



Universidad Popular Autónoma del Estado de Puebla

Centro Interdisciplinario de Posgrados

Investigación y Consultoría

Decanatura de Posgrados en Ingeniería y Negocios

Maestría en Ingeniería de Software

“DISEÑO Y DESARROLLO DE SISTEMA DE SOFTWARE DE CONTROL,
DE MISIÓN CRÍTICA, PARA EVENTOS POST-COLISIÓN: ÁREA
SEGURIDAD AUTOMOTRIZ”

Tesis que para obtener el Grado de Maestro en Ingeniería de Software

Presenta

Genaro de la Rosa Garcia

Director de Tesis

Dr. Jorge Rafael Aguilar Cisneros



UPAEP – Secretaría General

Dirección General de Apoyos Académicos

Dirección del Centro de Recursos para el Aprendizaje y la Investigación.

Biblioteca Central - **Karol Wojtyla**

Tesis Digitales Restricciones de uso:

DERECHOS RESERVADOS ©

PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de textos, imágenes, gráficas, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente de donde la obtuvo mencionando el autor o autores involucrados en el documento.

Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

DEDICATORIA

A mis hijos Diana Luz y Genaro, recomendándoles que el estudio sea su principal motivo de superación.

A mi esposa Flor por su apoyo incondicional.

A mi mamá y mi hermana por su comprensión y cariño.

A mi tío Godo, tía Carmela y tía Everth por preocuparse y confiar en mí.

AGRADECIMIENTOS

A mi asesor Dr. Jorge Rafael Aguilar Cisneros por guiarme y por el artículo logrado mediante este trabajo de investigación.

A mi directora Ma. Luisa A. De Ita Zafra, por creer en mí y por todo su apoyo.

ÍNDICE GENERAL

RESUMEN

INTRODUCCIÓN.....	i
CAPÍTULO 1. PROPÓSITO Y ORGANIZACIÓN	1
1.1. PLANTEAMIENTO DEL PROBLEMA	1
1.2. HIPÓTESIS.....	1
1.3. PREGUNTAS DE INVESTIGACIÓN	1
1.4. PROPOSITO DE LA INVESTIGACIÓN.....	2
1.5. OBJETIVO GENERAL	2
1.6. OBJETIVOS ESPECIFICOS	2
1.7. JUSTIFICACIÓN DE LA INVESTIGACIÓN	3
1.8. ALCANCES Y LIMITACIONES.....	3
1.9. ORGANIZACIÓN DEL DOCUMENTO	4
CAPÍTULO 2. MARCO TEÓRICO Y ESTADO DEL ARTE	5
2.1. ESTADO DEL ARTE	5
2.1.1. SISTEMAS PARA LA LIBERACIÓN AUTOMÁTICA DE “SEGUROS DE NIÑOS”	5
2.1.2. APERTURA DE PUERTAS DE EMERGENCIA PARA AUTOMÓVILES.....	6
2.1.3. APARATOS DE EMERGENCIA QUE ACTIVAN Y HABILITAN SEGUROS DE PUERTAS Y SISTEMAS DE VENTANAS.	7
2.1.4. SISTEMA DE ANÁLISIS, EMERGENCIA Y ALERTA DE RESCATE PARA AUTOMÓVILES DESPUÉS DE UN CHOQUE.	8
2.1.5. SISTEMA DE SOPORTE DESPUÉS DE LA COLISIÓN	9
2.1.6. ANÁLISIS DE LOS SISTEMAS PRESENTADOS Y ASPECTOS DE MEJORA.	11
2.2. MARCO TEÓRICO.....	13
2.2.1. ESTÁNDARES DE CALIDAD ISO	13
2.2.2. MODELO DE PROCESO DE SOFTWARE.....	25
2.2.3. ARQUITECTURA DE SOFTWARE	26
2.2.4. PRUEBAS UNITARIAS.....	28
2.2.5. PRUEBAS DE INTEGRACIÓN	29

2.2.6. PRUEBAS DE SISTEMA.....	29
2.2.7. PRUEBAS DE ACEPTACIÓN	29
CAPÍTULO 3. DESARROLLO DEL PROYECTO	31
3.1. ASPECTOS TÉCNICOS	31
3.2. ANÁLISIS Y REQUERIMIENTOS.....	31
3.2.1. DOCUMENTO DE VISIÓN Y ALCANCE	32
3.2.2. PROCESO DEFINICIÓN DE REQUERIMIENTOS DE STAKEHOLDER	35
3.2.3. DOCUMENTO DE CONCEPTO.....	36
3.2.4. PROCESO ANÁLISIS DE REQUERIMIENTOS.....	39
3.2.5. MODELADO DE REQUERIMIENTOS.....	39
3.2.6. DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE SRS	41
3.3. DISEÑO	46
3.3.1. ARQUITECTURA DEL SISTEMA	46
3.3.2. DISEÑO DETALLADO.....	49
3.4. DESARROLLO.....	52
3.4.1. ELEMENTOS UTILIZADOS.....	52
3.4.2. CREACIÓN DEL SISTEMA.....	53
3.4.3. CÓDIGO FUENTE	55
3.4.4. PRUEBAS DEL SISTEMA	57
3.5. ASPECTOS ADMINISTRATIVOS DEL SISTEMA	66
3.6.1. LISTA DE TAREAS	70
3.6.2. ESTIMACIÓN DE ESFUERZO PARA LLEVAR A CABO LAS TAREAS.....	71
3.6.3. DOCUMENTO DE PLANES CONTRA REALIDAD	71
3.6.4. TIEMPOS DE DESARROLLO	72
3.6.5. CALENDARIO DE ENTREGABLES	74
CAPÍTULO 4. RESULTADOS.....	75
CAPÍTULO 5. ANÁLISIS Y DISCUSIÓN	77
5.1.CONCLUSIONES Y RECOMENDACIONES.....	77
5.2.TRABAJO A FUTURO.....	77

