tenido que hacer sobre la base de datos y afecta a toda la plataforma. También, se ha realizado la siguiente modificación en los títulos de los hipertextos de la tabla “See Partida” por “See Simulation”.

Por otro lado, al igual que en el caso de otros gráficos, se ha añadido un selector bajo el gráfico de barras. Este selector permite filtrar a los usuarios las simulaciones realizadas por el tipo de vehículo con el que se han realizado. Al incluir el selector de tipo de vehículo en la gráfica “Traffic offenses in last simulations in the scenario”, se ha tenido que modificar la función rjsim_partidas_simulacion_main_form() del fichero rjsimulador.forms.inc. En esta función se ha tenido que incluir el código de la figura 161 para implementar el selector.

```

// Apartado del formulario que hace referencia al select tipo de vehículo
$form['rjsim_fieldset_infracciones_simulacion']['select_vehiculo_infracciones_simulacion'] = array(
  '#type' => 'select',
  '#title' => t('Select vehicle type'),
  '#options' => array(
    1 => t('Internal combustion engine passenger car'),
    2 => t('Electric passenger car'),
    3 => t('Internal combustion engine city bus'),
    4 => t('Electric city bus'),
    5 => t('Internal combustion engine truck'),
    6 => t('Electric truck')
  ),
  '#default_value' => 1,
  //'#description' => t('Select the type of vehicle.'),
  '#ajax' => array(
    'callback' => 'ajax_infracciones_simulacion_vehiculo_callback',
    'wrapper' => 'div-chart-infracciones-simulacion',
  ),
);

```

Figura 161: Código incluido en la función `rjsim_partidas_simulacion_main_form()` dentro del fichero `rjsimulador.forms.inc` para generar el selector del gráfico “Traffic offenses in last simulations in the scenario”.

Como en todas las gráficas y gráficos de la plataforma, en su título se han realizado modificaciones en función del selector añadiendo el tipo de vehículo seleccionado por el usuario. Estos cambios se han llevado a cabo en la función `rjsim_partidas_simulacion_main_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en la figura 162 y en la figura 113 en la que se muestra una codificación similar.

```

// Recuperamos el tipo de vehículo del select tipo de vehículo
$vehiculo = isset($form_state['values']['select_vehiculo_infracciones_simulacion']) ? $form_state['values']['select_vehiculo_infracciones_simulacion'] : 1;

```

Figura 162: Código incluido en la función `rjsim_partidas_simulacion_main_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos del selector de vehículos.

Como en otras ocasiones, en función del valor que se recuperó en la variable `$vehiculo`, se genera el título para la gráfica, utilizando unas líneas de código similares a las de la figura 113.

Siguiendo el procedimiento que se siguió con otras gráficas, para que los cambios que se producen en el selector de vehículos se reflejen en el título del gráfico, se ha modificado la parte del formulario que hace referencia a él. Estas modificaciones se han realizado en la función `rjsim_partidas_simulacion_main_form()` dentro del fichero `rjsimulador.forms.inc`, como se muestra en la figura 163.

```

$form['rjsim_fieldset_infracciones_simulacion'] = array(
    '#type' => 'fieldset',
    '#title' => t( string: 'Traffic offenses in last simulations in the scenario'),
    '#collapsible' => FALSE,
    '#attributes' => array('class' => array('rjsim-margin'))
);
// Modificamos el apartado del formulario que hace referencia a la gráfica para que interactúe con el select tipo de vehículo
$form['rjsim_fieldset_infracciones_simulacion']['chart_infracciones_simulacion'] = array(
    '#type' => 'chart',
    '#chart_type' => 'column',
    '#stacking' => TRUE,
    '#chart_library' => 'highcharts',
    '#title' => t( string: 'Traffic offenses in last Simulations'.&#226;ttulgrafco),
    '#title_font_weight' => 'bold',
    '#prefix' => '<div id="div-chart-infracciones-simulacion">',
    '#suffix' => '</div>',
    'xaxis' =>
        array(
            '#type' => 'chart_xaxis',
            '#labels' => $labelAxisX
        ),
    'yaxis' =>
        array(
            '#type' => 'chart_yaxis',
            '#title' => 'Number of traffic offense'
        )
);
if(count($data)!=0){
    foreach ($tiposInfracciones as $id => $nombre_infraccion) {
        $form['rjsim_fieldset_infracciones_simulacion']['chart_infracciones_simulacion'][$id] =
            array(
                '#type' => 'chart_data',
                '#title' => t( string: '@nombreInfraccion', array('@nombreInfraccion' => $nombre_infraccion)),
                '#data' => $data[$id],
            );
    }
}else{
    foreach ($tiposInfracciones as $id => $nombre_infraccion) {
        $form['rjsim_fieldset_infracciones_simulacion']['chart_infracciones_simulacion'][$id] =
            array(
                '#type' => 'chart_data',
                '#title' => t( string: '@nombreInfraccion', array('@nombreInfraccion' => $nombre_infraccion)),
            );
    }
}
}

```

Figura 163: Código de la función `rjsim_partidas_simulacion_main_form()` dentro del fichero `rjsimulador.forms.inc` que genera la gráfica.

Se ha tenido que modificar la función `rjsim_partidas_simulacion_main_form()` dentro del fichero `rjsimulador.forms.inc` como se muestra en la figuras 164. Se puede apreciar que se recuperan los datos de las partidas del usuario en función del escenario y del tipo de vehículo que se eligió en el selector.

```

// Obtenemos la lista de partidas del usuario actual en función del id de un escenario y el tipo de vehículo
$listaPartidas = $gestorSimulaciones->getUsuarioActual()->retrieveAllPartidasByIdSimulacionVehiculo($id_simulacion, $vehiculo);
$listaPartidas->sortBy( sortField: 'Fecha', sort: 'ASC');

```

Figura 164: Código modificado en la función `rjsim_partidas_simulacion_main_form()` dentro del fichero `rjsimulador.forms.inc` para obtener los datos necesarios para el gráfico.

En la figura 164 se puede observar que se utiliza la función `retrieveAllPartidasByIdSimulacionVehiculo($idSimulation,$vehiculo)`, creada en el fichero `UsuarioSimulacion.php`, para recuperar los datos necesarios. De esta nueva función se habló en apartados anteriores y se mostró su código en la figura 36. Por otro lado, en la figura 163 se puede ver que los datos que se han recuperado después de tratarlos se utilizan para pintarlos en el gráfico.

Para que cuando el usuario cambie en el selector el valor del tipo de vehículo y este cambio tenga efecto sobre el gráfico y sus datos, ha sido necesario crear una nueva función AJAX. La llamada a esta función AJAX se puede ver en la figura 161. La función `ajax_infracciones_simulacion_vehiculo_callback()` recarga la parte del formulario que hace referencia a la gráfica actualizando los datos en función del vehículo seleccionado. El código de esta función se puede ver en la figura 165.

```
// Función Ajax que actualiza el valor del tipo de vehículo para la gráfica
function ajax_infracciones_vehiculo_callback($form, &$form_state) {
    return $form['rjsim_fieldset_infracciones_simulacion']['chart_infracciones_simulacion_1'];
}
```

Figura 165: Nueva función `ajax_infracciones_simulacion_vehiculo_callback()` dentro del fichero `rjsimulador.forms.inc`.

En este gráfico, en la parte lateral derecha se aprecia que se ha modificado el título de las infracciones traduciéndolas a inglés. Esta modificación, como se ha comentado en anteriores ocasiones, se ha realizado sobre la base de datos y afecta a toda la plataforma.

Cuando en la tabla de la parte inferior de la página un usuario pulsa en uno de los enlaces “See Simulation”, accederá a la página “Simulation Data”, en la que se representan los datos de la simulación que se quieren analizar.

4.5.3 Modificaciones de la página Simulation Data

En la página “Partida Data X” que ahora ha pasado a llamarse “Simulation Data” después de la actualización, se han realizado numerosas modificaciones. Esta página cuenta con varios elementos. El primero de ellos es una tabla resumen con los datos de la simulación que se está analizando. Se ha modificado el título de esta tabla, de la siguiente manera: “General data from game” por “General simulation data”.

The screenshot shows a web interface for 'User with 2: Partida Data'. At the top, there is a breadcrumb trail: 'Home » Administration » Analysis of Simulations » User with UID 2: Data about Simulations » User with UID 2: Partidas by Simulation 1'. Below this, the title of the page is 'General data from game'. A table displays simulation data with the following columns and values:

Average consumption	Total time	Average speed	Speed typical deviation	Average RPM	RPM typical deviation
8.37	630.89	33.65	20.27	2812.07	935.67

Figura 166: Primera tabla de la página “Partida Data” antes de la actualización de la plataforma.

Se han modificado los títulos de las columnas de la tabla para que se ajusten más a los datos que se muestran, las modificaciones realizadas son las siguientes: “Average consumption” por “Average fuel consumption”, “Speed typical deviation” por “Standard deviation of speed” y “RPM typical deviation” por “Standard deviation of RPM”.

En esta tabla, después de la actualización se ha añadido una nueva columna llamada “Vehicle type”, que permite a los usuarios saber con qué tipo de

vehículo se ha realizado la simulación. Para que en esta tabla se muestre el nombre del tipo de vehículo con el que se realizó la simulación, se ha tenido que utilizar la función `getTipoVehiculoNombre()` del fichero `Partida.php`. De esta función ya se habló en apartados anteriores y se puede ver su código en la figura 160. Estas modificaciones se han realizado sobre la función `generarTablaResumenDatosPartida(Partida $partida, ListaPartidas $listaPartidasSimulacion)` del fichero `rjsimulador.pages.inc`, como se muestra en las figuras 167 y 168.

```
function generarTablaResumenDatosPartida(Partida $partida, ListaPartidas $listaPartidasSimulacion) {
    // Recuperamos los datos de esta partida
    $vehiculo = $partida ->getTipoVehiculoNombre();
```

Figura 167: Código incluido en la función `generarTablaResumenDatosPartida(Partida $partida, ListaPartidas $listaPartidasSimulacion)` dentro del fichero `rjsimulador.pages.inc` para obtener el tipo de vehículo de una partida.

En la figura 167 se puede ver cómo se recupera el tipo de vehículo utilizado en una simulación y cómo estos datos recuperados son utilizados para informar sobre la nueva columna creada, como se muestra en la figura 168.

```
// Creamos la tabla con los datos resumen de la partida
$headers = array(
    array('data' => t( string: 'Vehicle type')),
    array('data' => t( string: 'Average fuel consumption')),
    array('data' => t( string: 'Total time')),
    array('data' => t( string: 'Average speed')),
    array('data' => t( string: 'Standard deviation of speed')),
    array('data' => t( string: 'Average RPM')),
    array('data' => t( string: 'Standard deviation of RPM'))
);
$rows =
    array(
        'data' =>
            array(
                array('data' => $vehiculo),
```

Figura 168: Código incluido en la función `generarTablaResumenDatosPartida(Partida $partida, ListaPartidas $listaPartidasSimulacion)` dentro del fichero `rjsimulador.pages.inc` para crear la nueva columna con los datos obtenidos.

Otra nueva característica de esta tabla es que, en función del tipo de vehículo utilizado en la simulación, se verán unos datos o no. Cuando se está analizando una simulación que se realizó con un vehículo eléctrico, hay datos que no son relevantes porque se refieren a vehículos de combustión. Estas modificaciones se incluyen en la función `generarTablaResumenDatosPartida(Partida $partida, ListaPartidas $listaPartidasSimulacion)` dentro del fichero `rjsimulador.pages.inc` y se pueden apreciar en las figuras 169 y 170.

```
// Si el tipo de vehículo es de combustión aparecen en la tabla apartados relacionados con la relación de marchas del vehículo
if ($partida->getTipoVehiculo()==1 || $partida->getTipoVehiculo()==3 || $partida->getTipoVehiculo()==5){
// Creamos la tabla con los datos resumen de la partida
$headers = array(
    array('data' => t( string: 'Vehicle type')),
    array('data' => t( string: 'Average fuel consumption')),
    array('data' => t( string: 'Total time')),
    array('data' => t( string: 'Average speed')),
    array('data' => t( string: 'Standard deviation of speed')),
    array('data' => t( string: 'Average RPM')),
    array('data' => t( string: 'Standard deviation of RPM'))
);
```

Figura 169: Código incluido en la función generarTablaResumenDatosPartida(Partida \$partida, ListaPartidas \$listaPartidasSimulacion) dentro del fichero rjsimulador.pages.inc para vehículos de combustión.

En la figura 169 se puede ver cómo se genera un tipo de tabla para los vehículos con motor de combustión. En cambio, en la figura 170 se observa cómo se genera un tipo de tabla distinto para los vehículos de motor eléctrico.

```
// Si el tipo de vehículo no es de combustión no aparecen en la tabla apartados relacionados con la relación de marchas del vehículo
else {
// Creamos la tabla con los datos resumen de la partida
$headers = [
    ['data' => t( string: 'Vehicle type')],
    ['data' => t( string: 'Total time')],
    ['data' => t( string: 'Average speed')],
    ['data' => t( string: 'Standard deviation of speed')],
];
```

Figura 170: Código incluido en la función generarTablaResumenDatosPartida(Partida \$partida, ListaPartidas \$listaPartidasSimulacion) dentro del fichero rjsimulador.pages.inc para vehículos eléctricos.

Todas estas modificaciones visuales de la tabla “General Simulation data” de las que se ha estado hablando, se pueden valorar si se realiza una comparación entre las figuras 166 y 171.

Vehicle type	Average fuel consumption	Total time	Average speed	Standard deviation of speed	Average RPM	Standard deviation of RPM
Internal combustion engine passenger car	8.37	630.89	33.65	20.27	2812.07	935.67

Figura 171: Primera tabla de la página “Simulation Data” después de la actualización de la plataforma.

Debajo de la primera tabla existe otra tabla que también ha sufrido modificaciones con esta actualización de la plataforma. En esta tabla se ha modificado el título de la siguiente manera “Average RPMs by gear” por “Average RPMs per gear”.

Gear 1	Gear 2	Gear 3	Gear 4	Gear 5
1365.61	3080.2	3132.15	2887.74	2979.83

Figura 172: Segunda tabla de la página Partida Data antes de la actualización de la plataforma.

Estas modificaciones referentes al título de la tabla “Average RPMs per gear” se pueden analizar si se realiza una comparación entre las figuras 172 y 174. Una de las características en la actualización es que cuando se está analizando una simulación que se ha realizado con vehículo con motor eléctrico, la tabla “Average RPMs per gear” no aparece en la pantalla, ya que en esta tabla se muestra las marchas de un motor de combustión, que no tienen sentido en el caso de un motor eléctrico. Para poder realizar esta modificación en la página, se ha tenido que modificar `rjsimulador_partida_page($id_partida)` dentro del fichero `rjsimulador.pages.inc`, para que en función del tipo vehículo si es de motor de combustión se muestre la información relacionada con las marchas y si es un vehículo de motor eléctrico no, como se muestra en la figura 173.

```
// Creamos la tabla con los datos de RPMs por marcha en función del tipo de vehículo
if ($miPartida->getTipoVehiculo()==1 || $miPartida->getTipoVehiculo()==3 || $miPartida->getTipoVehiculo()==5){
    $renderArrayTableRpmMarchas = generarTablaRpmMarchas($miPartida);
}
else{
    $renderArrayTableRpmMarchas=null;
}
```

Figura 173: Código incluido en la función `rjsimulador_partida_page($id_partida)` dentro del fichero `rjsimulador.pages.inc` para generar la tabla “Average RPMs per gear”.

Average RPMs per gear				
Gear 1	Gear 2	Gear 3	Gear 4	Gear 5
1365.61	3080.2	3132.15	2887.74	2979.83

Figura 174: Segunda tabla de la página “Simulation Data” después de la actualización de la plataforma.

La siguiente parte de la página que se ha modificado es la imagen en la que se muestran los puntos por los que ha pasado el usuario al hacer el recorrido de la simulación. En esta imagen una de las modificaciones que se ha realizado y que salta a simple vista al usuario, es que antes de la simulación las etiquetas que marcaban los puntos del recorrido donde se han tomado datos eran muy grandes y tapaban el recorrido, por lo el usuario no era capaz de ver el recorrido real que había realizado en la simulación, como se puede apreciar en la figura 175.

