



Smart Software Solutions

EQUIPO DEL PROYECTO:

ALBERTO GARCÍA HERNÁNDEZ

JUAN ABASCAL SÁNCHEZ

CARLOS OLIVARES SÁNCHEZ MANJAVACAS

DANIEL GONZÁLEZ DE LA HERA

CARLOS TORMO SÁNCHEZ

Grupo 1

DOBLE GRADO EN: INGENIERÍA INFORMÁTICA Y ADE

DIRECCIÓN DE PROYECTOS DEL DESARROLLO DEL SOFTWARE

Índice general

Índice general	1
1 Modelo Conceptual	3
I. Modelo de Clases	4
Índice de figuras	23
Índice de cuadros	23
Bibliografía	25

MODELO CONCEPTUAL

I

MODELO DE CLASES

En este documento se presenta el diagrama de clases generado sobre el sistema de seguridad al volante que está siendo desarrollado para CARSAFETY. El diagrama, representado con notación UML (Unified Modeling Language), muestra la estructura del sistema, especificando sus diferentes clases, atributos y métodos.

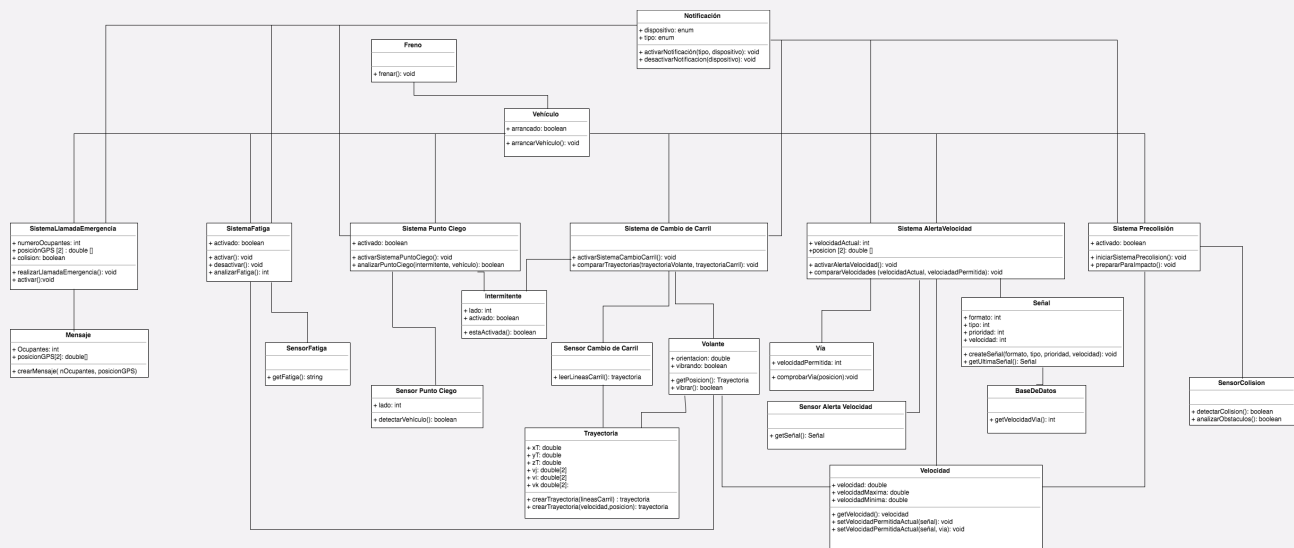


Figura 1.1: Modelo de Clases

Debido a la extensión del diagrama, hemos procedido a subdividirlo en función de los subsistemas involucrados. Todos los subsistemas se relacionan con las clases Vehículo [1.2](#) y Notificación [1.23](#).