REFERENCIAS

ANEXOS

ANEXO A. DOCUMENTO DE VISIÓN Y ALCANCE

ANEXO B. DOCUMENTO CONCEPTO DE OPREACIONES

ANEXO C. DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS SRS

ANEXO D. ARTÍCULO PRESENTADO EN CONISOFT 2017

ANEXO E. DIAGRAMAS DE SECUENCIA

ANEXO F. PSEUDOCÓDIGO

ANEXO G. CÓDIGO

ANEXO H. PRUEBAS

ÍNDICE DE FIGURAS

Figura 1. Sistema automotriz para desbloqueo automático en bloqueo de puerta de seguridad para niños. [8]	6
Figura 2. Primero y segundo piñón. Sistema de apertura de emergencia de ventana [9].	7
Figura 3. Detección de Colisión [10]	8
Figura 4. Caja Negra [11]	9
Figura 5. Componentes principales del vehículo usados por el sistema ACN [12]	10
Figura 6. Modelo de desarrollo de software [16]	17
Figura 7. Capacidades de herramientas en el uso de guías de ISO 26262. [17]	18
Figura 8. Procesos del ciclo de vida del sistema [19]	20
Figura 9. Diagrama de Requerimientos. Detección de impacto.	40
Figura 10. Diagrama de Requerimientos. Habilitar fuente de energía secundaria.	40
Figura 11. Diagrama de Requerimientos. Activación de actuadores.	41
Figura 12. Caso de Uso 1. Evalúa colisión.	44
Figura 13. Caso de Uso 9. Bajar ventanas.	45
Figura 14. Diagrama de clases.	46
Figura 15. Diagrama de secuencia.	47
Figura 16. Diagrama de despliegue I.	48
Figura 17. Diagrama de despliegue II.	48
Figura 18. Plantilla pseudocódigo.	49
Figura 19. Pseudocódigo. Evalúa colisión.	51
Figura 20. Escenario de pruebas.	53
Figura 21. Plantilla código.	55
Figura 22. Fragmento de código. Posición del vehículo.	56
Figura 23. Plantilla pruebas.	58
Figura 24. Prueba detectar posición del vehículo, todas las llantas arriba.	59
Figura 25. Prueba detectar posición del vehículo, llantas traseras arriba.	60
Figura 26. Prueba bajar ventanas, con posición de vehículo: todas las llantas abajo.	63
Figura 27. Prueba bajar ventanas, con posición de vehículo: llantas lado piloto arriba.	65

ÍNDICE DE TABLAS

Tabla 1. ISO 26262-6: Fases en el desarrollo de Software [15].....	15
Tabla 2. Tabla de costos	69
Tabla 3. Tabla de actividades.	71
Tabla 4. Tabla de plan contra realidad.	72
Tabla 5. Tiempos de desarrollo.	74
Tabla 6. Tabla de entregables.	74

RESUMEN

Los sistemas de seguridad en el sector automotriz son muy importantes. Entre estos se encuentran: Sistema de frenos ABS, sistema de control de estabilidad, sistema de bolsas de aire, entre otros. Un grupo de estos sistemas de seguridad están orientados a eventos de colisión. Estos se activa en diferentes momentos y tienen objetivos específicos: 1) Antes de una colisión su objetivo es: prevenir impactos. 2) Durante una colisión su objetivo es: minimizar los daños físicos de los ocupantes y, 3) Después de una colisión, su objetivo es: minimizar el tiempo que transcurre entre la colisión y el momento en que los ocupantes del vehículo siniestrado reciben ayuda del exterior. Sin embargo, mientras que los sistemas de tipo 1 (pasivos) y 2 (activos) se encuentran actualmente en muchos vehículos. Algunas empresas como la Volkswagen en algunos modelos empiezan a introducir sistemas como *Automatic Post-Collision Braking*, que establecen una acción específica en el sistema de frenos después de una colisión., En este trabajo de investigación se presenta un sistema de software de control automotriz que ayude a la asistencia de las personas dentro del vehículo después de una colisión, clasificando el sistema en una categoría tipo 3.