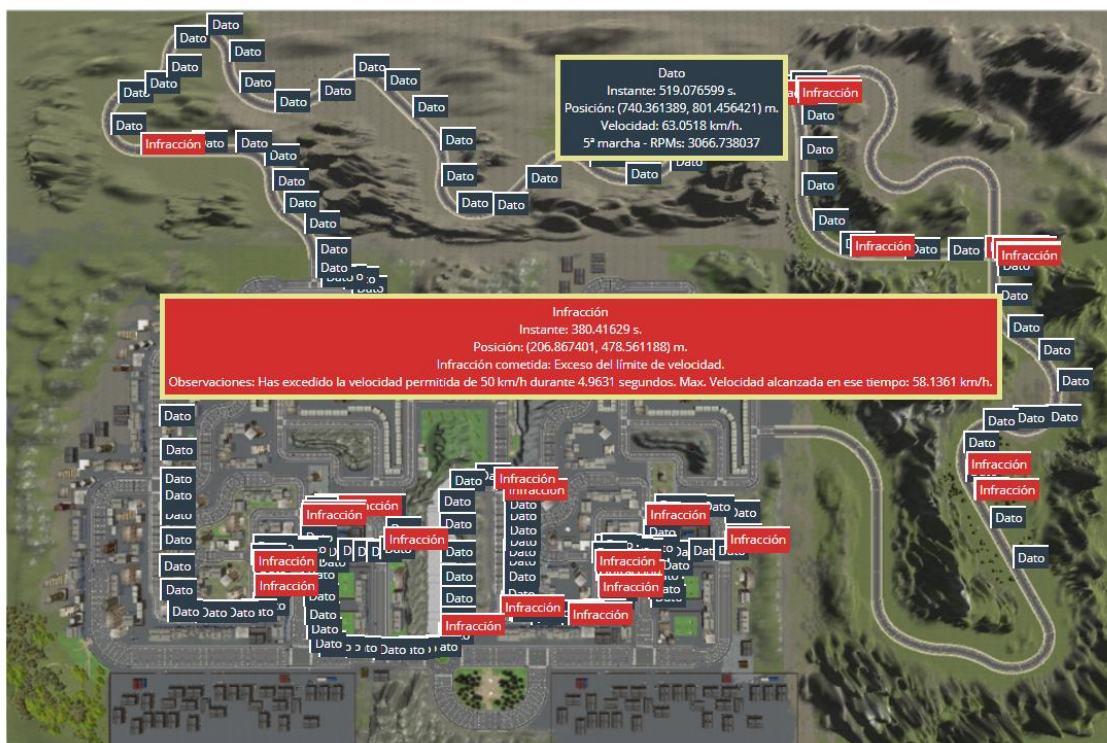


Figura 175: Imagen que muestra el recorrido del usuario en la simulación de la página Partida Data antes de la actualización de la plataforma.

Para solucionar este problema, se optó por eliminar los títulos de las etiquetas y dejar simplemente a la vista del usuario los puntos donde se recogieron los datos, diferenciando el tipo de dato entre datos como tal, siendo estos puntos azules, las infracciones de seguridad, que son puntos rojos, y las infracciones de eficiencia, que son puntos grises, en la figura 185 se puede ver esta implementación. Para poder hacer esta diferenciación entre los tipos de infracciones de seguridad y de eficiencia, se ha tenido que crear una nueva función, en el fichero `Infraccion.php`, llamada `getTipoInfraccion()`. El código de esta nueva función se puede ver en la figura 176.

```

/**
 * Método que devuelve el tipo de infracción
 * @return string
 */
public function getTipoInfraccion() {
    if (!isset($this->nombre_infraccion)) {
        $this->setNombreInfraccion(Constants::getNombreInfraccion($this->getIdInfraccion()));
    }
    switch ($this->nombre_infraccion)
    {
        case "Sudden breaking (not using engine braking)":
            $titulotipo = "Eficiencia";
            break;
        case "Not progressive acceleration (from 0 to 50km/h in less than 5 seconds)":
            $titulotipo = "Eficiencia";
            break;
        case "Sudden acceleration":
            $titulotipo = "Eficiencia";
            break;
        default:
            $titulotipo = "Seguridad";
            break;
    }
    return $titulotipo;
}

```

Figura 176: Nueva función `getTipoInfraccion()` dentro del fichero `Infraccion.php`.

Para poder pintar las infracciones diferenciándolas por el color en función del tipo, se ha tenido que modificar el código de la función `renderableArray(array $arrayDatosSimulacion)` en el archivo `Infraccion.php`. Para poder realizar esta diferenciación de infracciones ha sido necesario utilizar la función `getTipoInfraccion()` de la que se habló anteriormente y que se muestra en la figura 176. Se puede ver como se han llevado a cabo las modificaciones en la figura 177.

```

public function renderableArray(array $arrayDatosSimulacion) {
    $provider = FactoryDataManager::createDataProvider();
    $partidaUsu = Partida::loadById($SESSION['partida']);
    $usuario = $provider->loadSimulatorUser($partidaUsu->getUserUid());

    $left = round( $val: ($this->getPosicionX() + $arrayDatosSimulacion['offset_anchura']) * $arrayDatosSimulacion['proporcion_anchura'], precision: 0);
    $bottom = round( $val: ($this->getPosicionZ() + $arrayDatosSimulacion['offset_altura']) * $arrayDatosSimulacion['proporcion_altura'], precision: 0);

    $value = '<div class="breve"></div>';
    $value .= '<div class="completo oculto">';
    $value .= 'User: ' . $usuario->getName() . '<br>';
    $value .= 'Simulation date: ' . $partidaUsu->getFechaAsObject()->format( format: "Y-m-d H:i:s") . '<br>';
    $value .= 'Traffic offence: ' . $this->getNombreInfraccion() . '<br>';
    $value .= 'Vehicle type: ' . $partidaUsu->getTipoVehiculoNombre() . '<br><br>';
    $value .= 'Type traffic offence: ' . $this->getTipoInfraccion() . '<br>';
    $value .= 'Instant: ' . $this->getInstante() . ' s.<br>';
    $value .= 'Position: ( ' . $this->getPosicionX() . ' , ' . $this->getPosicionZ() . ' ) m.<br>';

    if (strlen($this->getObservaciones()) > 0) {
        $value .= 'Observations: ' . $this->getObservaciones();
    }
    $value .= '</div>';
    switch ($this->getTipoInfraccion()){
        case "Efficiency":
            $renderArray = array(
                '#type' => 'html_tag',
                '#tag' => 'div',
                '#attributes' => array(
                    'id' => array('infraccion-' . str_replace( search: '.', replace: '-', $this->getInstante())),
                    'class' => array('infraccion-tag'),
                    'style' => array('position: absolute;', 'bottom: ' . $bottom . 'px;', 'left: ' . $left . 'px;', 'background-color: #9B9B99;'),
                    '#value' => $value,
                );
            break;
        case "Security":
            $renderArray = array(
                '#type' => 'html_tag',
                '#tag' => 'div',
                '#attributes' => array(
                    'id' => array('infraccion-' . str_replace( search: '.', replace: '-', $this->getInstante())),
                    'class' => array('infraccion-tag'),
                    'style' => array('position: absolute;', 'bottom: ' . $bottom . 'px;', 'left: ' . $left . 'px;'),
                    '#value' => $value,
                );
            break;
    }
    return $renderArray;
}

```

Figura 177: Código de la función `renderableArray()` del archivo `Infraccion.php` para separar por colores el tipo de infracción.

Por otro lado, se han modificado los datos que se muestran cuando el usuario hace clic en las etiquetas que actualmente en la imagen del recorrido son simplemente puntos. Diferenciando entre los propios datos y las infracciones, se empezará analizando los puntos en que están marcados los datos como tal. Las etiquetas que aparecen se han modificado de la siguiente manera añadiendo los campos “User”, “Simulation date” y “Vehicle type” para poder aportar mayor información, también se han modificado las ya existentes de la siguiente manera: “Dato” por “Data”, “Instante” por “Instant”, “Posición” por “Position”, “Velocidad” por “Speed” y “Marcha” por “Gear”.

En el caso de las infracciones, las etiquetas también se han modificado de la siguiente manera, añadiendo los campos “User”, “Simulation date”, “Type traffic offence” y “Vehicle type” para poder aportar mayor información. También, se han modificado las ya existentes de la siguiente manera:

“Infracción” por “Traffic offence”, “Instante” por “Instant”, “Posición” por “Position” y “Observaciones” por “Observations”.

Con la actualización, se ha añadido un selector en la parte superior de la imagen, en la que se pinta el recorrido del usuario, con el nombre “Select the data to show in the image”. Este selector ofrece la posibilidad de seleccionar los datos que se desean mostrar en la imagen, ya sea todos los datos (“All data”), solo datos del recorrido (“Data only”), datos sobre infracciones de seguridad (“Traffic offenses: Security”) y datos sobre las infracciones de eficiencia (“Traffic offenses: Efficiency”). Para implementar este selector ha sido necesario crear la función `selectDataRecorrido()` en el fichero `rjsimulador.pages.inc`. En el código se puede apreciar cómo se ha generado el selector del que se ha estado hablando, el contenido de esta nueva función se muestra en la figura 178.

```
function selectDataRecorrido ($form,$form_state) {

    $_SESSION['dato'] = isset($form_state['values']['data_data_select_data']) ? $form_state['values']['data_data_select_data'] : 1;

    $form['data_data_container'] = [
        '#type' => 'fieldset',
        '#title' => t( string: 'Select the data to show in the image'),
        '#collapsible' => FALSE,
        '#attributes' => ['class' => ['rjsim-margin']]
    ];

    //Selector de tipo de dato a mostrar en el recorrido
    $form['data_data_container']['data_data_select_data'] = array(
        '#type' => 'select',
        '#title' => t( string: 'Select the data to show'),
        '#options' =>array(
            1 => t( string: 'All data'),
            2 => t( string: 'Data only'),
            3 => t( string: 'Traffic offenses: Security'),
            4 => t( string: 'Traffic offenses: Efficiency'),
        ),
        '#default_value' => 1,
        '#description' => t( string: 'Select the data you want to show in the image.'),
        '#ajax' => array(
            'callback' => 'ajax_recorrido_datos',
            'wrapper' => 'div-imagen-recorrido'),
    );

    return $form;
}
```

Figura 178: Nueva función `selectDataRecorrido()` dentro del fichero `rjsimulador.pages.inc`.

Si se observa el código de esta función se puede ver que se hace una llamada a una función AJAX. Esta función ha sido creada y es la encargada de recargar la imagen con los datos deseados cuando se cambia el valor del selector. La función `ajax_recorrido_datos()` se encuentra en el fichero `rjsimulador.pages.inc`. En esta función se realiza un filtrado de datos en función de la elección del usuario en el selector, este filtrado de datos se puede ver en la figura 179.

```

//Funcion para comparar recorridos en la imagen
function ajax_recorrido_datos($form, &$form_state) {

    $elementsDataAndInfraction [] = 0;
    $miPartida = Partida::loadById($SESSION['partida']);
    $simulacion = new \Drupal\rjsimulador\Simulacion($miPartida->getIdSimulacion(), $miPartida->getUserUid());

    if( $SESSION['dato']==1){
        //Obtenemos los datos de la simulacion actual
        foreach ($miPartida->getListaDatos() as $dato) {
            $elementsDataAndInfraction[] = $dato->renderableArray($simulacion->getArrayDatosImagen());
        }

        foreach ($miPartida->getListaInfracciones() as $elemento_lista) {
            $elementsDataAndInfraction[] = $elemento_lista->renderableArray($simulacion->getArrayDatosImagen());
        }
    }
    else if ( $SESSION['dato']==2){
        //Obtenemos los datos de la simulacion actual
        foreach ($miPartida->getListaDatos() as $dato) {
            $elementsDataAndInfraction[] = $dato->renderableArray($simulacion->getArrayDatosImagen());
        }
    }
    else if ( $SESSION['dato']==3) {
        //Obtenemos los datos de la simulacion actual
        foreach ($miPartida->getListaInfracciones() as $elemento_lista) {
            if($elemento_lista->getTipoInfraccion()=="Security") {
                $elementsDataAndInfraction[] = $elemento_lista->renderableArray($simulacion->getArrayDatosImagen());
            }
        }
    }
    else if ( $SESSION['dato']==4) {
        //Obtenemos los datos de la simulacion actual
        foreach ($miPartida->getListaInfracciones() as $elemento_lista) {
            if($elemento_lista->getTipoInfraccion()=="Efficiency") {
                $elementsDataAndInfraction[] = $elemento_lista->renderableArray($simulacion->getArrayDatosImagen());
            }
        }
    }
}

$escenario= 'div-imagen'.$simulacion->getIdSimulacion();
$renderImagen = [
    '#type' => 'container',
    '#prefix' => '<div id="div-imagen-recorrido">',
    '#suffix' => '</div>',
    '#attributes' => ['class' => [$escenario]],
    '#etiquetas' => $elementsDataAndInfraction,
];

return $renderImagen;
}

```

Figura 179: Nueva función ajax_recorrido_datos() dentro del fichero rjsimulador.pages.inc.

En la parte inferior de la imagen, se ha creado un nuevo selector que permite al usuario comparar el recorrido con el de otro usuario que haya utilizado el mismo tipo de vehículo y que haya realizado la simulación en el mismo escenario. El selector “Compare the simulation with the simulation of another user” muestra las opciones que existen para esa comparación. Este selector se puede ver en la parte inferior de la figura 185. Para poder generar el contenido del selector con un formato concreto, se ha desarrollado una función en el fichero GestorSimulaciones.php. La función getArrayUsuariosRecorrido() es la encargada de dar el siguiente formato a los datos “nombre de usuario ---- Simulation: fechas en la que se realizó la simulación”. Esto se consigue formateando los datos de cada usuario y de las simulaciones que ha realizado en ese escenario con el mismo tipo de vehículo que el usuario original. El código que implementa este comportamiento en esta función se puede ver en la figura 180.

```

/**
 * Devuelve un array de todos los usuarios para comparar el recorrido.
 *
 * El array viene de la forma uid partida => informacion.
 *
 * @return array Todos los usuarios de la forma uid partida=>informacion.
 */
public function getArrayUsuariosRecorrido() {
    $arrayPartidaUidInfo [0]="No user";
    $gestorSimulaciones = new GestorSimulaciones();

    $miPartida = Partida::loadById($_SESSION['partida']);
    $tipovehiculo = $miPartida->getTipoVehiculo();
    $idSimulacion = $miPartida->getIdSimulacion();

    //Recuperando datos para todos los usuarios
    $dataRetriever = new ListaUsuariosDataRetriever($gestorSimulaciones->getListaTodosUsuarios());
    $listaUsu = $dataRetriever->getListaUsuarios();

    foreach ($listaUsu as $usuario) {
        if ($miPartida->getUserUid() != $usuario->getUserUid()) {
            $listaPartidasUsu = $usuario->retrieveAllPartidasByIdSimulacionVehiculo($idSimulacion, $tipovehiculo);
            foreach ($listaPartidasUsu as $partidaUsu) {
                if ($partidaUsu != NULL) {
                    $arrayPartidaUidInfo[$partidaUsu->getIdPartida()] = $usuario->getName() . " ---- Simulation: " . $partidaUsu->getFechaAsObject()
                    ->format( format: "Y-m-d H:i:s");
                }
            }
        }
    }
    return $arrayPartidaUidInfo;
}

```

Figura 180: Nueva función selectUserCompararRecorrido() dentro del fichero rjsimulador.pages.inc.

Para poder implementar este selector, se ha tenido que crear la función selectUserCompararRecorrido() en el fichero rjsimulador.pages.inc, ha sido necesario apoyarse en la función getArrayUsuariosRecorrido() de la que se ha estado hablando anteriormente y que se muestra en la figura 180 para obtener los datos para este selector. En la figura 181 se puede ver el código de esta función.

```

function selectUserCompararRecorrido($form,$form_state) {

    $gestorSimulaciones = new GestorSimulaciones();
    $miPartida = Partida::loadById($_SESSION['partida']);

    $provider = FactoryDataManager::createDataProvider();
    // Cargamos el usuario de la partida
    $usuario = $provider->loadSimulatorUser($miPartida->getUserUid());
    //Asignamos el uid del usuario seleccionado a variable de sesion usuario
    $_SESSION['partidaComparar'] = isset($_form_state['values']['data_user_select_user']) ? $_form_state['values']['data_user_select_user'] : $usuario->getUserUid();

    $form['data_user_container'] = [
        '#type' => 'fieldset',
        '#title' => t( string: 'Compare the simulation with the simulation of another user'),
        '#collapsible' => FALSE,
        '#attributes' => ['class' => ['rjsim-margin']]
    ];

    //Selector de usuarios para comparar recorridos
    $form['data_user_container']['data_user_select_user'] = array(
        '#type' => 'select',
        '#title' => t( string: 'Select a user and a simulation'),
        '#options' => $gestorSimulaciones->getArrayUsuariosRecorrido(),
        '#default_value' => '-',
        '#description' => t( string: 'Select a user and a simulation to compare with the current simulation.'),
        '#ajax' => array(
            'callback' => 'ajax_recorrido_usuario',
            'wrapper' => 'div-imagen-recorrido',
        )
    );
    return $form;
}

```

Figura 181: Nueva función selectUserCompararRecorrido() dentro del fichero rjsimulador.pages.inc.