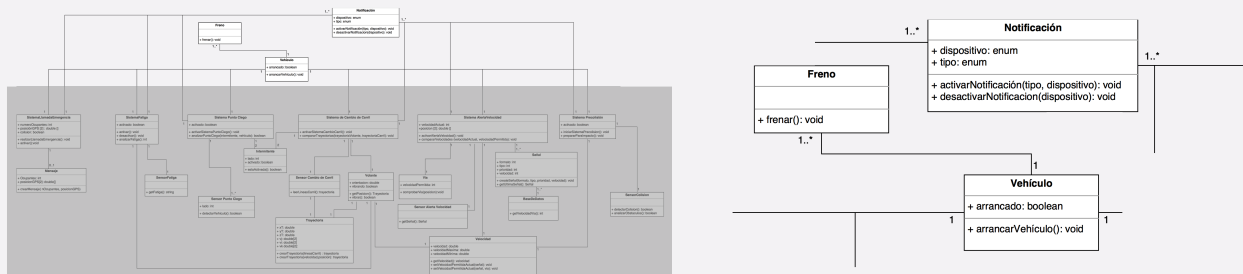


Figura 1.2: Clases Vehículo, Notificación

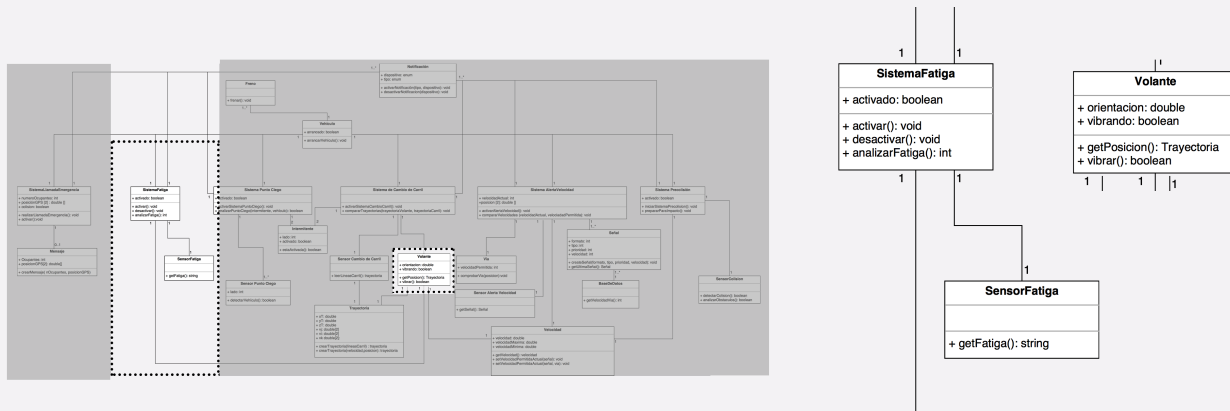


Figura 1.3: Clases pertenecientes al subsistema SistemaFatiga

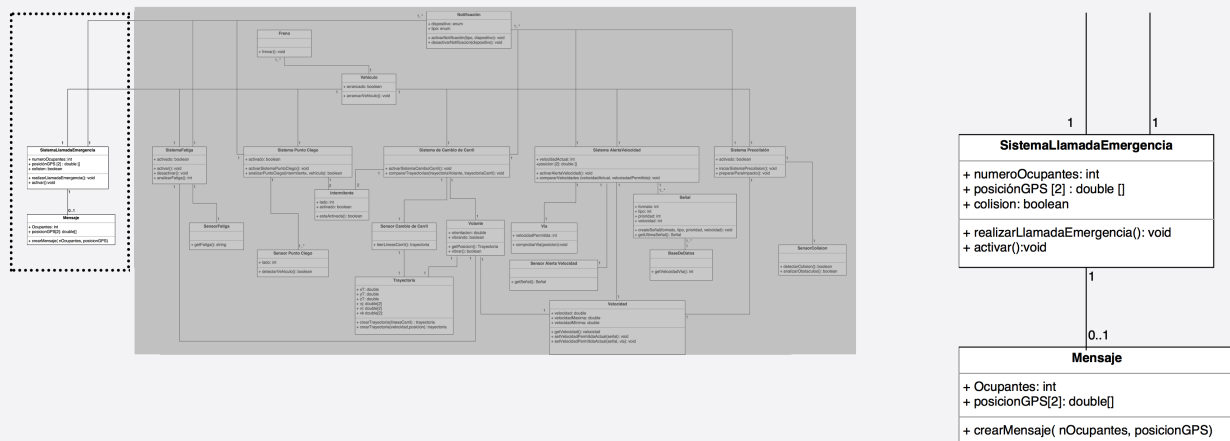


Figura 1.4: Clases pertenecientes al subsistema SistemaLlamadaEmergencia

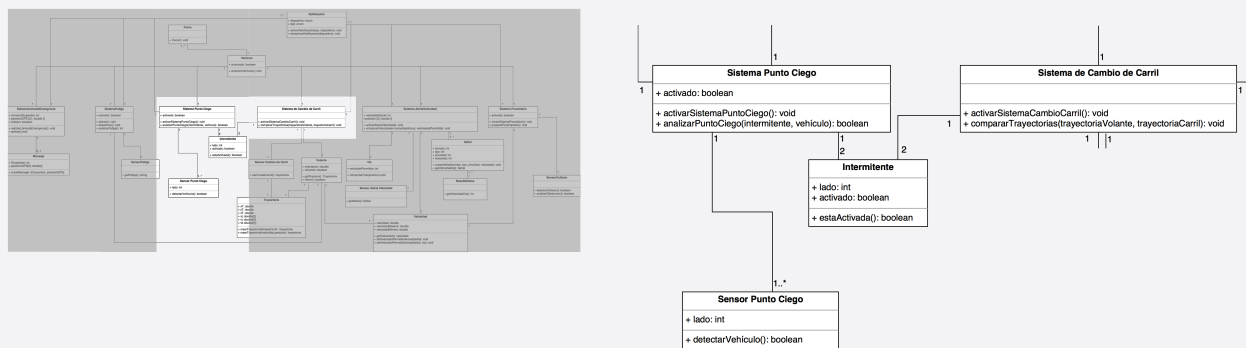