INTRODUCCIÓN

La industria automotriz es fundamental para la economía mexicana, según los censos económicos del 2014, en México represento el 14.4% de las industrias manufactureras, la industria automotriz dio empleo a 730,923 personas. La producción de esta industria, medida a través del PIB, ha aumentado su importancia al pasar de 12.8% en el 2010 a 16.9% en 2014, impulsando este comportamiento tanto por la fabricación de automóviles y camiones, como por la fabricación de partes para vehículos automotores. [1]. En 2013 el volumen de la producción mundial de vehículos automotores fue del 87,300 unidades. Los principales países productores son China con 22,117 unidades, Estados Unidos con 11,046 unidades, Japón con 9,630 unidades, Alemania con 5,718 unidades, Corea del Sur con 4,521 unidades, México con 3,052 unidades. [2].

La cantidad impresionante de vehículos implica un alto número de accidentes, por ejemplo, tan sólo en Canadá, en el 2015, hubo 1,669 personas fallecidas, y 116,735 personas lastimadas (*Canadian Motor Vehicle Traffic Collision Statistics 2015*). [3]. En Estados Unidos en el 2016 la NHTSA (*National Highway Traffic Safety Administration*) por sus siglas en inglés estima que de enero a junio se tiene un total de 17,775 personas fallecidas en accidentes automovilísticos cifra que se vio incrementada en un 10.4% en el mismo período del año 2015 con 16,100. [4]. En México, en 2016 el banco de indicadores del INEGI reporta:

- 235,998 accidentes de tránsito terrestre por colisión de vehículo en zonas urbanas y suburbanas, en el periodo de 1997 a 2016, el más alto fue de 339,857 en 2006.
- 13,362 accidentes de tránsito terrestre por colisión con peatón en zonas urbanas y suburbanas, en el mismo periodo al indicador anterior, el más alto fue de 32,308 en 2003.
- 47,220 accidentes de tránsito terrestre por colisión con objeto fijo en zonas urbanas y suburbanas, en el mismo periodo a los indicadores anteriores, el más alto fue de 55,321 en 2007.

- 3,994 accidentes de tránsito fatales, en el mismo periodo a los indicadores anteriores, el más alto fue de 5,960 en 2009.
- 67,844 accidentes de tránsito no fatales, en el mismo periodo a los indicadores anteriores el más alto fue de 107,732 en 2007.
- 288,213 accidentes de tránsito sólo daños, en el mismo periodo a los indicadores anteriores el más alto fue de 365,357 en 2006.

Entre otros indicadores. [5].

Ante esta situación, podemos inferir que los sistemas de seguridad automotriz deberían tener una gran importancia.

En este sentido los sistemas de seguridad para los pasajeros, en el caso de colisión de vehículos, desarrollan su efecto protector sin intervención de los ocupantes. Por ello, los tensores del cinturón, *airbags* y sistemas protectores en vueltas de campana se denominan también “sistemas de protección pasiva”. [6]. Otros dispositivos de protección pasiva que comprende este tipo de seguridad son: a) apoyacabezas, b) columna de dirección articulada, c) volante con absorción de energía, d) pedal de freno retráctil, e) interior del habitáculo, f) asientos, g) depósito de combustible, h) cristales y parabrisas, i) cableado y elementos eléctricos, j) chasis y carrocería, k) silla porta bebé. Existen también los sistemas de “seguridad activa”, este tipo de seguridad intenta evitar al máximo los accidentes. La seguridad activa comprende todas las soluciones que garanticen un frenado estable y potente así como un comportamiento previsible que nos permita superar las posibles situaciones críticas. Los dispositivos que comprende son: a) sistema de suspensión, b) sistema de frenos, c) sistema de dirección, d) neumáticos. Algunos autores aparte de clasificar los sistemas de seguridad activa y pasiva, también agregan una clasificación de seguridad preventiva. La cual depende de un conjunto de soluciones técnicas como a) visibilidad, b) confort y ergonomía y c) climatización. [7].

CAPÍTULO 1. PROPÓSITO Y ORGANIZACIÓN

1.1. PLANTEAMIENTO DEL PROBLEMA

Los sistemas de seguridad mencionados en las diferentes clasificaciones tienen en común un evento importante para su interacción o activación, este evento es la colisión, por lo tanto se puede decir que los sistemas de seguridad activa interactúan antes de la colisión, evitando lo más posible este evento., Los sistemas de seguridad pasiva se activan de forma inmediata en el momento de la colisión como es el caso de las bolsas de aire y los cinturones de seguridad., Pero ¿Qué sucede después de la colisión, cuando es necesario auxiliar a los ocupantes de un vehículo siniestrado?, estos sistemas de seguridad han sido poco estudiados y es precisamente esta problemática en la que se desarrolló una estrategia de solución. Para ello, se propone un sistema que se active después de una colisión, siempre y cuando esta acción de activación sea segura. El objetivo es facilitar el acceso a profesionales en la asistencia como es el caso de personal de la Cruz Roja, Protección civil o Bomberos, entre otros, así como cualquier otra persona que quiera o pueda ayudar a los pasajeros lesionados.

1.2. HIPÓTESIS

Los Sistemas de Software de control de misión crítica para eventos Post-Colisión, en el sector automotriz, debería definir su funcionalidad tomando como referencia el estado del auto, con el fin de preservar la integridad del conductor y los pasajeros del vehículo colisionado.