Observando el código de esta función se ve que realiza una llamada a una función AJAX. Esta función ha sido creada y es la encargada de recargar la imagen con los datos deseados cuando se cambia el valor del selector. La

función `ajax_recorrido_usuario()` se encuentra en el fichero `rjsimulador.pages.inc`. El código de esta función es muy extenso por lo que no se ha creído conveniente incluirlo dentro del documento con una captura. Para no llevar a confusiones, se ha optado por cambiar el color de las etiquetas del usuario cuyo recorrido se compara con el recorrido actual. =Para ello, ha sido necesario crear dos nuevas funciones, una para generar las etiquetas de los datos y otra para las etiquetas de las infracciones del usuario a comparar.

Para generar las etiquetas de los datos del usuario a comparar se ha creado la función `renderableArrayCompara()` dentro del fichero `DatoInstantaneo.php`. Esta función genera las etiquetas de los datos del usuario a comparar de color verde con el formato del usuario original. En esta función se han realizado modificaciones en el estilo de las etiquetas que permiten diferenciar las del usuario original con las del usuario a comparar. Estas modificaciones se pueden ver en la figura 182.

```

public function renderableArrayCompara(array $arrayDatosSimulacion) {
    $provider = FactoryDataManager::createDataProvider();
    $partidaUsu = Partida::loadById($_SESSION['partidaComparar']);
    $usuario = $provider->loadSimulatorUser($partidaUsu->getUserUid());

    $left = round( $val: ($this->getPosicionX()+1 + $arrayDatosSimulacion['offset_anchura']) * $arrayDatosSimulacion['proporcion_anchura'], precision: 0);
    $bottom = round( $val: ($this->getPosicionZ() + $arrayDatosSimulacion['offset_altura']) * $arrayDatosSimulacion['proporcion_altura'], precision: 0);
    $left = $left+10;
    $bottom= $bottom+5;

    $value = '<div class="breve"></div>';
    $value .= '<div class="completo oculto">';
    $value .= 'User: '.$usuario->getName().<br>';
    $value .= 'Simulation date: '.$partidaUsu->getFechaAsObject()->format( format: "Y-m-d H:i:s").<br>';
    $value .= 'Vehicle type: '.$partidaUsu->getTipoVehiculoNombre().<br><br>';
    $value .= 'Data<br>';
    $value .= 'Instant: ' . $this->getInstante() . ' s.<br>';
    $value .= 'Position: ( ' . $this->getPosicionX() . ' , ' . $this->getPosicionZ() . ' ) m.<br>';
    $value .= 'Speed: ' . $this->getVelocidad() . ' km/h.<br>';
    $value .= $this->getMarcha() . " gear - RPMs: " . $this->getRpm() . ' ';
    $value .= '</div>';

    $renderArray = array(
        '#type' => 'html_tag',
        '#tag' => 'div',
        '#attributes' => array(
            'id' => array('dato-' . str_replace( search: '.', replace: '-', $this->getInstante())),
            'class' => array('dato-tag'),
            'style' => array('position:absolute;', 'bottom: ' . $bottom . 'px;', 'left: ' . $left . 'px;', 'background-color:#029020;'),
            '#value' => $value,
        );
    );

    return $renderArray;
}

```

Figura 182: Nueva función `renderableArrayCompara()` dentro del fichero `DatoInstantaneo.php`.

Por otro lado, para poder generar las etiquetas de las infracciones del usuario a comparar, se ha creado la función `renderableArrayCompara()` dentro del fichero `Infraccion.php`. Esta función genera las etiquetas de las infracciones de seguridad y de eficiencia del usuario a comparar de un color distinto a las del usuario original, pero con el mismo formato. La función crea las etiquetas de las infracciones de seguridad del usuario a comparar en color naranja y las etiquetas de las infracciones de eficiencia en color amarillo, en la imagen del recorrido en figura 185 se puede ver esta implementación. En esta función se han realizado modificaciones en el estilo de las etiquetas para poder diferenciar las infracciones del usuario original de las del usuario a comparar, en figura 183 se puede ver el código de esta función.

```

public function renderableArrayCompara(array $arrayDatosSimulacion) {
    $provider = FactoryDataManager::createDataProvider();
    $partidaUsu = Partida::loadById($_SESSION['partidaComparar']);
    $usuario = $provider->loadSimulatorUser($partidaUsu->getUserUid());
    $left = round( $this->getPosicionX() + $arrayDatosSimulacion['offset_anchura'], precision: 0);
    $bottom = round( $this->getPosicionZ() + $arrayDatosSimulacion['offset_altura'], precision: 0);
    $left = $left-10;
    $bottom= $bottom+5;
    $value = '<div class="breve"></div>';
    $value .= '<div class="completo oculto">';
    $value .= 'User: ' . $usuario->getName() . '<br>';
    $value .= 'Simulation date: ' . $partidaUsu->getFechaAsObject()->format( format: "Y-m-d H:i:s") . '<br>';
    $value .= 'Vehicle type: ' . $partidaUsu->getTipoVehiculoNombre() . '<br><br>';
    $value .= 'Traffic offence: ' . $this->getNombreInfraccion() . '<br>';
    $value .= 'Type traffic offence: ' . $this->getTipoInfraccion() . '<br>';
    $value .= 'Instant: ' . $this->getInstante() . ' s.<br>';
    $value .= 'Position: (' . $this->getPosicionX() . ', ' . $this->getPosicionZ() . ') m.<br>';

    if (strlen($this->getObservaciones()) > 0) {
        $value .= 'Observations: ' . $this->getObservaciones();
    }
    $value .= '</div>';
    switch ($this->getTipoInfraccion()){
        case "Efficiency":
            $renderArray = array(
                '#type' => 'html_tag',
                '#tag' => 'div',
                '#attributes' => array(
                    'id' => array('infraccion-' . str_replace( search: '.', replace: '-', $this->getInstante())),
                    'class' => array('infraccion-tag'),
                    'style' => array('position:absolute;', 'bottom: ' . $bottom . 'px;', 'left: ' . $left . 'px;', 'background-color:#000002;'),
                ),
                '#value' => $value,
            );
            break;
        case "Security":
            $renderArray = array(
                '#type' => 'html_tag',
                '#tag' => 'div',
                '#attributes' => array(
                    'id' => array('infraccion-' . str_replace( search: '.', replace: '-', $this->getInstante())),
                    'class' => array('infraccion-tag'),
                    'style' => array('position:absolute;', 'bottom: ' . $bottom . 'px;', 'left: ' . $left . 'px;', 'background-color:#B25B03;'),
                ),
                '#value' => $value,
            );
            break;
    }
    return $renderArray;
}

```

Figura 183: Nueva función renderableArrayCompara() dentro del fichero Infraccion.php.

En la figura 183 se puede ver que se utiliza la función `getTipoInfraccion()`, de la que se habló con anterioridad, y de la que se puede ver su codificación en la figura 176. Esta función ha sido necesaria para poder implementar la función `renderableArrayCompara()`, que genera las etiquetas para las infracciones del usuario a comparar.

También se creyó conveniente añadir un título a la imagen en la que se comparan los recorridos, para facilitar a los usuarios el análisis de los datos que están viendo en cada momento. Para ello, ha sido necesario crear la función `renderableArrayTitulo()` en el fichero `DatoInstantaneo.php`. El código de esta función se puede ver en la figura 184.

```

/**
 * @inheritdoc
 */
public function renderableArrayTitulo(array $arrayDatosSimulacion) {
    $provider = FactoryDataManager::createDataProvider();
    $partidaUsu = Partida::loadById($SESSION['partida']);
    $usuario = $provider->loadSimulatorUser($partidaUsu->getUserUid());
    $partidaUsuCom = Partida::loadById($SESSION['partidaComparar']);
    $usuarioCom = $provider->loadSimulatorUser($partidaUsuCom->getUserUid());
    $left = 10;
    $bottom = -165;
    $value = '<div class="breve" style="font-size: 18px;color:#000000;">'. 'Comparison of the current simulation: <br> - User:
    '.$usuario->getName().' - Date: '.$partidaUsu->getFechaAsObject()->format('format: "Y-m-d H:i:s"').' - Rectangles:
    (Data: blue, Traffic offenses-Security: red, Traffic offenses-Efficiency: gray) <br> with the simulation: <br> - User:
    '.$usuarioCom->getName().' - Date: '.$partidaUsuCom->getFechaAsObject()->format('format: "Y-m-d H:i:s"').' - Rectangles:
    (Data: green, Traffic offenses-Security: orange, Traffic offenses-Efficiency: yellow)</div>';

    $value .= '<div class="completo oculo">';
    $value .= '<br>'.$partidaUsu->getNombreSimulacion(). '<br>';
    $value .= 'Vehicle type: '.$partidaUsu->getTipoVehiculoNombre(). '<br><br>';
    $value .= '- Current simulation:<br>';
    $value .= 'User: '.$usuario->getName(). '<br>';
    $value .= 'Simulation date: '.$partidaUsu->getFechaAsObject()->format('format: "Y-m-d H:i:s"').'<br><br>';
    $value .= '- Simulation to compare:<br>';
    $value .= 'User: '.$usuarioCom->getName(). '<br>';
    $value .= 'Simulation date: '.$partidaUsuCom->getFechaAsObject()->format('format: "Y-m-d H:i:s"').'<br><br>';
    $value .= '</div>';

    $renderArray = array(
        '#type' => 'html_tag',
        '#tag' => 'div',
        '#attributes' => array(
            'id' => array('dato-' . str_replace(' ', '-', $this->getInstante()),
            'class' => array('dato-tag'),
            'style' => array('position:absolute;', 'bottom: ' . $bottom . 'px;', 'left: ' . $left . 'px;', 'background-color:#ffffff;')),
        '#value' => $value,
    );

    return $renderArray;
}

```

Figura 184: Nueva función renderableArrayTitulo() dentro del fichero DatoInstantaneo.php.

Todas las modificaciones de las que se ha hablado anteriormente y que se han realizado sobre la imagen en la que se muestran los recorridos de los usuarios, se pueden apreciar si se realiza una comparación entre las figuras 175 y 185.

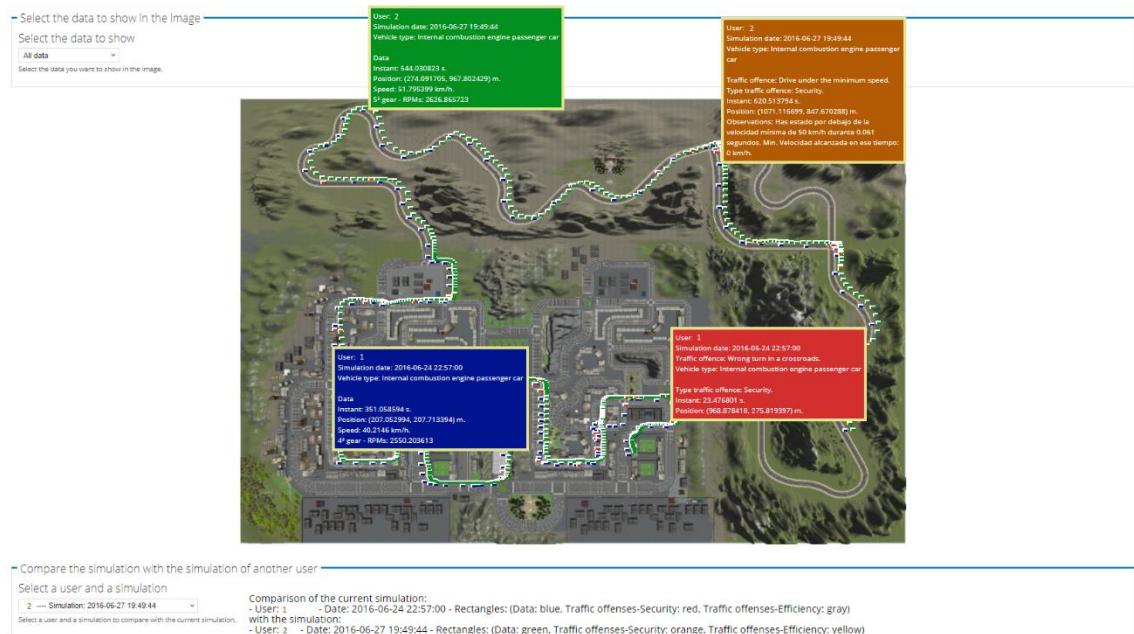


Figura 185: Imagen que muestra el recorrido del usuario en la simulación de la página “Simulation Data” después de la actualización de la plataforma.

En la parte final de la página existen dos tablas, una que muestra los instantes en los que se han tomado los datos del recorrido y otra que muestra la

información de las infracciones y los instantes en los que se han cometido. En la figura 186 se puede ver cómo eran estas tablas antes de la actualización de la plataforma.

La única de estas tablas que se ha modificado en esta actualización es la que muestra la información sobre las infracciones cometidas en el recorrido.

Data Number	Instant	Infraction number	Instant
1	0.0785	1	23.020399
2	3.0635	2	23.476801
3	5.9987	3	57.171398
4	9.0564	4	78.8937
5	12.0653	5	82.583099
6	15.0485	6	94.358803
7	18.0187	7	105.557701
8	21.077801	8	113.638208

[Go Back](#)

Figura 186: Tablas de la parte final de la página Partida Data antes de la actualización de la plataforma.

En la tabla de infracciones se han añadido dos columnas nuevas como se puede observar en la figura 187, una de ellas que se titula “Name”, en la que se muestra el nombre de la infracción cometida y otra de ellas que se titula “Type”, en la que se muestra el tipo de infracción cometida. Para ello, se utiliza la función getTipoInfraccion() de la que se habló antes. Su código se puede ver en la figura 176.

Data Number	Instant	Traffic offenses number	Name	Type	Instant
1	0.0785	1	Failing to stop at a red traffic light	Security	23.020399
2	3.0635	2	Wrong turn in a crossroads	Security	23.476801
3	5.9987	3	Wrong turn in a crossroads	Security	57.171398
4	9.0564	4	Not switch on the corresponding turn light in a turn	Security	78.8937
5	12.0653	5	Exceed the speed limit	Security	82.583099
6	15.0485	6	Exceed the speed limit	Security	94.358803
7	18.0187	7	Exceed the speed limit	Security	105.557701
8	21.077801	8	Exceed the speed limit	Security	113.638208

[Go Back](#)

Figura 187: Tablas de la parte final de la página “Simulation Data” después de la actualización de la plataforma.

Por otro lado, se ha modificado uno de los títulos de la tabla de infracciones de la siguiente manera: “Infraction number” por “Traffic offenses number”. Las modificaciones que se han realizado sobre esta tabla y de las que se han estado comentando, se pueden apreciar si se observan y comparan las figuras 186 y 187 (R.J.Corchero, 2016).

5. Manual de usuario de la plataforma web

A continuación, se van a analizar las funcionalidades que aportan las nuevas características que se han desarrollado en la actualización de la plataforma para los distintos usuarios. Teniendo en cuenta eso, vamos a separar el análisis de estas nuevas características de la aplicación y la explicación de su funcionamiento en dos apartados, uno con la información para el administrador y otro con la información para los usuarios comunes de la plataforma.

5.1 Manual de usuario de la plataforma para el administrador

Se comenzará analizando desde el momento en el que el usuario se valida como administrador de la plataforma web. Cuando sucede esto, lo primero que ve el usuario en la plataforma es la página principal, como se muestra en la figura 188.