Figura 1.5: Clases pertenecientes al subsistema SistemaPuntoCiego

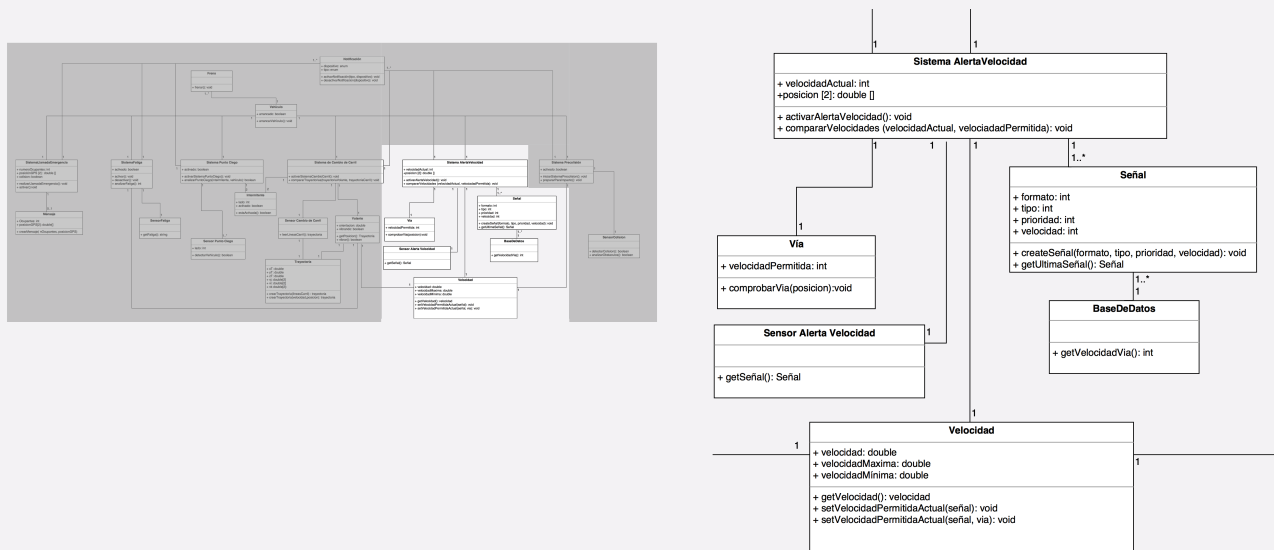


Figura 1.6: Clases pertenecientes al subsistema SistemaVelocidad

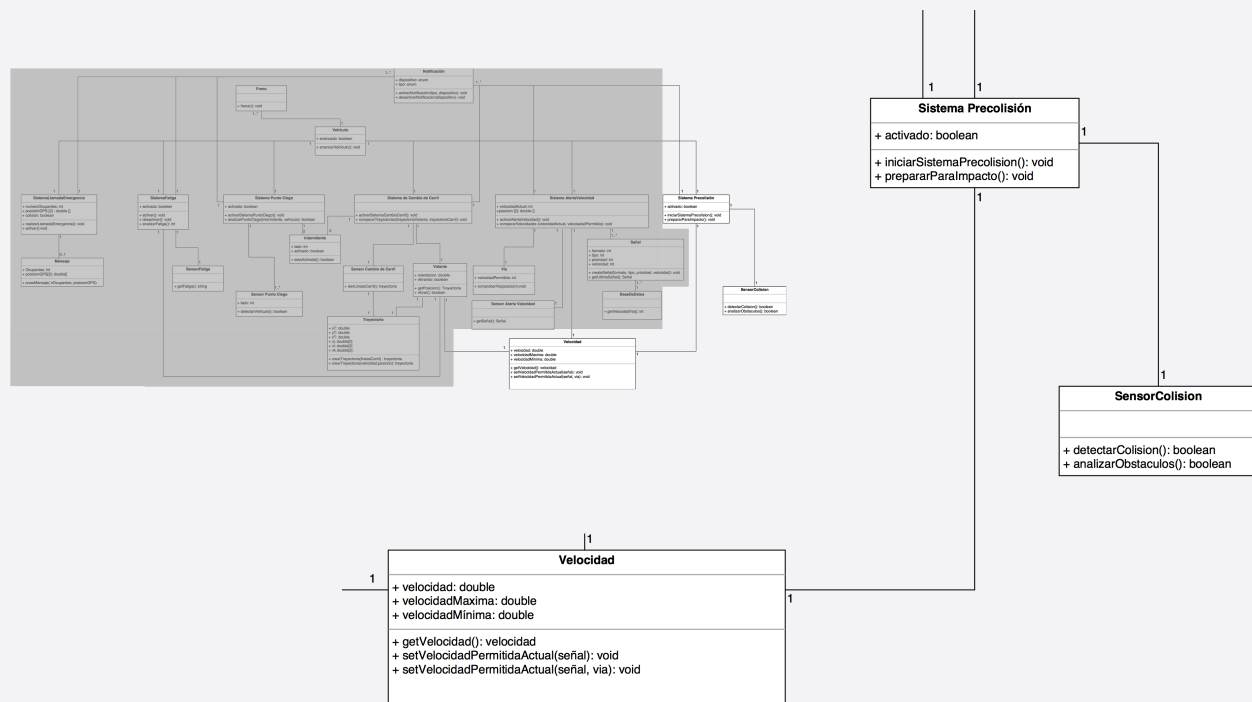


Figura 1.7: Clases pertenecientes al subsistema SistemaPrecolision

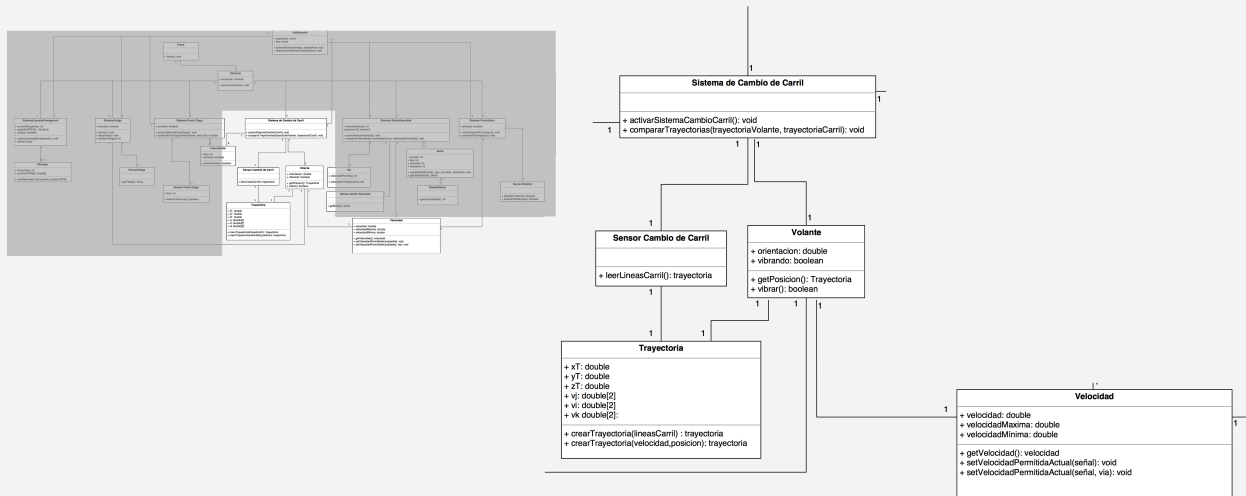


Figura 1.8: Clases pertenecientes al subsistema SistemaCambioCarril

A continuación, se procede a explicar la funcionalidad de cada una de las clases representadas en el diagrama, junto con sus atributos y métodos asociados. Para ello se utilizará una tabla como la que se presenta a continuación:

Nombre de la clase
Responsabilidad
Atributos
Relaciones
Métodos

Cuadro 1.1: Formato de las tablas de las clases del modelo conceptual.