1.3. PREGUNTAS DE INVESTIGACIÓN

¿Un Sistema de Software para seguridad automotriz, facilitaría el auxilio a los ocupantes de un auto siniestrado, en eventos Post-Colisión?

¿Cuáles serían las funciones de un sistema de software de seguridad automotriz que permitirían el auxilio de los ocupantes de un vehículo después de una colisión?

1.4. PROPOSITO DE LA INVESTIGACIÓN

Desarrollar un sistema de software de control automotriz enfocado en la seguridad de este sector, en específico en eventos después de una colisión, que faciliten la ayuda a los ocupantes del vehículo colisionado.

1.5. OBJETIVO GENERAL

Construcción de un sistema de software de control automotriz que facilite el auxilio a los vehículos colisionados después de que suceda un accidente, mediante el uso de un sistema de software de control de misión crítica para eventos post-colisión, con el fin de agilizar que las personas que se encuentren dentro puedan salir con facilidad y las personas que se encuentran fuera puedan auxiliar más rápido y con mayor facilidad.

1.6. OBJETIVOS ESPECIFICOS

- Identificar los escenarios que faciliten el auxilio de personas después de una colisión, mediante el análisis y desarrollo de documentos de Concepto de Operaciones, con el fin de definir los requerimientos que deben ser considerados para el diseño y desarrollo de un sistema de software de misión crítica para eventos post-colisión dentro del sector automotriz.
- Crear un prototipo del sistema de software de control de misión crítica para eventos post-colisión, tomando como base los documentos de Concepto de Operaciones y

Especificación de Requerimientos de Software, con el propósito de probar la funcionalidad identificada.

1.7. JUSTIFICACIÓN DE LA INVESTIGACIÓN

En México, el número de accidentes de tránsito terrestre fue de 255,863 en el 2015, según el banco de indicadores del INEGI [5], por lo cual los sistemas de seguridad automotriz tienen un papel muy importante y son necesarios para todas las personas que se trasladen en automóviles, ya que cualquier persona puede tener el riesgo de sufrir un accidente y necesite contar con sistemas de seguridad que faciliten la asistencia., La viabilidad de este tipo de investigaciones en la UPAEP son factibles gracias al factor humano e infraestructura.

Es posible llevar a cabo la investigación ya que la universidad cuenta personal altamente calificado en temas de investigación y asesoría, para dirigir el trabajo de tesis y con laboratorios de ingeniería donde se tiene la infraestructura necesaria para realizarla. Así como motivación, entusiasmo, deseos de aprender en el desarrollo de la investigación y tiempo disponible al desarrollo del trabajo de tesis por parte del autor y asesor. Se presentó un artículo científico alineado a esta tesis, en el quinto congreso internacional CONISOFT 2017. Independientemente de los recursos con los que cuenta la universidad, actualmente se cuenta con tecnología que permite probar mediante el uso de sensores la lógica propuesta en la investigación, mediante el uso de tarjetas de control.

1.8. ALCANCES Y LIMITACIONES

El alcance del sistema Post-Colisión será la generación de los requerimientos y diseño del sistema, para las etapas de desarrollo y pruebas, serán de modo parcial, mediante la elección

de algunos módulos, para probar cierta parte de la funcionalidad., Será trabajo a futuro los requerimientos no funcionales.

El sistema Post-Colisión no funcionará en caso que tanto la fuente primaria de alimentación como la secundaria hayan sido deshabilitadas por la colisión. Los servicios que proporcionará el sistema no podrán llevarse a cabo por daños físicos de los componentes del auto que impidan, por ejemplo, que los vidrios bajen, que los seguros se desactiven o que los cinturones de seguridad se liberen., Otra limitación es que el sistema de software de control automotriz no será desarrollado con elementos automotrices.

1.9. ORGANIZACIÓN DEL DOCUMENTO

Este documento de tesis se organiza de la siguiente manera: Resumen; Introducción; En el capítulo 1 se presenta Propósito y Organización, se menciona la problemática, objetivos generales y particulares, justificación y organización del documento; En el capítulo 2. Marco Teórico y Estado del Arte, se menciona el estado del arte y el marco teórico; En el capítulo 3. Desarrollo del Proyecto, se menciona el detalle del desarrollo del sistema de software de control automotriz; En el capítulo 4. Resultados; En el capítulo 5. Análisis y Discusión; Conclusiones y Recomendaciones; Referencias; y Anexos.

CAPÍTULO 2. MARCO TEÓRICO Y ESTADO DEL ARTE

2.1. ESTADO DEL ARTE

Se han realizado diferentes estudios, algunos para deshabilitar cierta funcionalidad de los sistemas de seguridad con los que cuentan los automóviles, otros para proponer funciones modificadas en algunos componentes, todos con el fin de ayudar a los ocupantes, pero cada uno aborda el tema de manera distinta con diferentes aciertos y diferencias.