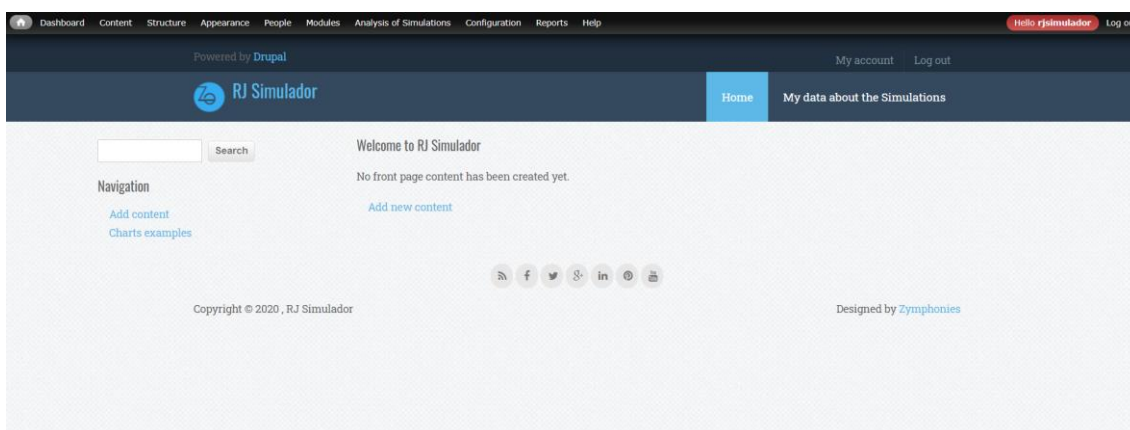


Figura 188: Página principal de la plataforma web cuando se logea un usuario como administrador.

La característica principal de esta página, a la que solo tiene acceso el administrador de la plataforma web, es el menú que se sitúa en la parte superior de la misma. Este menú aporta características funcionales muy importantes para el mantenimiento de la página por parte del administrador. Aunque este menú dispone de muchas opciones, solo se van a analizar las pestañas en las que se han realizado modificaciones en esta actualización. En la página principal también existe un apartado llamado “My data about Simulation” en la que el administrador puede analizar los datos de sus simulaciones como un usuario normal. Como este apartado trata los datos del administrador como los de un usuario normal, se tratará en la sección siguiente de la memoria llamado “Manual de usuario de la plataforma para un usuario común”.

La pestaña del menú principal que se ha visto afectada por la actualización de la plataforma web es “Analysis of simulations”. Si el administrador sitúa el ratón

sobre esta pestaña, se abre un desplegable con otras dos pestañas más: “Analysis of vehicle data” y “Analysis of traffic offenses”.

De estas tres pestañas, vamos a analizar primero “Analysis of vehicle data”. En esta pestaña nos vamos a centrar en el apartado “Simulation data”. Esta página permite analizar y comparar los datos de todos los usuarios de la plataforma a la vez, permitiendo seleccionar un escenario en el que se han realizado las simulaciones en el selector “Select scenario”, el tipo de dato que se va a analizar en el selector “Select data to show” y el tipo de vehículo con el que los usuarios realizaron las simulaciones en el selector “Select vehicle type”.

En esta página, se le presenta al administrador una gráfica con la información de las simulaciones de todos los usuarios, en función de la información que seleccione en los selectores comentados, como se puede ver en la figura 189.

En el primero de estos tres selectores “Select scenario”, el administrador tiene la posibilidad de seleccionar el escenario en el que los usuarios han realizado las simulaciones. Este selector de escenarios dispone de las siguientes opciones:

- *Escenario 1: Ruta urbana.*
- *Escenario 3: Ruta urbana de adentro hacia fuera.*
- *Escenario 2: Ruta urbana afuera.*
- *Escenario 4: Ruta urbana de afuera hacia adentro.*
- *Escenario 5: Ruta urbana no guiada.*
- *Escenario 6: Ruta de circunvalación M-30.*
- *Escenario 7: A través de la ruta por carretera.*
- *Escenario 8: Ruta de la autopista.*
- *Escenario 9: Ruta urbana en Valladolid (Pilarica - San Juan).*
- *Escenario 10: Vía urbana, autopista y vía de paso en Valladolid.*
- *Escenario 11: Ruta urbana, autopista, carretera interurbana y carretera a través de Valladolid.*



Figura 189: Pestaña “Simulation data” de la página “Analysis of vehicle data” de la plataforma web.

En el segundo de estos tres selectores “Select data to show”, el administrador puede seleccionar el tipo de dato a analizar en la gráfica, disponiendo de las siguientes opciones:

- *Consumo medio de combustible.*
- *Tiempo promedio.*
- *Velocidad media.*
- *RPM promedio.*

En el tercero y último de los selectores “Select vehicle type”, se ofrece la posibilidad de filtrar las simulaciones en función del tipo de vehículo con el que los usuarios han realizado las simulaciones. En este selector el administrador dispone de las siguientes opciones:

- *Vehículo de pasajeros con motor de combustión interna.*
- *Turismo eléctrico.*
- *Autobús urbano con motor de combustión interna.*
- *Autobús eléctrico de la ciudad.*
- *Camión con motor de combustión interna.*
- *Camión eléctrico.*

Utilizando la información que el administrador elige en los tres selectores y, después de pulsar en el botón “Filter graph”, la información almacenada en la base de datos referente a las simulaciones se filtra siguiendo estos criterios. Después de este proceso, la información deseada por el administrador se muestra en la gráfica. Esta página ofrece la posibilidad de analizar este tipo de datos y compararlos a los distintos usuarios en función de distintos criterios.

El siguiente apartado de la pestaña “Analysis of vehicle data” es “Simulation data per group”. Este apartado permite crear grupos con distintas características para poder analizar y comparar los datos de las simulaciones de los grupos de usuarios creados. Al igual que en el apartado anterior, esta página permite seleccionar un escenario en el que se han realizado las simulaciones en el selector “Select escenario”, el tipo de dato que se va a analizar en el selector “Select data to show” y el tipo de vehículo con el que los usuarios realizaron las simulaciones en el selector “Select vehicle type”, como se muestra en la figura 190.

El comportamiento de este apartado para el administrador es el mismo que el del apartado anterior pero enfocado a grupos y no a usuarios. Se le presenta una gráfica con la información de las simulaciones de todos grupos creados utilizando los selectores de la parte inferior de la página. Sobre estos selectores para crear los grupos se comentará el funcionamiento más adelante.

Ahora, se hablará de los selectores que se sitúan bajo la gráfica. En el primero de estos tres selectores “Select escenario”, el administrador tiene la posibilidad de seleccionar un escenario en el que los usuarios de los grupos han realizado las simulaciones. Este selector de escenarios dispone de las siguientes opciones:

- *Escenario 1: Ruta urbana.*
- *Escenario 3: Ruta urbana de adentro hacia fuera.*
- *Escenario 2: Ruta urbana afuera.*
- *Escenario 4: Ruta urbana de afuera hacia adentro.*
- *Escenario 5: Ruta urbana no guiada.*
- *Escenario 6: Ruta de circunvalación M-30.*
- *Escenario 7: A través de la ruta por carretera.*
- *Escenario 8: Ruta de la autopista.*
- *Escenario 9: Ruta urbana en Valladolid (Pilarica - San Juan).*
- *Escenario 10: Vía urbana, autopista y vía de paso en Valladolid.*
- *Escenario 11: Ruta urbana, autopista, carretera interurbana y carretera a través de Valladolid.*

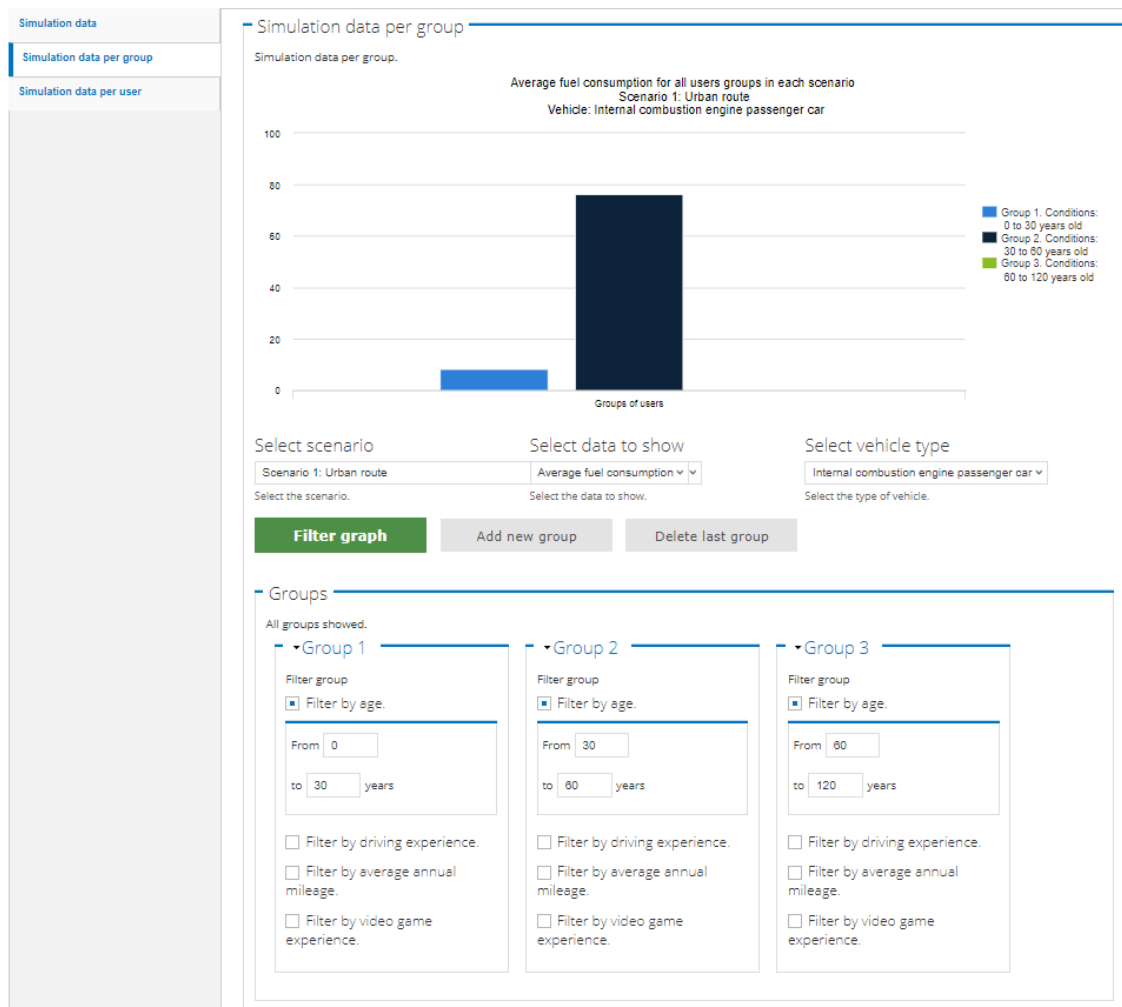


Figura 190: Pestaña “Simulation data per group” de la página “Analysis of vehicle data” de la plataforma web.

En el segundo de estos tres selectores “Select data to show”, el administrador puede seleccionar el tipo de dato a analizar en la gráfica, disponiendo de las siguientes opciones:

- *Consumo medio de combustible.*
- *Tiempo promedio.*
- *Velocidad media.*
- *RPM promedio.*

En el tercero y último de los selectores “Select vehicle type”, se ofrece la posibilidad de filtrar las simulaciones en función del tipo de vehículo con el que los usuarios de los grupos han realizado las simulaciones. En este selector, el administrador dispone de las siguientes opciones:

- *Vehículo de pasajeros con motor de combustión interna.*
- *Turismo eléctrico.*
- *Autobús urbano con motor de combustión interna.*

- *Autobús eléctrico de la ciudad.*
- *Camión con motor de combustión interna.*
- *Camión eléctrico.*

Como se comentó anteriormente, en esta página el administrador tiene la posibilidad de agrupar a usuarios en función de distintos criterios, así puede analizar y comparar los datos de estos usuarios que cumplen ciertos criterios, con los datos de otros grupos de usuarios creados en función de otros criterios.

Los criterios que dispone el administrador para filtrar a los usuarios y poder agruparlos son los siguientes:

- Filtrar por edad: permite al administrador añadir, al grupo que va a crear, a los usuarios que se encuentran dentro de un rango de edad.
- Filtrar por experiencia de conducción: permite al administrador añadir, al grupo que va a crear, a los usuarios que se encuentran dentro de un rango de años de experiencia como conductores.
- Filtrar por kilometraje promedio anual: permite al administrador añadir, al grupo que va a crear, a los usuarios que se encuentran dentro de un rango de kilómetros recorridos por año.
- Filtrar por experiencia en videojuegos: permite al administrador añadir, al grupo que va a crear, a los usuarios que tienen o no experiencia jugando a videojuegos.

Encima de este apartado, el administrador dispone de dos botones: uno para añadir un nuevo grupo “Add new group” y otro para eliminar el último grupo creado “Delete last group”.

Utilizando los datos que el administrador elige en los distintos selectores y después de pulsar en el botón “Filter graph”, la información almacenada en la base de datos referente a las simulaciones se filtra siguiendo estos criterios. Después de este proceso, la información seleccionada por el administrador se muestra en la gráfica, ofreciendo la posibilidad de analizar este tipo de datos y compararlos con los distintos grupos de usuarios creados en función de distintos criterios.

El último apartado de la pestaña “Analysis of vehicle data” es “Simulation data per user”. Este apartado permite comparar los datos de un usuario en concreto con grupos de usuarios creados con distintas características, permitiendo seleccionar el usuario del que se quieren comparar los datos con grupos en el selector “Select User”. Además, al igual que en los apartados anteriores, esta página permite seleccionar el escenario en el que se realizaron las simulaciones en el selector “Select scenario”, el tipo de dato que se va a analizar en el selector “Select data to show” y el tipo de vehículo con el que los usuarios realizan las simulaciones en el selector “Select vehicle type”.

El comportamiento de este apartado para el administrador es el mismo que el de los apartados anteriores pero enfocado a un usuario en concreto en comparación con grupos de usuarios. Se le presenta una gráfica con la información de las simulaciones del usuario y de todos grupos creados utilizando los selectores de la parte inferior de la página. Se comentará el funcionamiento de estos selectores para crear los grupos más adelante, todo esto se puede observar en la figura 191. Ahora, se hablará de los selectores que se sitúan bajo la gráfica. En el primero de estos cuatro selectores “Select User”, el administrador puede elegir el usuario del que quiere analizar la información, en el segundo de los selectores “Select escenario”, el administrador tiene la posibilidad de seleccionar un escenario en el que los usuarios de los grupos y el usuario seleccionado han realizado las simulaciones. Este selector de escenarios dispone de las opciones que se comentaron con anterioridad, por lo que no se repetirán.

Home » Administration » Analysis of Simulations

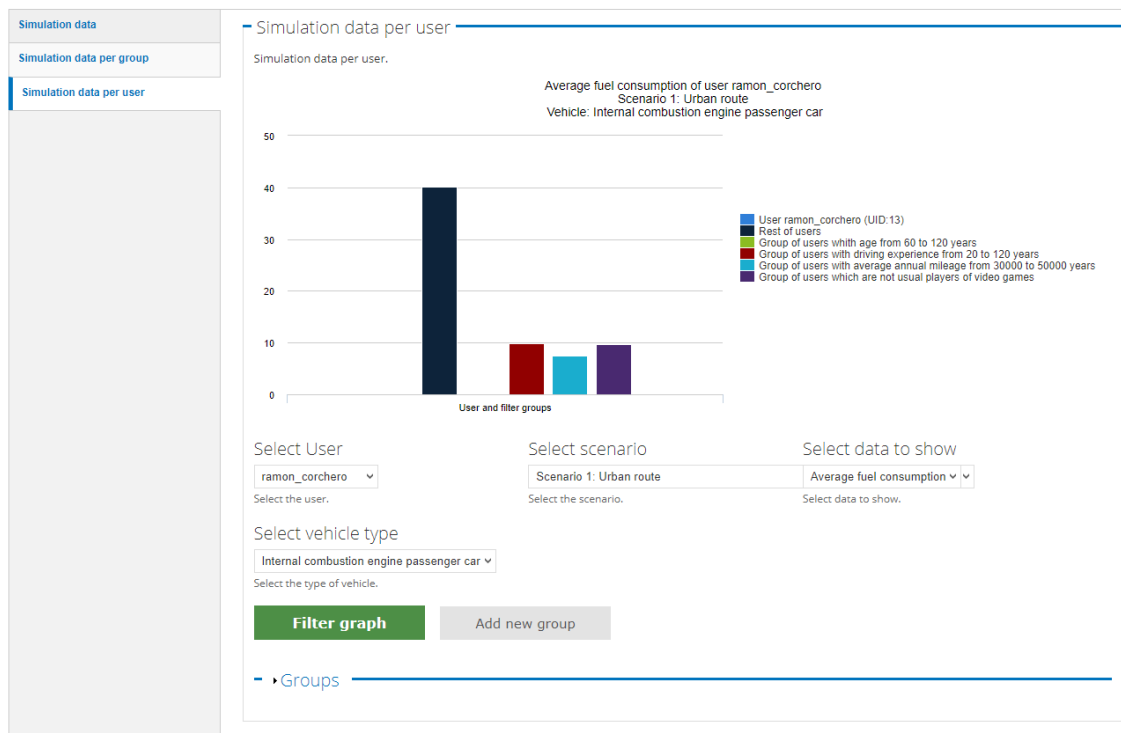


Figura 191: Pestaña “Simulation data per user” de la página “Analysis of vehicle data” de la plataforma web.