La explicación de cada uno de los campos analizados por la tabla 1.1 es el siguiente:

- **Nombre de la clase:** nombre escogido que identifica inequívocamente la clase referenciada.
- **Responsabilidad:** descripción detallada de las acciones y responsabilidades que recaen sobre la clase.
- **Atributos:** enumeración de todos los atributos contenidos en la clase y que son necesarios para llevar a cabo las responsabilidades asociadas a la misma.
- **Relaciones:** relaciones con otras clases, ya sea de composición, agregación, asociación, etc.
- **Métodos:** métodos que se encuentran en la clase y que son necesarios para llevar a cabo las responsabilidades asociadas a la misma.

Vehículo

Responsabilidad	Encargada de arrancar todos los subsistemas de control y de gestionar la frenada del vehículo en caso de emergencia
------------------------	---

Atributos

- arrancado: boolean

Relaciones

Tiene una relación de composición con las siguientes clases:

- SistemaPuntoCiego [1.3](#)
- SistemaCambioCarril [1.6](#)
- SistemaFatiga [1.18](#)
- SistemaLlamadaEmergencia [1.20](#)
- SistemaPreColisión [1.16](#)
- SistemaAlertaVelocidad [1.10](#)
- Freno [1.22](#)

Métodos

- arrancarVehiculo(): void
-

Cuadro 1.2: Clase Vehículo

SistemaPuntoCiego

Responsabilidad	Sistema encargado de procesar los datos recibidos por los diversos sensores que lo componen así como de decidir cuando se debe activar algún tipo de notificación para hacer saber al conductor que hay un objeto en el punto ciego del vehículo
------------------------	--

Atributos

- activado: boolean

Relaciones

Tiene una relación de composición con las siguientes clases:

- SensorPuntoCiego [1.4](#)
- Vehículo [1.2](#)

Tiene una relación de agregación con las siguientes clases:

- Intermitente [1.5](#)

Tiene una relación de asociación con las siguientes clases:

- Notificación [1.23](#)

Métodos

- activar(): void
- desactivar(): void
- analizarPuntoCiego(intermitente, vehiculo): boolean

Cuadro 1.3: Clase SistemaPuntoCiego

SensorPuntoCiego

Responsabilidad	Conjunto de sensores que detectan cuándo un objeto se encuentra situado en la zona del punto ciego del vehículo
------------------------	---

Atributos

- lado: int

Relaciones

Tiene una relación de composición con las siguientes clases:

- SistemaPuntoCiego [1.3](#)

Métodos

- detectarVehiculo(): boolean

Cuadro 1.4: Clase SensorPuntoCiego

Intermitente

Responsabilidad	Conjunto de intermitentes físicos del vehículo
------------------------	--

Atributos

- lado: int
- activado: boolean

Relaciones

Tiene una relación de agregación con las siguientes clases:

- SistemaPuntoCiego [1.3](#)
- SistemaCambioCarril [1.6](#)

Métodos

- estaActivada(): boolean
-

Cuadro 1.5: Clase Intermitente

SistemaCambioCarril

Responsabilidad	Sistema encargado de procesar los datos recibidos por los diversos sensores que lo componen para poder calcular la desviación respecto al carril que va a seguir el vehículo, así como hacer vibrar o mover el volante para permanecer en el carril por el se circula y decidir cuando se debe activar algún tipo de notificación para hacer saber al conductor del vehículo que se está saliendo del carril.
Atributos	No contiene atributos
Relaciones	<p>Tiene una relación de composición con las siguientes clases:</p> <ul style="list-style-type: none"> ■ Vehículo 1.2 ■ SensorCambioCarril 1.7 <p>Tiene una relación de agregación con las siguientes clases:</p> <ul style="list-style-type: none"> ■ Volante 1.9 ■ Intermitente 1.5 <p>Tiene una relación de asociación con las siguientes clases:</p> <ul style="list-style-type: none"> ■ Notificación 1.23
Métodos	<ul style="list-style-type: none"> ■ activar(): void ■ desactivar(): void ■ compararTrayectorias(trayectoriaVolante, trayectoriaCarril): void