2.1.1. SISTEMAS PARA LA LIBERACIÓN AUTOMÁTICA DE “SEGUROS DE NIÑOS”

Estos se encuentran instalados en las puertas traseras de algunos autos. Este sistema desbloquea el seguro automáticamente. Se compone de un sensor de emergencia, que puede ser el sensor de las bolsas de aire, un mecanismo de seguro de la puerta y un mecanismo para liberar el seguro de la puerta. Cuando el sensor detecta una colisión, el sensor produce una señal que es enviada al mecanismo de liberación. El mecanismo de liberación desbloquea el “seguro de niños” de la puerta [8], ver figura 1.

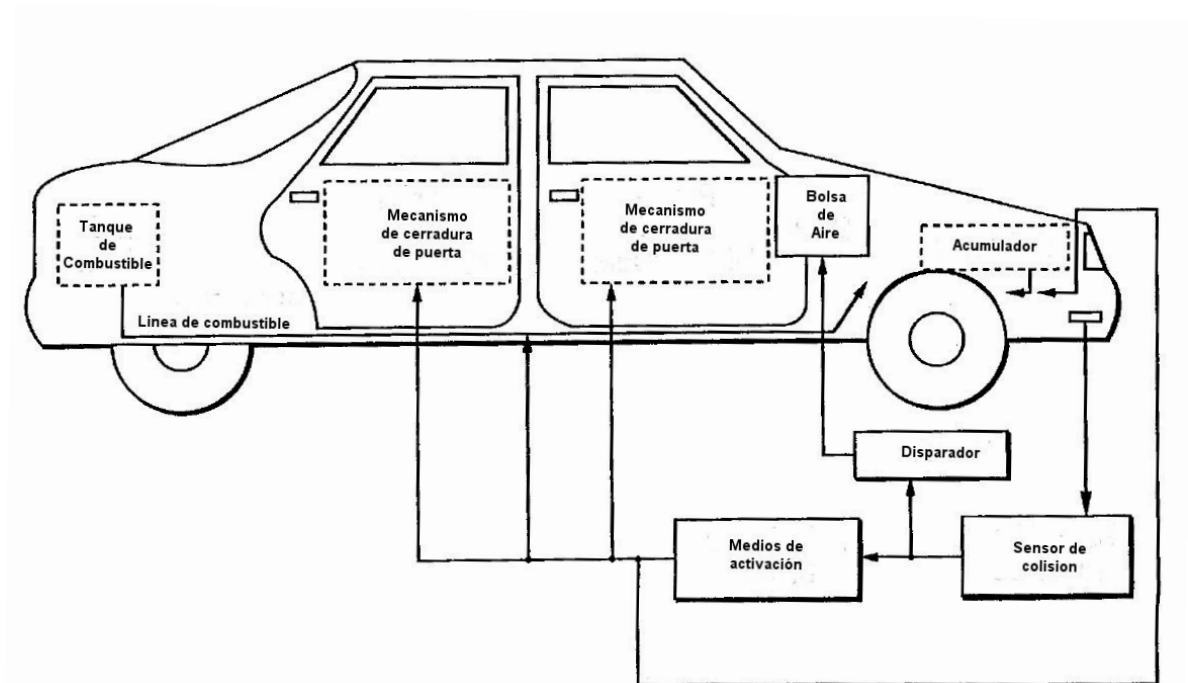


Figura 1. Sistema automotriz para desbloqueo automático en bloqueo de puerta de seguridad para niños. [8]

El sensor también puede enviar una señal para desconectar la batería del circuito eléctrico del automóvil e interrumpir la línea de alimentación de combustible del tanque hacia el motor. El sensor de emergencia, adicionalmente, puede incluir un sensor de temperatura o humo.

2.1.2. APERTURA DE PUERTAS DE EMERGENCIA PARA AUTOMÓVILES

Este Sistema de asistencia a pasajeros es cuando el automóvil llega involucrarse en un accidente y una o varias puertas quedan atoradas en el marco, la asistencia se enfoca en la apertura de emergencia de ventana, incluye una manija retráctil que se extiende para que manualmente se abra o cierre una ventana del vehículo, un primer piñón de engranaje está conectado a la manija retráctil, un segundo piñón de engranaje que se engancha con el primer piñón [9], ver figura 2.