En el tercero de los cuatro selectores “Select data to show”, el administrador puede seleccionar el tipo de dato a analizar en la gráfica, disponiendo de las mismas opciones que se comentaron antes.

El cuarto y último de los selectores “Select vehicle type”, ofrece la posibilidad de filtrar las simulaciones en función del tipo de vehículo con el que los usuarios de los grupos y el usuario seleccionado han realizado las simulaciones, en este selector el administrador dispone de las mismas opciones que los selectores de este tipo de los que se habló con anterioridad.

Como se comentó anteriormente, en esta página el administrador tiene la posibilidad de agrupar a usuarios en función de distintos criterios, para así poder analizar y comparar los datos de los usuarios que cumplen ciertos criterios, con los datos de un usuario en concreto.

Los criterios que dispone el administrador para filtrar a los usuarios y poder agruparlos son los siguientes:

- Filtrar por edad: permite al administrador añadir al grupo que va a crear a los usuarios que se encuentran dentro de un rango de edad.
- Filtrar por experiencia de conducción: permite al administrador añadir al grupo que va a crear a los usuarios que se encuentran dentro de un rango de años de experiencia como conductores.
- Filtrar por kilometraje promedio anual: permite al administrador añadir al grupo que va a crear a los usuarios que se encuentran dentro de un rango de kilómetros recorridos por año.
- Filtrar por experiencia en videojuegos: permite al administrador añadir al grupo que va a crear a los usuarios que tienen o no experiencia jugando a videojuegos.

El administrador dispone, encima de este apartado, de dos botones: uno para añadir un nuevo grupo “Add new group” y otro para eliminar el último grupo creado “Delete last group”.

Utilizando los datos que el administrador elige en los distintos selectores y después de pulsar en el botón “Filter graph”, la información almacenada en la base de datos referente a las simulaciones se filtra siguiendo estos criterios. Después de este proceso, la información deseada por el administrador se muestra en la gráfica. Esta página ofrece la posibilidad de analizar este tipo de datos y comparar a los distintos grupos de usuarios creados con los datos de un usuario seleccionado en función de distintos criterios.

A continuación, se analizará la pestaña “Analysis of traffic offenses”. En esta pestaña, nos vamos a centrar en el apartado del mismo nombre. Esta página permite analizar y comparar los datos de todos los usuarios de la plataforma a la vez, permitiendo seleccionar un escenario en el que se han realizado las simulaciones en el selector “Select scenario”, el tipo de infracción cometida por los usuarios en el selector “Select traffic offenses” y el tipo de vehículo con el que los usuarios realizaron las simulaciones en el selector “Select vehicle type”.

En esta página, al administrador se le presenta una gráfica con la información de las infracciones cometidas en las simulaciones realizadas por todos los usuarios, en función de la información que seleccione en los selectores situados debajo de la gráfica, como se muestra en la figura 192.

En el primero de estos tres selectores “Select scenario”, el administrador tiene la posibilidad de seleccionar un escenario en el que los usuarios han realizado las simulaciones, este selector de escenarios dispone de las mismas opciones que los otros selectores de escenario de los que se habló con anterioridad.



Figura 192: Pestaña “Analysis of traffic offenses” de la página “Analysis of traffic offenses” de la plataforma web.

En el segundo de estos tres selectores “Select traffic offenses”, el administrador puede seleccionar el tipo de infracción cometida por los usuarios en el escenario seleccionado en el selector anterior, disponiendo de las siguientes opciones, siendo infracciones de seguridad:

- *Golpear a un peatón*
- *Colisión con otro vehículo*
- *Conducir fuera de la carretera*
- *Excede el límite de velocidad*
- *Salida incorrecta en una rotonda*
- *Camino equivocado en una rotonda*
- *Dirección opuesta en la rotonda*
- *Salida incorrecta en una encrucijada*
- *No detenerse en una señal de alto*
- *Falta de rendimiento en un signo de rendimiento*
- *No detenerse en un semáforo rojo*
- *Colisión con mobiliario urbano*
- *Cruzar una línea continua*
- *Las luces bajas no se cambian conduciendo debajo de un túnel*
- *Vigas no cambiadas por la noche*
- *Giro equivocado en una encrucijada*
- *Estar más cerca de la distancia mínima al vehículo que está adelante*
- *No encienda la luz de giro correspondiente en una vuelta*
- *No encienda la luz de giro correspondiente en un adelantamiento*
- *Conduzca por debajo de la velocidad mínima*
- *Invasión de carriles de autobuses*

- *Pare en líneas entrecruzadas amarillas*
- *Atropellar a un ciclista*
- *Adelantar a un ciclista sin abandonar la distancia mínima de seguridad*

Y siendo infracciones de eficiencia:

- *Rotura repentina (sin usar el frenado del motor)*
- *Aceleración no progresiva (de 0 a 50 km / h en menos de 5 segundos)*
- *Aceleración repentina*

En el tercero y último de los selectores “Select vehicle type”, ofrece la posibilidad de filtrar las simulaciones en función del tipo de vehículo con el que los usuarios las han realizado., en este tipo de selector ya se comentaron las opciones de las que dispone el administrador.

Utilizando la información que el administrador elige en los tres selectores y, después de pulsar en el botón “Filter graph”, la información almacenada en la base de datos referente a las simulaciones se filtra siguiendo estos criterios. Después de este proceso, la información deseada por el administrador se muestra en la gráfica. Esta página ofrece la posibilidad de analizar el número de infracciones cometidas en las simulaciones de todos los usuarios de la plataforma y compararlas en función de distintos criterios.

El siguiente apartado de la pestaña “Analysis of traffic offenses” es “Analysis of traffic offenses per group”. Este apartado permite crear grupos con distintas características para poder analizar y comparar las infracciones cometidas por estos grupos de usuarios en las simulaciones. Al igual que en el apartado anterior, esta página permite seleccionar el escenario en el que se han realizado las simulaciones en el selector “Select escenario”, el tipo de infracción cometida en la simulación en el selector “Select traffic offenses” y el tipo de vehículo con el que los usuarios realizaron las simulaciones en el selector “Select vehicle type”.

El comportamiento de este apartado para el administrador es el mismo que el del apartado anterior pero enfocado a grupos y no a usuarios. Se le presenta una gráfica con la información de las infracciones cometidas en las simulaciones de todos los grupos creados utilizando los selectores de la parte inferior de la página. Sobre estos selectores para crear los grupos se comentará el funcionamiento más adelante, todo esto se puede ver en la figura 193.

Ahora se hablará de los selectores que se sitúan bajo la gráfica, en el primero de estos tres selectores “Select escenario”, el administrador tiene la posibilidad de seleccionar el escenario en el que los usuarios de los grupos han realizado las simulaciones, el contenido del selector de escenarios ya se comentó con anterioridad.

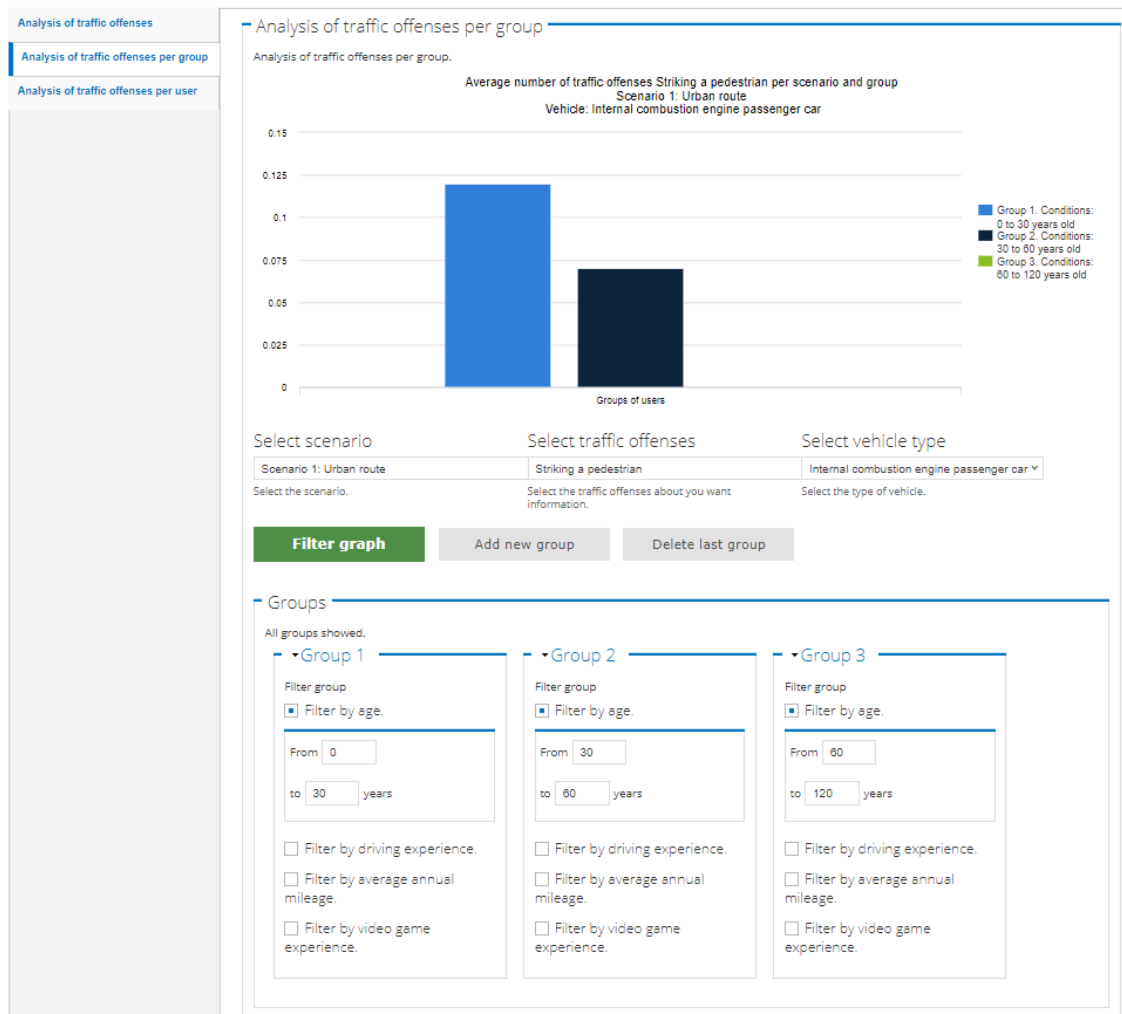


Figura 193: Pestaña “Analysis of traffic offenses per group” de la página “Analysis of traffic offenses” de la plataforma web.

En el siguiente selector “Select traffic offenses”, el administrador puede seleccionar el tipo de infracción cometida por los usuarios de los grupos creados en el escenario seleccionado en el selector anterior. De los tipos de infracciones disponibles ya se habló con anterioridad.

En el tercero y último de los selectores “Select vehicle type”, ofrece la posibilidad de filtrar las simulaciones en función del tipo de vehículo con el que los usuarios de los grupos han realizado las simulaciones, los tipos de vehículos disponibles en la plataforma ya se comentaron antes.

Como se comentó anteriormente en esta página, el administrador tiene la posibilidad de agrupar a usuarios en función de distintos criterios, así puede analizar y comparar las infracciones cometidas por estos usuarios que cumplen ciertos criterios, con las infracciones cometidas por otros grupos de usuarios creados en función de otros criterios. De los criterios que dispone el administrador para filtrar a los usuarios y poder agruparlos ya se habló con anterioridad.

El administrador dispone, encima de este apartado, de dos botones: uno para añadir un nuevo grupo “Add new group” y otro para eliminar el último grupo creado “Delete last group”.

Utilizando los datos que el administrador elige en los distintos selectores y después de pulsar en el botón “Filter graph”, la información almacenada en la base de datos referente a las infracciones cometidas en las simulaciones se filtra siguiendo estos criterios. Después de este proceso, la información deseada por el administrador se pinta en la gráfica. Esta página ofrece la posibilidad de analizar las infracciones cometidas en un escenario y compararlas con las de distintos grupos de usuarios creados en función de distintos criterios.

El último apartado de la pestaña “Analysis of traffic offenses” es “Analysis of traffic offenses per user”. Este apartado permite comparar las infracciones cometidas en las simulaciones de un usuario en concreto con grupos de usuarios creados con distintas características, haciendo posible seleccionar el usuario del que se quieren comparar las infracciones cometidas con las infracciones cometidas por grupos de usuarios en el selector “Select User”. Además, al igual que en los apartados anteriores, esta página permite seleccionar el escenario en el que se realizaron las simulaciones en el selector “Select scenario”, el tipo de infracciones cometidas en las simulaciones en el selector “Select traffic offenses” y el tipo de vehículo con el que los usuarios realizan las simulaciones en el selector “Select vehicle type”, todo esto se puede apreciar en la figura 194.

El comportamiento de este apartado para el administrador es el mismo que el de los apartados anteriores pero enfocado a un usuario en concreto en comparación con grupos de usuarios. Se le presenta una gráfica con la información de las simulaciones del usuario y de todos grupos creados utilizando los selectores de la parte inferior de la página. Se comentará el funcionamiento de estos selectores para crear los grupos más adelante.

Ahora, se hablará de los selectores que se sitúan bajo la gráfica. En el primero de estos cuatro selectores “Select User”, el administrador puede elegir el usuario del que quiere analizar la información, en el segundo de los selectores “Select scenario”, el administrador tiene la posibilidad de seleccionar un escenario en el que los usuarios de los grupos y el usuario seleccionado han realizado las simulaciones, de las opciones disponibles en este selector ya se habló anteriormente.

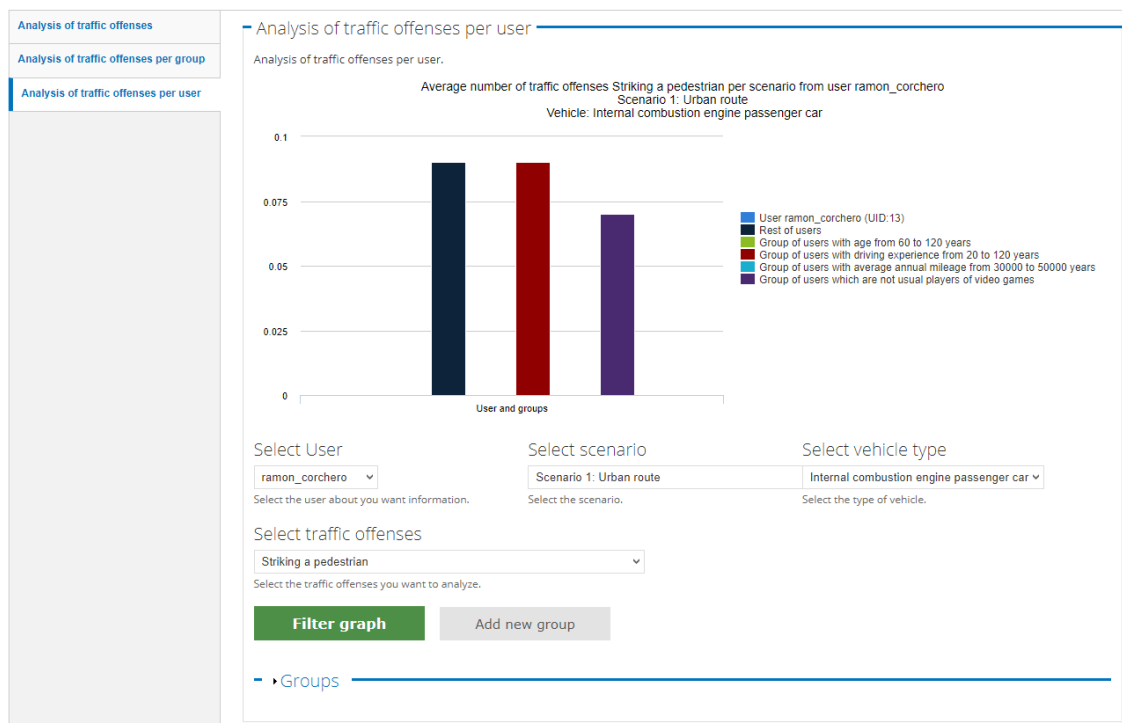


Figura 194: Pestaña “Analysis of traffic offenses per user” de la página “Analysis of traffic offenses” de la plataforma web.