Cuadro 1.6: Clase SistemaCambioCarril

SensorCambioCarril

Responsabilidad	Conjunto de sensores que detectan la trayectoria del carril por el que se encuentra circulando el vehículo.
Atributos	No contiene atributos
Relaciones	<p>Tiene una relación de composición con las siguientes clases:</p> <ul style="list-style-type: none"> ■ SistemaCambioCarril 1.6 <p>Tiene una relación de agregación con las siguientes clases:</p> <ul style="list-style-type: none"> ■ Trayectoria 1.8
Métodos	<ul style="list-style-type: none"> ■ leerLineasCarril(): trayectoria

Cuadro 1.7: Clase SensorCambioCarril

Trayectoria	
Responsabilidad	Trayectoria detectada
Atributos	<ul style="list-style-type: none">▪ xT: double▪ yT: double▪ zT: double▪ vi: double[2]▪ vj: double[2]▪ vk: double[2]
Relaciones	Tiene una relación de agregación con las siguientes clases: <ul style="list-style-type: none">▪ SensorCambioCarril 1.7▪ Volante 1.9
Métodos	<ul style="list-style-type: none">▪ generarTrayectoria(lineasCarril): trayectoria▪ generarTrayectoria(velocidad, posicion): trayectoria

Cuadro 1.8: Clase Trayectoria

Volante

Responsabilidad	Clase que recopila información sobre la posición del volante y con la capacidad de hacer vibrar al mismo.
------------------------	---

Atributos

- orientacion: double
- vibrando: boolean

Relaciones

Tiene una relación de composición con las siguientes clases:

- SistemaCambioCarril [1.6](#)

Tiene una relación de agregación con las siguientes clases:

- Velocidad [1.15](#)
- Trayectoria [1.8](#)

Tiene una relación de asociación con las siguientes clases:

- SistemaFatiga [1.18](#)

Métodos

- getPosicion(): trayectoria
 - vibrar(): boolean
-

Cuadro 1.9: Clase Volante

SistemaAlertaVelocidad

Responsabilidad	Sistema encargado de gestionar la velocidad máxima para la vía por la que el vehículo se encuentra circulando, en todo momento.
------------------------	---

Atributos

- velocidadActual: int
- posicion[2]: double []

Relaciones

Tiene una relación de composición con las siguientes clases:

- SensorAlertaVelocidad [1.13](#)
- Vía [1.11](#)
- Vehículo [1.2](#)

Tiene una relación de agregación con las siguientes clases:

- Señal [1.14](#)
- Velocidad [1.15](#)

Tiene una relación de asociación con las siguientes clases:

- Notificación [1.23](#)

Métodos

- activar(): void
 - desactivar(): void
 - compararVelocidades(velocidadActual, velocidadPermitida): void
-

Cuadro 1.10: Clase SistemaAlertaVelocidad

Vía	
Responsabilidad	Clase que debe comprobar en qué tipo de vía se encuentra el vehículo y determinar de esta forma la velocidad máxima permitida.
Atributos	<ul style="list-style-type: none">■ velocidadPermitida: int
Relaciones	Tiene una relación de composición con las siguientes clases: <ul style="list-style-type: none">■ SistemaAlertaVelocidad 1.10■ BaseDeDatos 1.12
Métodos	<ul style="list-style-type: none">■ comprobarVia(posición): void

Cuadro 1.11: Clase Vía

BaseDeDatos	
Responsabilidad	Clase encargada de almacenar toda la información referente a los tipos de vía existentes y las velocidades máximas a las que está permitido circular en las mismas.
Atributos	Esta clase no tiene atributos
Relaciones	Tiene una relación de composición con las siguientes clases: <ul style="list-style-type: none">■ Vía 1.11
Métodos	<ul style="list-style-type: none">■ getVelocidadVia(): int

Cuadro 1.12: Clase BaseDeDatos

SensorAlertaVelocidad

Responsabilidad	Clase encargada de detectar las señales de trafico.
Atributos	Esta clase no tiene atributos
Relaciones	<p>Tiene una relación de composición con las siguientes clases:</p> <ul style="list-style-type: none"> ■ SistemaAlertaVelocidad 1.10 <p>Tiene una relación de agregación con las siguientes clases:</p> <ul style="list-style-type: none"> ■ Señal 1.14
Métodos	<ul style="list-style-type: none"> ■ getSeñal(): señal