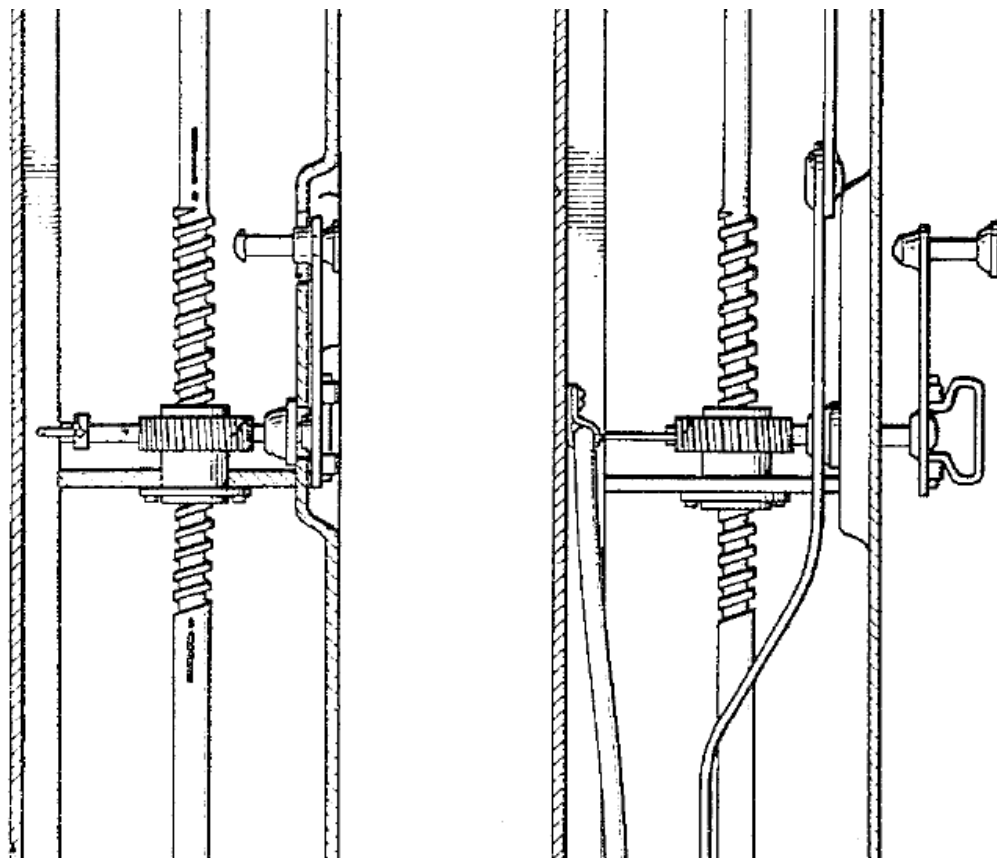


Figura 2. Primero y segundo piñón. Sistema de apertura de emergencia de ventana [9].

2.1.3. APARATOS DE EMERGENCIA QUE ACTIVAN Y HABILITAN SEGUROS DE PUERTAS Y SISTEMAS DE VENTANAS.

En un evento de choque, de manera automática e inmediata se desbloquean los seguros y se bajan las ventanas. El objetivo es que el personal médico pueda ayudar a los pasajeros sin pérdida de tiempo. Alternativamente, después de un accidente, las ventanas eléctricas del auto también podrían ser operadas mediante una manija exterior. Los vehículos automotores que tienen dispositivos que pueden ser configurados para ser activados o desactivados, tales como “seguro de niños”, son factibles de ser controlados automáticamente con el fin de permitir la liberación de puertas de un vehículo, evitando el peligro de quedar atrapado dentro de vehículos dañados[10], ver figura 3.

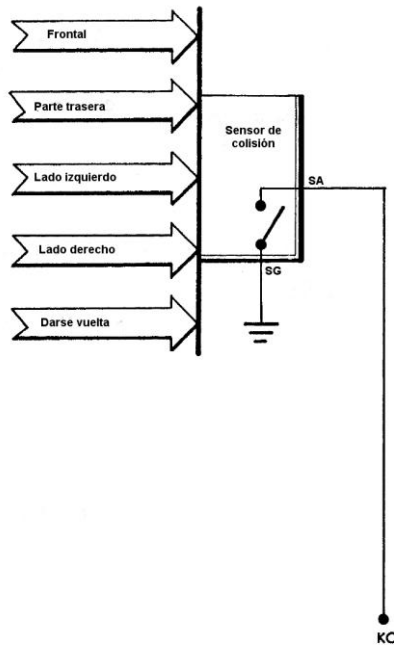


Figura 3. Detección de Colisión [10]

Algunos estudios se enfocan en sistemas que son activados después de la colisión, como es el caso de los presentados a continuación:

2.1.4. SISTEMA DE ANÁLISIS, EMERGENCIA Y ALERTA DE RESCATE PARA AUTOMÓVILES DESPUÉS DE UN CHOQUE.

Este estudio propone el uso de cajas negras, tecnología que viene de la industria de la aviación y que se quiere utilizar en el sector automotriz., Estos dispositivos servirán de análisis al momento de revisar que causas fueron factores importantes en la colisión, para esto el sistema grabará imágenes, que permitan a los investigadores conocer como estaba siendo conducido el vehículo, cuáles eran las condiciones del terreno, etc. Otra funcionalidad aparte del análisis es que también sirvan de ayuda para una rápida operación de rescate de emergencia, utilizando el GPS, enviando la ubicación del accidente, enviando mensajes de texto, y apoyándose con

equipos celulares dando aviso a los centros de emergencias, como hospitales, bomberos y policía. [11], ver figura 4.