En el tercero de los cuatro selectores “Select vehicle type”, ofrece la posibilidad de filtrar las simulaciones en función del tipo de vehículo con el que los usuarios de los grupos y el usuario seleccionado han realizado las simulaciones, de los tipos de vehículos disponibles ya se ha hablado varias veces anteriormente

En el cuarto y último de los selectores “Select traffic offenses”, el administrador puede seleccionar el tipo de infracción cometida por el usuario seleccionado y por los usuarios de los grupos creados en el escenario seleccionado en el selector anterior, las opciones disponibles en este selector son las mismas que en otros de este mismo tipo de los que ya se ha hablado.

Como se comentó anteriormente, en esta página el administrador tiene la posibilidad de agrupar a usuarios en función de distintos criterios, para así poder analizar y comparar los datos de los usuarios que cumplen ciertos criterios, con los datos de un usuario en concreto, estos criterios son los mismos que de los que se habló anteriormente.

El administrador dispone, encima de este apartado, de dos botones: uno para añadir un nuevo grupo “Add new group” y otro para eliminar el último grupo creado “Delete last group”.

Utilizando los datos que el administrador elige en los distintos selectores y después de pulsar en el botón “Filter graph”, la información almacenada en la base de datos referente a las simulaciones se filtra siguiendo estos criterios. Después de este proceso, la información deseada por el administrador se muestra en la gráfica. Esta página ofrece la posibilidad de analizar este tipo de datos y comparar a los distintos grupos de usuarios creados con los datos de un usuario seleccionado en función de distintos criterios.

Una vez que se ha comentado la posibilidad que ofrecen las pestañas “Analysis of vehicle data” y “Analysis of traffic offenses”, se pasará a hablar de qué puede hacer el administrador en la pestaña “Analysis of Simulations per user”, esta pagina se puede ver en la figura 195.

Dentro de la pestaña “Analysis of Simulations per user” el administrador tiene la posibilidad de ver la información que es accesible para todos los usuarios cuando en la pantalla de inicio seleccionan “My data about the simulation”. El administrador tiene una serie de posibilidades de las que los usuarios no disponen en la página “My data about the simulation”, que se comentarán más adelante.

Si el administrador pulsa en el hipertexto “Show info about simulations” de cualquier usuario de la tabla, incluyéndole a él mismo, puede ver la información sobre las simulaciones de los usuarios, pero con la posibilidad de ver el análisis de datos de las simulaciones de una forma que no puede ver en las páginas anteriormente comentadas. Esta información de las simulaciones se comentará en la sección “Manual de usuario de la plataforma para un usuario común”.

The screenshot shows a web application interface with a dark header and a light-colored content area. The header includes the title 'Analysis of Simulations' and navigation tabs for 'Analysis of Simulations per user', 'Simulation data', and 'Analysis of traffic offenses'. Below the header, there is a 'Load from file' button and a table with the following columns: User ID, User Name, Age, Driving experience (years), Average annual mileage, Usual videogame player, Email, Created, Last login, Last access, and Links. The table contains 20 rows of user data, each with a 'Show info about simulations' link in the 'Links' column. At the bottom of the table, there is a pagination control showing '1 2 3 4 5 next last'.

User ID	User Name	Age	Driving experience (years)	Average annual mileage	Usual videogame player	Email	Created	Last login	Last access	Links
1		32	14	8000	Yes		17-02-2016 21:08:05	04-02-2020 19:45:59	04-02-2020 20:01:34	Show info about simulations
2		47	20	12000	No		18-02-2016 20:59:58	01-12-2018 13:21:45	01-12-2018 14:20:38	Show info about simulations
13		60	40	30000	No		17-06-2016 09:59:55	19-06-2016 21:15:06	19-06-2016 21:13:13	Show info about simulations
14		31	13	50000	Yes		17-06-2016 10:01:29	07-07-2016 11:56:53	07-07-2016 11:56:53	Show info about simulations
15		58	22	16000	No		18-06-2016 19:37:48	18-06-2016 19:37:48	01-01-1970 01:00:00	Show info about simulations
16		39	20	20000	Yes		19-06-2016 20:55:51	23-06-2016 19:52:29	23-06-2016 19:52:29	Show info about simulations
26		29	12	30000	No		27-06-2016 13:37:19	27-06-2016 13:37:19	01-01-1970 01:00:00	Show info about simulations
27		29	12	1000	Yes		27-06-2016 14:25:26	27-06-2016 14:25:26	01-01-1970 01:00:00	Show info about simulations
31		25	7	10000	Yes		28-06-2016 13:53:50	30-10-2018 18:13:21	30-10-2018 18:13:21	Show info about simulations
35		47	15	3500	No		29-06-2016 19:08:36	29-06-2016 19:08:36	01-01-1970 01:00:00	Show info about simulations
38		29	12	1000	Yes		07-07-2016 12:40:42	15-02-2018 19:18:07	15-02-2018 19:18:07	Show info about simulations
41		40	20	20000	Yes		07-07-2016 18:15:01	13-02-2017 21:37:13	13-02-2017 21:37:13	Show info about simulations
52		28	11	10000	Yes		13-07-2016 14:54:41	23-12-2016 12:03:01	23-12-2016 12:03:01	Show info about simulations
69		27	10	6000	No		15-07-2016 17:52:27	15-07-2016 17:52:27	15-07-2016 17:52:27	Show info about simulations
77		28	10	20000	Yes		13-12-2016 14:09:30	19-12-2016 18:07:55	19-12-2016 18:06:86	Show info about simulations
79		28	10	7000	Yes		19-12-2016 17:46:31	23-12-2016 17:52:13	23-12-2016 17:50:58	Show info about simulations
83		28	29	200	No		20-12-2016 12:41:57	07-02-2018 16:28:37	07-02-2018 16:28:37	Show info about simulations
84		43	25	15000	Yes		21-12-2016 10:01:34	21-12-2016 12:47:54	21-12-2016 12:46:44	Show info about simulations
92		24	6	10000	No		21-12-2016 18:11:11	22-12-2016 10:19:49	22-12-2016 10:19:49	Show info about simulations
93		28	11	6000	Yes		23-12-2016 17:54:25	07-02-2017 18:34:04	07-02-2017 18:32:27	Show info about simulations

Figura 195: Pagina “Analysis of Simulation per user” de la plataforma web.

Cuando el administrador pulsa en el hipertexto “Show info about simulations” de cualquier usuario, incluido él, y navega por las páginas a las que accede, llega a la página “Simulation in escenario X”. Aunque las funcionalidades de esta página se comentarán en el siguiente apartado, resulta interesante comentar aquí una característica de la que dispone el administrador en esta página. En la tabla de la parte inferior de la página donde aparecen todas las simulaciones realizadas en el escenario, que se ha debido seleccionar para llegar hasta esta página, el administrador tiene la posibilidad de borrar de la base de datos la información de una simulación pulsando en el hipertexto “Delete Simulation” de la fila que contiene la información de la simulación que se desea eliminar, se puede ver observando la figura 196.

Date	Scenario name	Vehicle type	Average fuel consumption	Total fuel consumption	Total time	Links	Delete Link
2017-02-14 20:08:01	Scenario 1: Urban route	Internal combustion engine passenger car	9.82	0.02	30.89	See Simulation	Delete this game
2016-12-20 20:32:41	Scenario 1: Urban route	Internal combustion engine passenger car	10.39	0.16	312.92	See Simulation	Delete this game
2016-07-13 14:50:16	Scenario 1: Urban route	Internal combustion engine passenger car	8.19	0.02	16.66	See Simulation	Delete this game
2016-07-13 14:22:01	Scenario 1: Urban route	Internal combustion engine passenger car	7.33	0.02	16.4	See Simulation	Delete this game
2016-07-13 14:18:03	Scenario 1: Urban route	Internal combustion engine passenger car	11.24	0.01	11.64	See Simulation	Delete this game

1 2 Next »

[Go Back](#)

Figura 196: Parte inferior de la página “Simulation in scenario X” de la plataforma web.

En este apartado se ha buscado explicar todas las funcionalidades características que aporta la plataforma web después de su actualización, cuando se valida en ella un usuario con permisos de administrador.

5.2 Manual de usuario de la plataforma para un usuario común

Una vez que se han comentado las posibilidades que aporta la plataforma web cuando un usuario se valida en ella con permisos de administrador, en este apartado se analizará qué puede ver y hacer un usuario sin permisos de administrador al validarse en la plataforma.

Cuando un usuario común sin permisos de administrador que ha realizado simulaciones en el simulador de conducción se valida en la plataforma web lo primero que ve en ella es la pantalla de inicio.

En la pantalla de inicio de la plataforma web un usuario sin permisos de administrador no dispone de muchas posibilidades. Solo puede acceder a la página “My data about the Simulations” que le permite acceder al análisis de los datos de las simulaciones que ha realizado, esta página se puede observar en la figura 197.

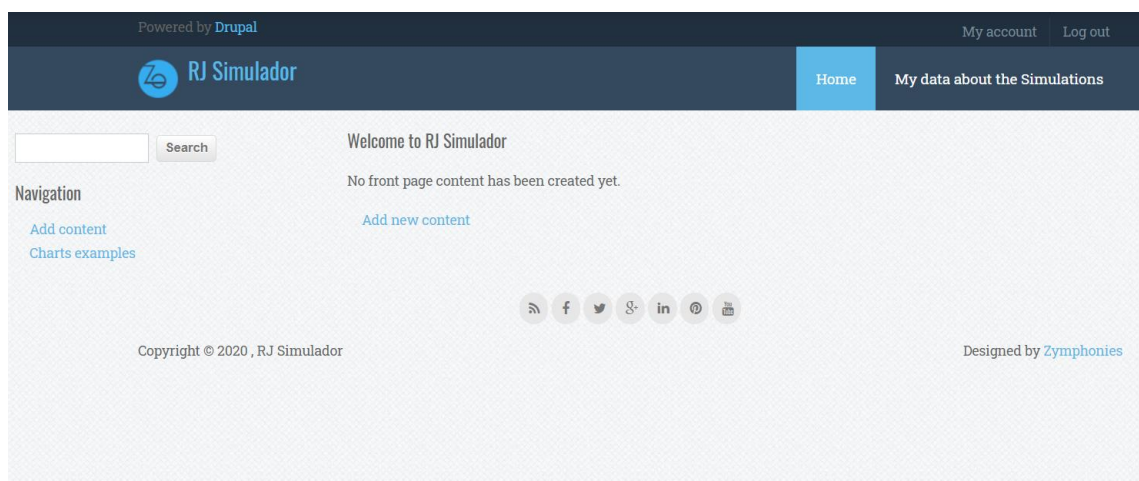


Figura 197: Página de inicio de la plataforma.

Una vez que el usuario ha pulsado en la pestaña de la parte superior derecha de la página de inicio de la plataforma llamada “My data about the Simulations”, accede a la página que tiene el mismo nombre que la pestaña que pulsó.

Una vez dentro de la página “My data about the Simulations” se van a explicar las funcionalidades de la misma separándola en dos partes, la parte superior que

abarcará los cuatro primeros gráficos y la parte inferior que incluye un gráfico circular y una tabla.

La parte superior cuenta con cuatro gráficos, como se ha comentado, “Total number of simulations fulfilled by the user”, “Average number of traffic offenses per scenario”, “Total number of simulations fulfilled by the user in the last days” y “Average time to drive in each scenario”. Se empezará analizando el primero de ellos “Total number of simulations fulfilled by the user”. Este gráfico de barras muestra el número total de simulaciones realizadas por el usuario. El usuario tiene la posibilidad de filtrar las distintas simulaciones realizadas por el tipo de vehículo con el que se hicieron utilizando el selector “Select vehicle type”, de este tipo de selector ya se habló antes y de sus opciones. En el eje horizontal del gráfico aparecen los nombres de los escenarios para que el usuario pueda comparar los datos de todos los escenarios a la vez y saber a qué escenario pertenece cada número de simulaciones que indica el gráfico.

El segundo de los gráficos es “Average number of traffic offenses per scenario”. Este gráfico de barras permite comparar el número promedio de infracciones de tráfico por escenario del usuario con el resto de usuarios. En este gráfico el usuario tiene la posibilidad de filtrar las distintas simulaciones realizadas por el tipo de vehículo con el que se hicieron utilizando el selector “Select vehicle type”, de selectores similares a este se habló en apartados anteriores.

En el eje horizontal del gráfico aparecen los nombres de los escenarios para que el usuario pueda comparar los datos de todos los escenarios a la vez y saber a qué escenario pertenece el número promedio de infracciones de tráfico, tanto suyas como del resto de los usuarios.

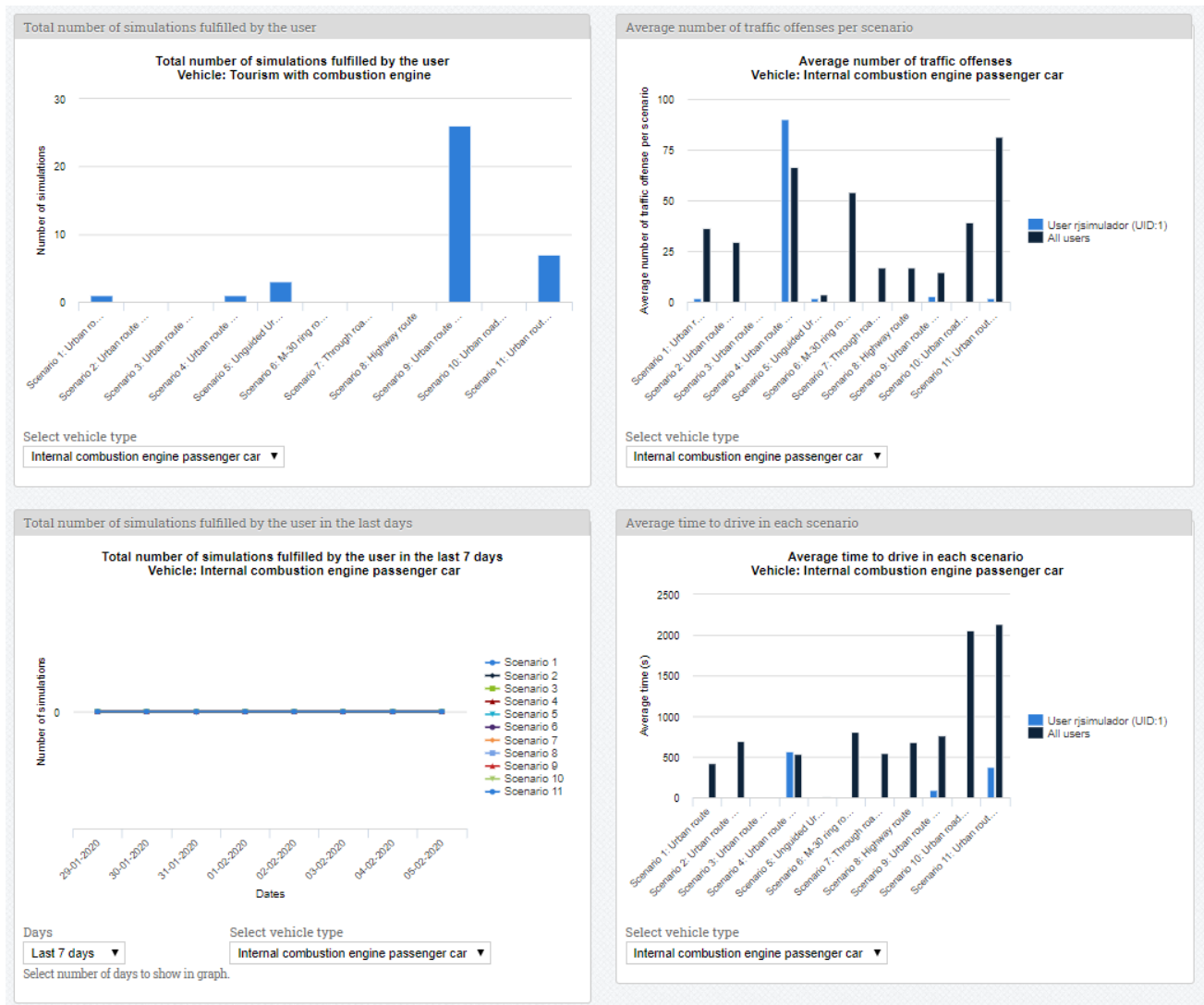


Figura 198: Parte superior de la página “My data about the Simulation” de la plataforma web.