Cuadro 1.13: Clase SensorAlertaVelocidad

Señal

Responsabilidad	Clase que crea y almacena la informacion obtenida por una lectura de una señal de tráfico.
Atributos	<ul style="list-style-type: none"> ■ formato: int ■ tipo: int ■ prioridad: int ■ velocidad: int
Relaciones	<p>Tiene una relación de agregación con las siguientes clases:</p> <ul style="list-style-type: none"> ■ SensorAlertaVelocidad 1.13 ■ SistemaAlertaVelocidad 1.10
Métodos	<ul style="list-style-type: none"> ■ createSeñal(formato, tipo, prioridad, velocidad): void ■ getUltimaSeñal(): señal

Cuadro 1.14: Clase Señal

Velocidad

Responsabilidad	Clase encargada de obtener la velocidad a la que circula el vehículo y de fijar la velocidad máxima a la que se puede circular en función de una señal o del tipo de vía en la que se encuentre circulando el vehículo.
------------------------	---

Atributos

- velocidad: double
 - velocidadMáxima: double
 - velocidadMínima: double
-

Relaciones

Tiene una relación de agregación con las siguientes clases:

- Volante [1.9](#)
 - SistemaAlertaVelocidad [1.10](#)
 - SistemaPreColisión [1.16](#)
-

Métodos

- getVelocidad(): velocidad
 - setVelocidadPermitidaActual(señal): void
 - setVelocidadPermitidaActual(señal, via): void
-

Cuadro 1.15: Clase Velocidad

SistemaPreColisión

Responsabilidad	Sistema encargado de detectar cuando el vehículo se encuentra en riesgo de sufrir una colisión, solicitar la activación de una notificación y preparar el vehículo para el impacto.
Atributos	<ul style="list-style-type: none">■ activado: boolean
Relaciones	<p>Tiene una relación de composición con las siguientes clases:</p> <ul style="list-style-type: none">■ Vehículo 1.2■ SensorColisión 1.17 <p>Tiene una relación de agregación con las siguientes clases:</p> <ul style="list-style-type: none">■ Velocidad 1.15 <p>Tiene una relación de asociación con las siguientes clases:</p> <ul style="list-style-type: none">■ Notificación 1.23
Métodos	<ul style="list-style-type: none">■ activar(): void■ desactivar(): void■ prepararParaImpacto(): void

Cuadro 1.16: Clase SistemaPreColisión

SensorColisión

Responsabilidad	Conjunto de sensores que deben detectar cuando hay un objeto en la trayectoria del vehículo que sea susceptible de generar una colisión
Atributos	Esta clase no contiene atributos
Relaciones	<p>Tiene una relación de composición con las siguientes clases:</p> <ul style="list-style-type: none">■ SistemaPreColisión 1.16
Métodos	<ul style="list-style-type: none">■ detectarColisión(): void■ analizarObstaculos(): void

Cuadro 1.17: Clase SensorColisión

SistemaFatiga

Responsabilidad	Sistema encargado de analizar la situación física en la que se encuentra el conductor del vehículo. Deberá analizar la posición de los parpados la presión ejercida sobre el volante, tomando las acciones oportunas en caso de emergencia.
------------------------	---

Atributos

- activado: boolean

Relaciones

Tiene una relación de composición con las siguientes clases:

- Vehículo [1.2](#)
- SensorFatiga [1.19](#)

Tiene una relación de agregación con las siguientes clases:

- Volante [1.9](#)

Tiene una relación de asociación con las siguientes clases:

- Notificación [1.23](#)

Métodos

- activar(): void
- desactivar(): void
- analizarFatiga(): void

Cuadro 1.18: Clase SistemaFatiga

SensorFatiga

Responsabilidad	Conjunto de sensores que deben detectar cuando el conductor del vehículo se está quedando o dormido o ha perdido la concentración en la carretera.
------------------------	--

Atributos	Esta clase no contiene atributos
------------------	----------------------------------

Relaciones	Tiene una relación de composición con las siguientes clases: <ul style="list-style-type: none">■ SistemaPreColisión 1.18
-------------------	--

Métodos

- getFatiga(): string

Cuadro 1.19: Clase SensorFatiga

SistemaLlamadaEmergencia

Responsabilidad	Sistema encargado de realizar una llamada de emergencia cuando ha ocurrido un accidente.
------------------------	--

Atributos

- numeroOcupantes: int
- posicionGPS[2]: double[]
- colision: boolean

Relaciones

Tiene una relación de composición con las siguientes clases:

- Vehículo [1.2](#)
- Mensaje [1.21](#)

Tiene una relación de agregación con las siguientes clases:

- Volante [1.9](#)

Tiene una relación de asociación con las siguientes clases:

- Notificación [1.23](#)

Métodos

- activar(): void
 - desactivar(): void
 - enviarMensaje(): void
-

Cuadro 1.20: Clase SistemaLlamadaEmergencia

Mensaje

Responsabilidad	Clase encargada de generar el mensaje que posteriormente será enviado por la clase SistemaLlamadaEmergencia 1.20
Atributos	<ul style="list-style-type: none">■ Ocupantes: int■ posicionGPS[2]: double[]
Relaciones	Tiene una relación de composición con las siguientes clases: <ul style="list-style-type: none">■ SistemaLlamadaEmergencia 1.20
Métodos	<ul style="list-style-type: none">■ crearMensaje(): string

Cuadro 1.21: Clase Mensaje

Freno

Responsabilidad	Clase encargada de accionar el freno físico del vehículo en caso de emergencia
Atributos	Esta clase no contiene atributos
Relaciones	Tiene una relación de composición con las siguientes clases: <ul style="list-style-type: none">■ Vehículo 1.2
Métodos	<ul style="list-style-type: none">■ frenar(): void

Cuadro 1.22: Clase Freno

Notificación

Responsabilidad	Clase encargada de generar, activar y desactivar todas las notificaciones del sistema. Para ello necesitará saber el dispositivo que solicita la notificación y el tipo de la misma
------------------------	---

Atributos

- dispositivo: enum
- tipo: enum

Relaciones

Tiene una relación de composición con las siguientes clases:

- Vehículo [1.2](#)

Tiene una relación de asociación con las siguientes clases:

- SistemaPuntoCiego [1.3](#)
- SistemaCambioCarril [1.6](#)
- SistemaFatiga [1.18](#)
- SistemaLlamadaEmergencia [1.20](#)
- SistemaPreColisión [1.16](#)
- SistemaAlertaVelocidad [1.10](#)

Métodos

- activarNotificacion(tipo, dispositivo): void
 - desactivarNotificacion(tipo, dispositivo): void
-

Cuadro 1.23: Clase Notificación

Índice de figuras

1.1. Modelo de Clases	4
1.2. Clases Vehiculo, Notificación	4
1.3. Clases pertenecientes al subsistema SistemaFatiga	5
1.4. Clases pertenecientes al subsistema SistemaLlamadaEmergencia	5
1.5. Clases pertenecientes al subsistema SistemaPuntoCiego	5
1.6. Clases pertenecientes al subsistema SistemaVelocidad	6
1.7. Clases pertenecientes al subsistema SistemaPrecolision	6
1.8. Clases pertenecientes al subsistema SistemaCambioCarril	7

Índice de cuadros

1.1. Formato de las tablas de las clases del modelo conceptual.	7
1.2. Clase Vehículo	8
1.3. Clase SistemaPuntoCiego	9
1.4. Clase SensorPuntoCiego	9
1.5. Clase Intermitente	10
1.6. Clase SistemaCambioCarril	11
1.7. Clase SensorCambioCarril	11
1.8. Clase Trayectoria	12
1.9. Clase Volante	13
1.10. Clase SistemaAlertaVelocidad	14
1.11. Clase Vía	15
1.12. Clase BaseDeDatos	15
1.13. Clase SensorAlertaVelocidad	16
1.14. Clase Señal	16
1.15. Clase Velocidad	17
1.16. Clase SistemaPreColisión	18
1.17. Clase SensorColisión	18
1.18. Clase SistemaFatiga	19

1.19. Clase SensorFatiga	19
1.20. Clase SistemaLlamadaEmergencia	20
1.21. Clase Mensaje	21
1.22. Clase Freno	21
1.23. Clase Notificación	22

BIBLIOGRAFÍA

- [1] U. C. I. de Madrid, Revisión de Métrica Versión 3. Versión subida en www.aulaglobal.uc3m.es.
- [2] I. C. S. S. E. S. Committee and I.-S. S. Board, “Ieee recommended practice for software requirements specifications,” Institute of Electrical and Electronics Engineers, 1998.
- [3] X. F. Grau and M. I. S. Segura, “Desarrollo orientado a objetos con uml,” Facultad de Informática - UPM; Escuela Politécnica Superior - UC3M, vol. 1, 2008.