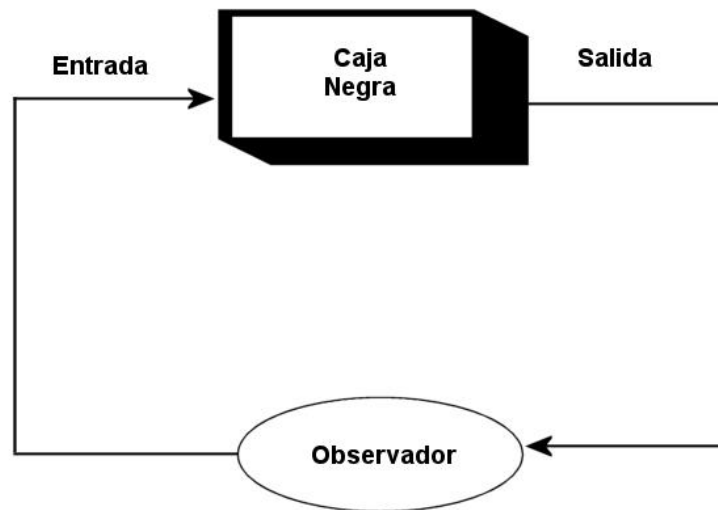


Figura 4. Caja Negra [11]

2.1.5. SISTEMA DE SOPORTE DESPUÉS DE LA COLISIÓN

También llamada Notificación Automática de Colisión (*ACN Automatic Crash Notification*) por sus siglas en ingles. Este sistema utiliza los datos de los sensores y tecnología de comunicación, realizando inmediatamente una llamada de auxilio, enviando la localización exacta y la magnitud del impacto a los centros de rescate, con las características del siniestro, para que sean enviados equipos de rescate para la asistencia. En 1995 algunos fabricantes comenzaron a equipar los vehículos con tecnología básica ACN para transmitir señales de auxilio en colisiones de moderadas a fuertes, haciendo uso de funcionalidad del airbag y enviando coordenadas exactas del GPS, estableciendo comunicación mediante el servicio de 911., En años más recientes se agregó la detección de la dirección del impacto, presencia de pasajeros, conocimiento de tres puntos de uso del cinturón de seguridad en asientos, ocurrencia de un evento de volcadura y reconocimiento de múltiples colisiones. [12], ver figura 5.