El siguiente gráfico es “Total number of simulations fulfilled by the user in the last days”. Este gráfico permite saber el número total de simulaciones realizadas por el usuario en los últimos días. En él, el usuario tiene la posibilidad de seleccionar el número de días sobre los que se muestra información utilizando el selector “Days”. Este selector cuenta con las siguientes opciones:

- Últimos 7 días.
- Últimos 15 días.
- Últimos 30 días.

Por otro lado, el usuario puede filtrar las distintas simulaciones realizadas por el tipo de vehículo con el que se hicieron utilizando el selector “Select vehicle type”, como en otros casos de este selector y de las opciones de las que dispone ya se ha hablado anteriormente.

En el eje horizontal del gráfico aparecen las fechas en función de lo que seleccionó el usuario en el selector “Days”. Así, el usuario puede saber el número de simulaciones que ha realizado cada día orientándose por las fechas.

El último de los gráficos de la parte superior de la página del que se hablará es “Average time to drive in each scenario”. Este gráfico de barras permite comparar el tiempo promedio para completar la conducción en cada escenario del usuario con el resto de usuarios de la plataforma. En este gráfico el usuario tiene la posibilidad de filtrar las distintas simulaciones realizadas por el tipo de vehículo con el que se hicieron utilizando el selector “Select vehicle type”, ya se ha hablado varias veces del contenido de este tipo de selector con anterioridad.

En el eje horizontal del gráfico aparecen los nombres de los escenarios para que el usuario pueda comparar los datos de todos los escenarios a la vez y saber a qué escenario pertenece el tiempo promedio mostrado tanto suyo como del resto de los usuarios, todo esto que se ha venido comentando se puede apreciar en la figura 198.

La parte inferior de la página cuenta con un gráfico circular y una tabla con los distintos escenarios en los que se pueden realizar simulaciones. El gráfico se llama “Percentages of traffic offences for all simulations”. Es un gráfico circular que permite analizar el porcentaje de infracciones de tráfico para todas las simulaciones del usuario. En él, el usuario tiene la posibilidad de filtrar las distintas simulaciones realizadas por el tipo de vehículo con el que se hicieron utilizando el selector “Select vehicle type”, del contenido de este selector ya se ha hablado con anterioridad.

En el lateral derecho del gráfico aparece una leyenda de colores para ayudar al usuario a identificar las distintas infracciones por colores. Cada una se representa por un color distinto en el gráfico.

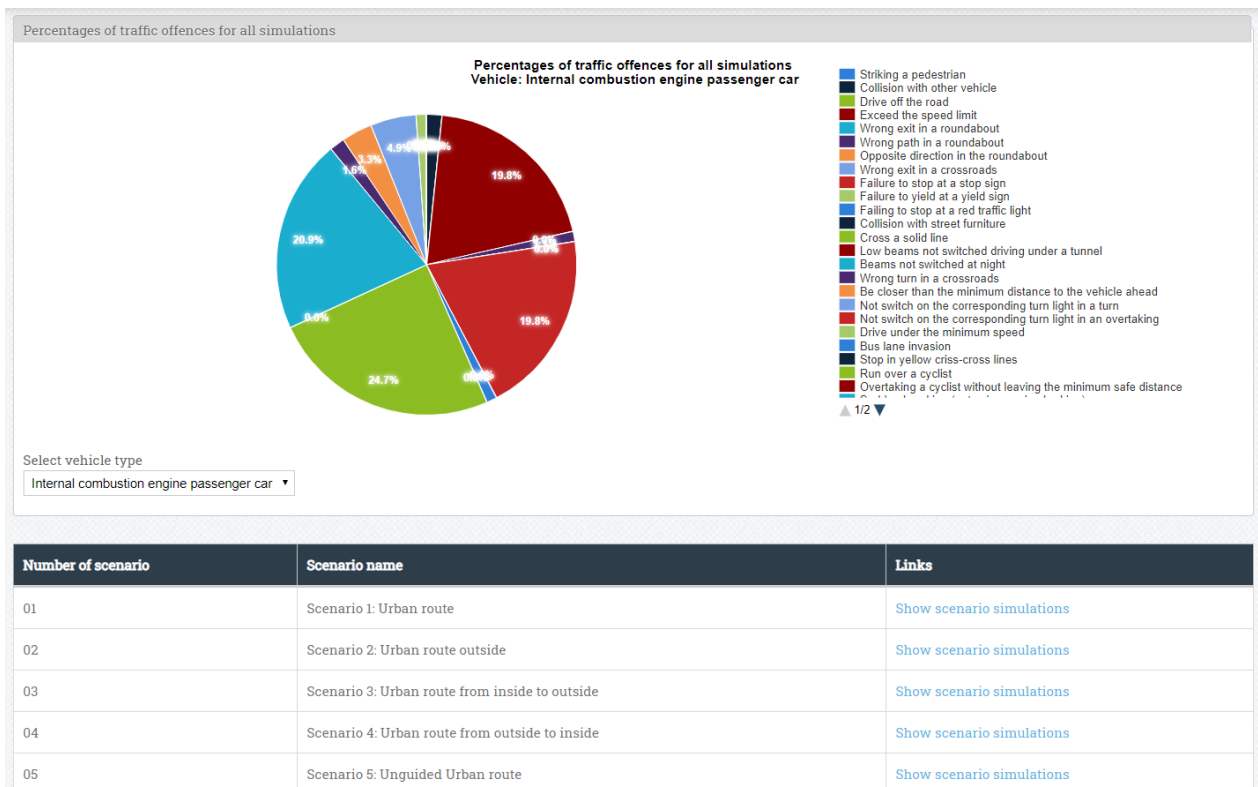


Figura 199: Parte inferior de la página “My data about the Simulation” de la plataforma web.

En la parte final de la página existe una tabla, como se ha comentado anteriormente. En esta tabla vienen enumerados los distintos escenarios. Si el usuario pulsa en uno de los enlaces “Show scenario simulations” de uno de los escenarios de la tabla, el usuario accederá a la página “Simulations in scenario X” del escenario correspondiente, todo lo comentado anteriormente se puede observar en la figura 199.

En la página “Simulations in scenario X”, el usuario ya puede analizar los datos de las simulaciones filtrados por el escenario seleccionado en la tabla de la página anterior. El análisis de esta página se va a dividir en dos partes, en la parte superior existen dos gráficos y en la parte inferior existe un gráfico y una tabla. Los dos gráficos que existen en la parte superior se llaman “Comparison of average fuel consumption” y “Average time to drive in scenario”. El primero de los gráficos del que se va a comentar sus funcionalidades es “Comparison of average fuel consumption”. Este gráfico permite realizar una comparación del consumo medio de combustible en este escenario del usuario con respecto al resto de usuarios de la plataforma. En él, el usuario tiene la posibilidad de filtrar las distintas simulaciones realizadas por el tipo de vehículo con el que se hicieron utilizando el selector “Select vehicle type”, de este selector se viene hablando durante todo el manual y su contenido es el mismo que para otros casos.

En el eje horizontal del gráfico aparecen las fechas en las que se realizaron las simulaciones en este escenario para que el usuario pueda comparar sus datos con los del resto de los usuarios.

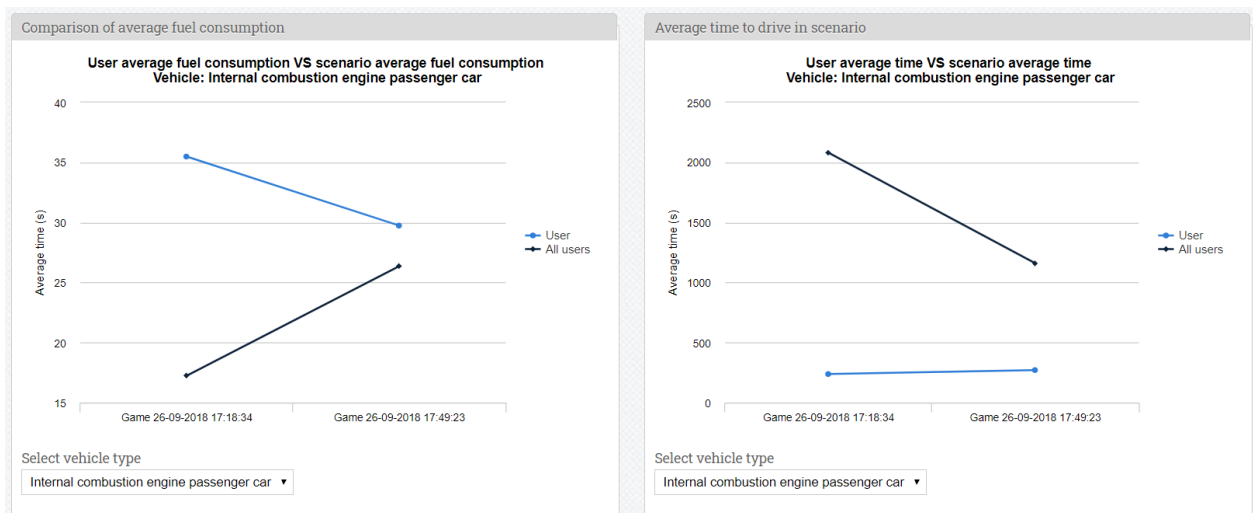


Figura 200: Parte superior de la página “Simulation in scenario X” de la plataforma web.

El segundo de los gráficos es “Average time to drive in scenario”. Este gráfico permite comparar el tiempo promedio para conducir en el escenario del usuario con respecto al resto de usuarios de la plataforma. En él, el usuario tiene la posibilidad de filtrar las distintas simulaciones realizadas por el tipo de vehículo con el que se hicieron utilizando el selector “Select vehicle type”, como en el caso del gráfico anterior de este selector ya se ha hablado con anterioridad.

En el eje horizontal del gráfico aparecen las fechas en las que se realizaron las simulaciones en este escenario para que el usuario pueda comparar sus datos con los del resto de los usuarios, todo lo comentado se puede ver en la figura 200.

La parte inferior de la página cuenta con un gráfico circular y una tabla con los distintos escenarios en los que se pueden realizar simulaciones. El gráfico se llama “Traffic offenses in last simulations in the scenario”. Permite comparar infracciones de tráfico en las últimas simulaciones. En él, el usuario tiene la posibilidad de filtrar las distintas simulaciones realizadas por el tipo de vehículo con el que se hicieron utilizando el selector “Select vehicle type”, se repite lo mismo que para los dos gráficos anteriores sobre este selector y su contenido ya se ha hablado anteriormente.

En el lateral derecho del gráfico aparece una leyenda de colores para ayudar al usuario a identificar las distintas infracciones por colores.

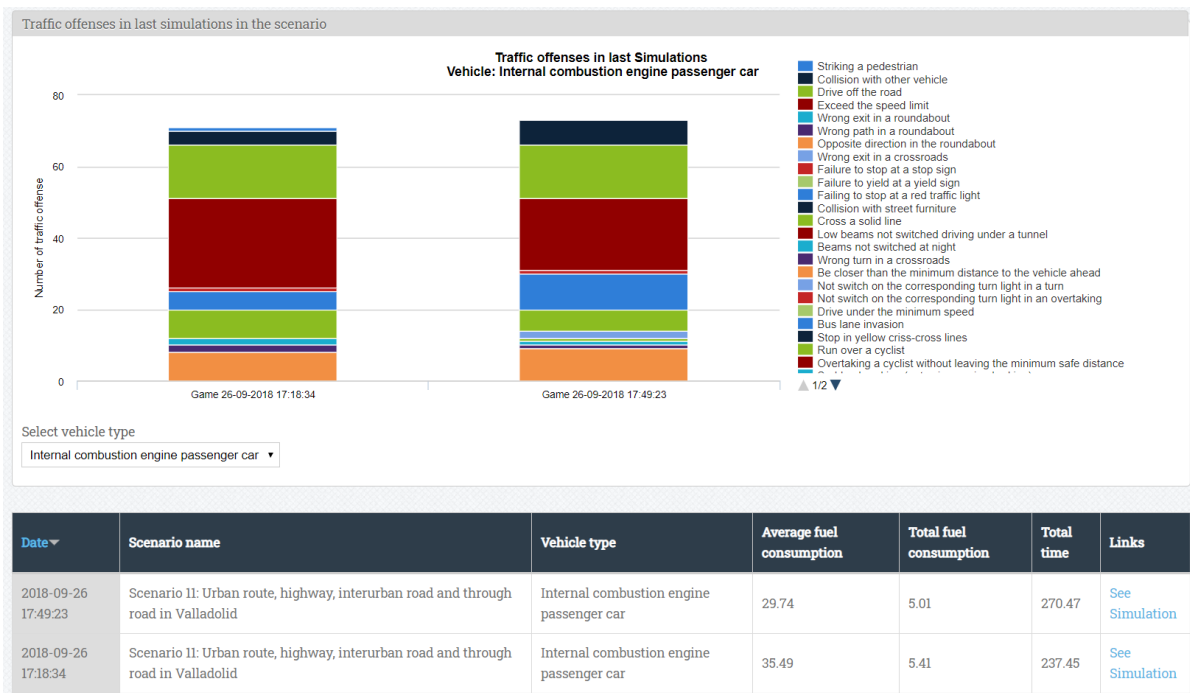


Figura 201: Parte inferior de la página “Simulation in scenario X” de la plataforma web.

En la parte final de la página existe una tabla, como se ha comentado anteriormente. En ella, vienen enumeradas las simulaciones en este escenario. Si el usuario pulsa en unos de los enlaces “See simulation” de la columna “Links” de una de las simulaciones de la tabla, el usuario accederá a la página “Simulation Data”, todo de lo que se ha hablado se puede ver en la figura 201.

En la página “Simulation Data” se muestran los datos de la simulación que se seleccionaron en la tabla de la página anterior. Esta página cuenta con una serie de tablas que facilitan el análisis de los datos de la simulación. La primera de estas tablas se llama “General simulation data”. En esta tabla, se muestran los siguientes datos de la simulación, eliminando las columnas “Consumo medio de combustible”, “RPM promedio” y “Desviación estándar de RPM” cuando la simulación se realizó con un vehículo con motor eléctrico:

- Tipo de vehículo
- Consumo medio de combustible
- Tiempo Total

- *Velocidad media*
- *Desviación estándar de velocidad*
- *RPM promedio*
- *Desviación estándar de RPM*

La siguiente tabla que aparece en la página se llama “Average RPMs per gear”. Muestra las RPM promedio por marcha y solo es visible para el usuario cuando el vehículo utilizado para realizar la simulación es de motor de combustión. Si el vehículo utilizado es de motor eléctrico, esta tabla no aparece, estas tablas se pueden ver en la figura 202.

General simulation data

Vehicle type	Average fuel consumption	Total time	Average speed	Standard deviation of speed	Average RPM	Standard deviation of RPM
Internal combustion engine passenger car	9.82 -0.47	30.69 -135.67	25.98 -6.07	12.2 -13.86	3069.73 -66.67	1014.31 -916.58

Average RPMs per gear

Gear 1	Gear 2	Gear 3	Gear 4	Gear 5
1676.99	3481.16	3111.98	0	0

Figura 202: Parte superior de la página “Simulation data” de la plataforma web.

A continuación de esta tabla, existe un selector que permite al usuario elegir qué datos se van a mostrar en la imagen del recorrido, pudiendo elegir qué información del recorrido se va a mostrar: toda la información, infracciones de seguridad o infracciones de eficiencia.

Select the data to show in the Image

Select the data to show

All data

Select the data you want to show in the image:

Compare the simulation with the simulation of another user

Select a user and a simulation

2 — Simulation: 2016-06-27 19:49:44

Select a user and a simulation to compare with the current simulation.

Comparison of the current simulation with the simulation:

- User: 1 - Date: 2016-06-24 22:57:00 - Rectangles: (Data: blue, Traffic offences-Security: red, Traffic offences-Efficiency: gray)

- User: 2 - Date: 2016-06-27 19:49:44 - Rectangles: (Data: green, Traffic offences-Security: orange, Traffic offences-Efficiency: yellow)

Figura 203: Parte intermedia de la página “Simulation data” de la plataforma web.

Debajo de la tabla “Average RPMs per gear”, existe una imagen en la que se marca el recorrido de la simulación con puntos. Estos puntos son azules para los datos del vehículo recogido de forma periódica en el recorrido, son rojos para las infracciones de seguridad cometidas por el usuario en la simulación. Las

infracciones de eficiencia en esta imagen que representan el recorrido se muestran como puntos grises. Respecto a los puntos de diferentes colores que se muestran en esta imagen, si el usuario pulsa estos puntos, se despliegan mostrando información relevante sobre ese tipo de dato o esa infracción. Cada uno de estos puntos hace referencia a un momento exacto de la simulación y a un punto exacto del recorrido de la misma, ya sean datos o infracciones. Por eso, al desplegarse las etiquetas, la información que se muestra hace referencia a ese punto exacto, a ese momento exacto, a la propia simulación que se está analizando, al vehículo con el que se realizó y al usuario que la realizó.

Situado bajo la imagen anterior, existe un selector que permite al usuario comparar su recorrido con otro usuario que haya realizado una simulación en ese mismo recorrido y con el mismo tipo de vehículo, replicando el comportamiento de los puntos en el recorrido del usuario original. Al seleccionar una simulación en el selector en la imagen, se representará el recorrido con puntos del usuario con el que se quiere comparar. Como se ha comentado, el recorrido del usuario a comparar que se pinta con puntos en el recorrido junto a los del usuario original, sigue el mismo comportamiento que los puntos que muestran el recorrido del usuario original, pero con distinto color para poder diferenciar los dos usuarios la imagen. Para el recorrido del usuario a comparar, se muestran en color verde los datos como tal del recorrido, en naranja las infracciones de seguridad y en amarillo las infracciones de eficiencia. Para aportar mayor información y que el usuario sepa qué está viendo en cada momento, junto a este selector aparece la información de las simulaciones que se están comparando cuando el usuario selecciona una simulación de otro usuario en el selector. Todo lo anteriormente comentado se observa en la figura 203.

Debajo del selector del que se ha estado hablando existen dos tablas, una para los datos y otra para las infracciones que aparecen representadas en la imagen del recorrido. La información que aparece en la tabla de los datos incluye el número de dato en función de cuándo se ha recogido en el recorrido y el instante en el que se generó este punto. Por otro lado, en la tabla de las infracciones, la información incluye el número de infracción en función del orden en el que se ha cometido en el recorrido, el nombre de la infracción, tipo de infracción y el instante en el que se produjo la infracción, las tablas de las que se ha estado hablando se pueden ver en la figura 204.

Data Number	Instant
1	0
2	0.965
3	1.9872
4	2.9863
5	3.9674
6	4.9753
7	5.9673
~	~

Traffic offenses number	Name	Type	Instant
1	Wrong turn in a crossroads	Security	24.8577
2	Not switch on the corresponding turn light in a turn	Security	24.8577

[Go Back](#)

Figura 204: Parte inferior de la página “Simulation data” de la plataforma web.

En este apartado se ha buscado explicar todas las funcionalidades características que aporta la plataforma web después de su actualización, cuando se valida en ella un usuario sin permisos de administrador de la misma.

6. Presupuesto económico

En este apartado se comentará cual sería el coste aproximado de llevar a cabo el desarrollo de este proyecto en un entorno de trabajo real, teniendo en cuenta los elementos empleados tanto hardware como software y el tiempo empleado para su desarrollo por un programador con un perfil junior. Por otro lado, también se tendrá en cuenta si resultaría rentable llevarlo a cabo para una consultoría.

A continuación, se presentará una lista con los elementos empleados y el coste que implicarían a una empresa:

- Portátil ASUS A55V => 800,00€
- Teclado y ratón Lenovo => 50,00€
- Sistema operativo Windows 8.1 => 50,00€
- Entorno de desarrollo integrado (IDE) PhpStorm => 200,00€
- Notepad++ => 0,00€
- XAMPP v3.2.2 => 0,00€
- Postman => 0,00€
- Mozilla Firefox => 0,00€
- CMS Drupal => 0,00€
- Git => 0,00€

Por otro lado, para este trabajo de fin de grado al ser un programador junior y no tener experiencia con el CMS Drupal el tiempo necesario para poder realizarlo ha sido de unas 300 horas. Teniendo en cuenta todo esto y que el salario de un programador junior ronda los 11€/h, el precio de mano de obra empleada sería superior a 3300€.

Como resumen de los costes del proyecto, estos serían de unos 4400€. Este proyecto de desarrollo de una actualización para una plataforma web sería rentable para una empresa de consultoría, si esta pudiera obtener un valor por este proyecto de entre 6000€ y 7000€. Este proyecto sería más rentable si este proyecto lo desarrollara un programador con un perfil más alto o con más experiencia con el CMS Drupal.

7. Conclusiones y líneas futuras

Este proyecto se presentó como una actualización de una plataforma web que ya existía y que actualmente se encuentra en funcionamiento, esta plataforma se desarrolló tomando como base el CMS Drupal 7 y enfocando su interfaz para el análisis de datos de distintos usuarios.

Pero en mi caso al no saber nada del CMS Drupal, trabajar sobre un desarrollo que ya está en funcionamiento y que ha hecho una persona con grandes conocimientos sobre este CMS, me supuso un gran problema ya que algo que resultaría fácil para una persona con conocimientos en este CMS a mí me resultó difícil.

Lo anteriormente planteado implicó que tuve que entender qué era Drupal, cómo funcionaba, entender el código de la plataforma actual, con las dificultades que supone trabajar y entender algo que no has hecho tú con una tecnología que no conoces. Todo este proceso me llevó mucho tiempo, más de lo que me esperaba, pero al final entendiendo el funcionamiento pude llevar a cabo, las modificaciones realizadas que buscaban permitir al usuario de la plataforma analizar las simulaciones con más detalle. Analizar qué es Drupal me llevó a entender el porqué de la elección del anterior desarrollador, ya que al disponer de conocimientos de Drupal para él le resultó fácil desarrollar la plataforma web basándose en él.

Teniendo en cuenta esto para futuras modificaciones de la misma, pienso que podía resultar interesante dejar a un lado Drupal 7 y desarrollar la plataforma web desde cero, de esta manera se le podrán aportar a la plataforma web más características y personalizar más su desarrollo y estilo, ya que Drupal restringe ciertas características que podría aportar un desarrollo desde cero sin tener que tomar como base un CMS. Por otro lado, también es reseñable que esta plataforma aportaba muchas funcionalidades para el análisis de datos. Estas funcionalidades serían más complicadas de desarrollar si se empezara desde cero el desarrollo sin utilizar Drupal como base ni ningún otro CMS. Si se planteara cambiar de CMS para una futura implementación de la plataforma web, la mejor opción desde mi punto de vista sería WordPress, ya que es un CMS muy conocido y muy utilizado actualmente, lo que implica que se dispone de mucha información si se desea implementar un desarrollo desde cero. El inconveniente es que esta plataforma se apoya en una API REST para comunicarse con el simulador de conducción. Esta API REST en Drupal resulta mucho más sencilla de implementar que en WordPress. Ese fue uno de los puntos importantes por los que se eligió Drupal cuando se comenzó el desarrollo de la plataforma en sus orígenes.

Es importante comentar que en mi caso sin conocimientos de Drupal desarrollar una plataforma con estas características hubiera sido muy complicado, por lo que trabajar sobre este desarrollo en funcionamiento por una parte también me ha facilitado el trabajo.

Otra de las posibilidades para posibles modificaciones de la plataforma es actualizarla desde Drupal 7 a una versión más moderna como Drupal 8, ya que puede aportar muchas características que la plataforma web actualmente no dispone.

Estas dos posibilidades se plantearon, pero debido a mi falta de conocimiento de Drupal parecía muy complicado migrar la plataforma de Drupal 7 a Drupal 8, sin conocer el código por no haberlo hecho desde cero. La otra opción, que era realizar un desarrollo de la plataforma sin tomar como base un CMS podía haber sido factible, pero lo que

hubiera sido más complicado sería implementar una funcionalidad similar a la API REST de la que dispone Drupal.

Una vez comentadas las posibles futuras implementaciones tecnológicas de la plataforma web, es interesante comentar las posibles nuevas funcionalidades que se podrían incluir en futuras actualizaciones de la misma. En la página que ofrece la posibilidad de comparar recorridos, se podría incluir una gráfica en la que se comparan las infracciones cometidas por los dos usuarios. En esta página también se podría incluir una gráfica que permitiera comparar los tiempos de ambos usuarios al pasar por ciertos puntos del recorrido y una tabla dentro de esta página que permitiera comparar consumos y relación de marchas de los dos usuarios en ese recorrido.

Dejando a un lado las futuras modificaciones tecnológicas y funcionales de esta plataforma, con esta actualización se ha buscado ayudar a los usuarios a poder diferenciar con qué tipo de vehículos han realizado las simulaciones y permitir analizar los datos de las simulaciones diferenciándolas por el tipo de vehículo con el que se han realizado. Se ha aportado una funcionalidad completamente nueva que permite al usuario comparar el recorrido realizado en un escenario con el de otro usuario en función del vehículo utilizado. También, se ha adecuado el contenido de la plataforma para que sea más fácil de comprender para los usuarios.

Este proyecto me ha servido para obtener conocimientos sobre Drupal y sobre las posibilidades que existen para desarrollar una plataforma web, como los CMS y los *frameworks*. Toda esta información me resultara muy interesante para mi futuro laboral ya que me aporta nociones básicas para el desarrollo de plataformas webs.

8. Bibliografía

- Ramón Javier Corchero (2016) Desarrollo de escenarios virtuales para simulación de conducción. Trabajo de fin de grado – Grado en ingeniería de tecnologías de telecomunicación.
- PhpStorm: El IDE de PHP Lightning-Smart. Accesible en: <https://www.jetbrains.com/phpstorm/>. Última visita, diciembre 2019.
- Notepad++: What is Notepad++. Accesible en: <https://notepad-plus-plus.org/>. Última visita, diciembre 2019.
- XAMPP: XAMPP Apache + MariaDB + PHP + Perl. Accesible en: <https://www.apachefriends.org/es/index.html>. Última visita, diciembre 2019.
- Postman: The Collaboration Platform for API Development. Accesible en: <https://www.getpostman.com/>. Última visita, diciembre 2019.
- Mozilla Firefox: El navegador tan solo es el principio. Accesible en: <https://www.mozilla.org/es-ES/firefox/>. Última visita, diciembre 2019.
- Cuatro: ¿Se puede aprender a conducir con un videojuego? Las autoescuelas se apuntan a la moda de los simuladores. Accesible en: https://www.cuatro.com/blogs/vig/aprender-conducir-videojuego-simulador-autoescuela_6_1952565005.html. Última visita, diciembre 2019.
- DGT: De la pantalla a la realidad. Accesible en: <http://revista.dgt.es/es/reportajes/2019/05MAYO/0523Simuladores.shtml#.XYUcZFUza1t>. Última visita, diciembre 2019.
- Indra - a: ENTAC - SIMULADORES DE ALTAS PRESTACIONES. Accesible en: https://www.indracompany.com/sites/default/files/triptico_entac_simulador_conduccion_vehiculos_pesados_v01_baja_esp_.pdf. Última visita, diciembre 2019.
- Indra - b: ENTAC - Simuladores de altas prestaciones. Accesible en: https://www.indracompany.com/sites/default/files/indra_entac_simulador_conduccion_vehiculos_pesados_es_2019.pdf. Última visita, diciembre 2019.
- LANDER - a: Simbio - Simulador de Vehículos Pesados. Accesible en: <http://www.landertsimulation.com/es/simuladores/simuladores-automocion/simulador-vehiculos-pesados-simbio>. Última visita, diciembre 2019.
- TECNOTRON – a: Gold Truck - Simulador de camión. Accesible en: <http://tecnotron.devdinamiq.com/simuladores-ampliados-gold-truck.html>. Última visita, enero 2020.
- TECNOTRON – b: Gold Bus - Simulador de autobús. Accesible en: <http://tecnotron.devdinamiq.com/simuladores-ampliados-gold-bus.html>. Última visita, enero 2020.
- Indra – c: SMARTSIM - SIMULADOR DE CONDUCCIÓN DE AUTOMÓVILES. Accesible en:

https://www.indracompany.com/sites/default/files/triptico_smartsim_simulador_conduccion_automovil_v0716_baja_esp_.pdf. Última visita, diciembre 2019.

- TECNOTRON – c: Silver - Simulador de automóvil de ALTA inmersión. Accesible en: <http://tecnotron.devdinamiq.com/simuladores-ampliados-silver.html>. Última visita, enero 2020.
- LANDER – b: Simulador de Coche - La solución para vehículos ligeros. Accesible en: <http://www.landersistimulation.com/es/simuladores/simuladores-automocion/simulador-coche>. Última visita, diciembre 2019.
- TECNOTRON – d: Ambar - Simulador de automóvil de MEDIA inmersión. Accesible en: <http://tecnotron.devdinamiq.com/simuladores-ampliados-ambar.html>. Última visita, enero 2020.
- DriveSim: Simulador de conducción. Accesible en: <https://drivesimsimulator.com/caracteristicas-drivesim/>. Última visita, diciembre 2019.
- Virtual Driver Interactive: Virtual Driver Interactive. Accesible en: <https://www.driverinteractive.com/#>. Última visita, diciembre 2019.
- City Car Driving: City Car Driving - Car Driving Simulator, PC Game. Accesible en: <https://citycardriving.com/products/citycardriving>. Última visita, diciembre 2019.
- Blog de la Auto escuela: Driver Test un simulador de conducción para casa. Accesible en: <http://www.blogdelaautoescuela.com/blog/driver-test-un-simulador-de-conduccion-para-casa/>. Última visita, diciembre 2019.
- Google Play - a: Car Driving School Simulator. Accesible en: <https://play.google.com/store/apps/details?id=com.boombitgames.DrivingSchoolParking&hl=es>. Última visita, diciembre 2019.
- Google Play - b: Dr. Driving 2. Accesible en: <https://play.google.com/store/apps/details?id=com.ansangha.drjb&hl=es>. Última visita, diciembre 2019.
- MDM web docs: Frameworks Web de lado servidor. Accesible en: https://developer.mozilla.org/es/docs/Learn/Server-side/Primeros_pasos/Web_frameworks. Última visita, diciembre 2019.
- ZendFramework: Zend Framework transitions to the Linux Foundation. Accesible en: <https://framework.zend.com/>. Última visita, diciembre 2019.
- Universidad de Alicante: Modelo vista controlador (MVC). Accesible en: <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>. Última visita, enero 2020.
- Fuel PHP: The Fuel PHP Framework is a fast, simple and flexible PHP 5.4+ framework, born from the best ideas of other frameworks, with a fresh start! Accesible en: <https://fuelphp.com/>. Última visita, diciembre 2019.
- Slim: Slim a micro framework for PHP. Accesible en: <http://www.slimframework.com/>. Última visita, diciembre 2019.

- WordPress.org: Meet WordPress. Accesible en: <https://wordpress.org/>. Última visita, diciembre 2019.
- Joomla!: The Flexible Platform Empowering Website Creators. Accesible en: <https://www.joomla.org>. Última visita, diciembre 2019.
- Christophe Aubry. (2012) Drupal 7 crear y administrar sus sitios web (2012) ENI. Barcelona
- Drupal - a: We're the leading open-source CMS for ambitious digital experiences that reach your audience across multiple channels. Accesible en: <https://www.drupal.org/home>. Última visita, diciembre 2019.
- Drupal - b: Friendly and powerful: Drupal 7. Accesible en: <https://www.drupal.org/drupal-7.0>. Última visita, diciembre 2019.