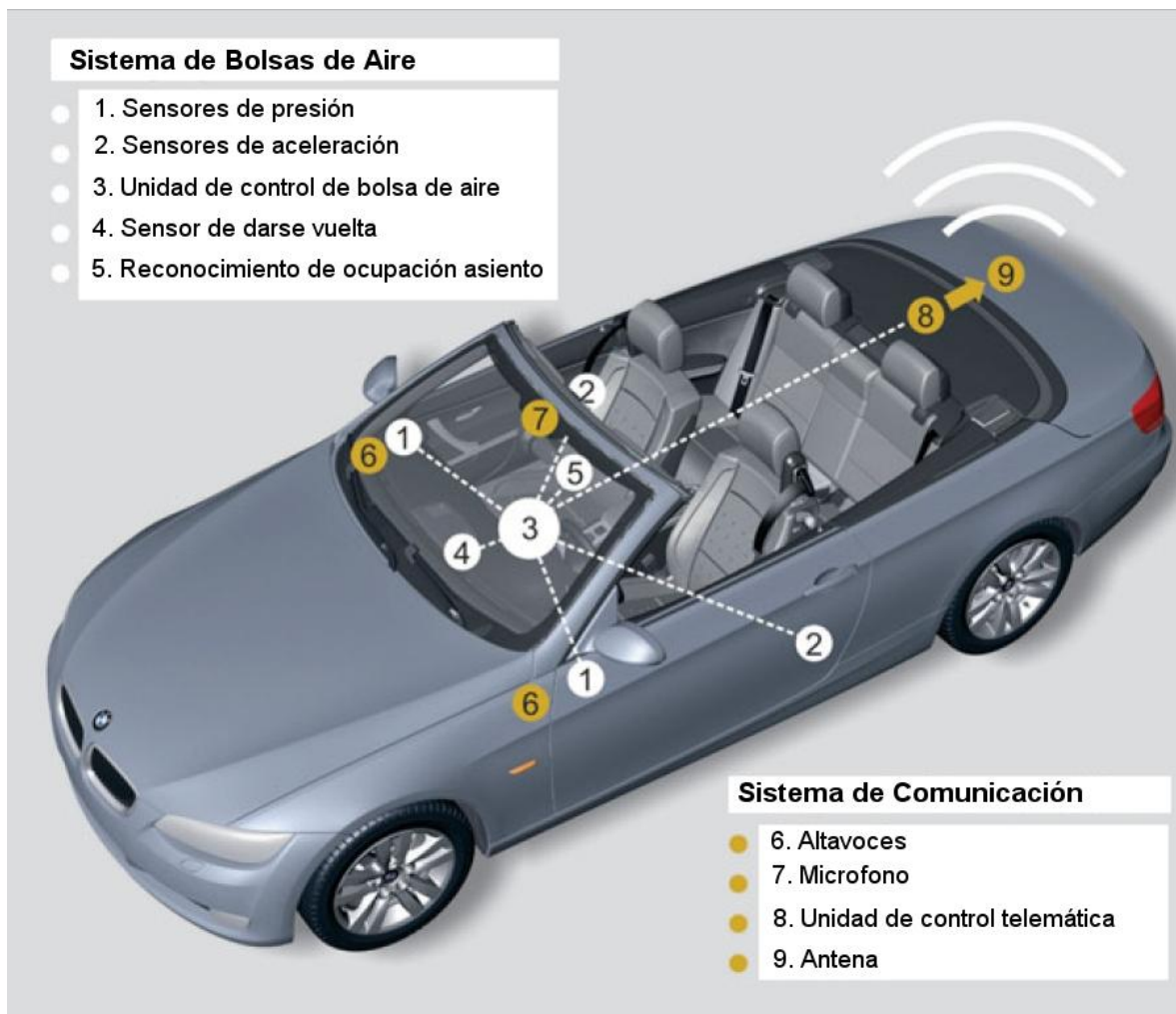


Figura 5. Componentes principales del vehículo usados por el sistema ACN [12]

En años recientes algunos modelos de automóviles incluyen de manera específica funcionalidad Post-Colisión. En marcas como Volkswagen en modelos 2016, en autos como Beetle, Jetta o Golf cuentan con un sistema de frenado Post-Colisión, para mantener seguros a los ocupantes, si es que el vehículo se ve involucrado en una colisión. Este sistema de asistencia de manejo puede ser de ayuda evitando colisiones potenciales. La forma de operar es cuando el sensor de impacto del airbag detecta una colisión, el vehículo disminuye su velocidad aplicando los frenos. [13].

La implementación e investigación de la seguridad automotriz, previniendo y reaccionando a los accidentes automovilísticos, siempre actúan en función de la integridad de los ocupantes. Cuando ocurre una colisión por alcance, los ocupantes pueden tener lesiones de cuello (whiplash). Las lesiones de los pasajeros se presentan por varios factores, entre ellos la severidad del impacto, el diseño del cinturón de seguridad, la postura del ocupante. Para disminuir la severidad de estas lesiones se han hecho estudios que muestran la importancia del uso del cinturón de seguridad, del control del asiento y cabecera que mantiene seguros a los ocupantes. En Estados Unidos en el 2014 aproximadamente el 24% de los accidentes automovilísticos fueron por alcance. [14].

2.1.6. ANÁLISIS DE LOS SISTEMAS PRESENTADOS Y ASPECTOS DE MEJORA.

Los aspectos de mejora que se encuentran en los inventos mencionados, algunos estudios dependen de la funcionalidad con la que cuentan las bolsas de aire, para dar inicio a las acciones propuestas., En específico, al momento de detectar la colisión, sin tomar en cuenta que no todos los modelos de automóviles cuentan con este sistema de seguridad.

De las propuestas analizadas, ninguna contempla el momento oportuno para que el sistema que controla las -ventanas, seguros, cinturones de seguridad, ignición del motor, entre otros-, entren en operación. Este momento debe ser determinado por el sistema con el fin de no exponer a los ocupantes del vehículo siniestrado a un riesgo mayor al que ya sufrieron por el choque. Los sistemas analizados, al no determinar este momento oportuno plantea que de manera inmediata se lleve a cabo la apertura de seguros, liberación de cinturones de seguridad, bajar las cristales de las ventanas, desbloquear el seguro de niños. Estos sistemas al operar de manera inmediata puede poner en riesgo a los pasajeros porque, pueden existir diferentes escenarios, entre otros: a) Inmediatamente después de ocurrir el accidente, el auto todavía se sigue moviendo, b) El auto puede girar y voltearse, c) Los cinturones de seguridad se liberaron y las ventanas se bajaron, por lo que los pasajeros podrían salir disparados a través de las ventanas. Dado estos escenarios, es necesario determinar la posición física final en que quedo

el automóvil y determinar el momento oportuno en el que el sistema manipulará los diversos mecanismos ya mencionados, estos son las aportaciones de este trabajo de tesis.

Ante la problemática descrita, el sistema que se propone, toma en consideración los siguientes aspectos con el fin de dar solución a los aspectos de mejora, mencionados en los párrafos anteriores:

1. Tomar en consideración la posición final del auto en el momento posterior al impacto.
2. Determinar el momento oportuno de manipulación de los diferentes elementos descritos (vidrios, seguros, cinturones, encender luz interior, habilitación de fuente secundaria de energía). El momento oportuno será cuando, por un lapso determinado de tiempo el auto no se ha movido de su posición final. En otras palabras, se asignará un lapso de tiempo desde que el carro dejó de moverse, cuando se cumpla el intervalo definido, el sistema iniciará con la manipulación de los elementos.
3. El sistema, tomando como base la posición final del vehículo siniestrado, define 8 escenarios posibles y son los siguientes: 1) las cuatro llantas del vehículo se encuentran en el piso, 2) El vehículo yace sobre su lado derecho, 3) El vehículo yace sobre su lado izquierdo, 4) Las cuatro llantas del vehículo se encuentran en el aire, 5) Las cuatro llantas del vehículo se encuentran en el aire con el cofre o frente sobre el piso, 6) Las cuatro llantas del vehículo en el aire con parte trasera del coche sobre el piso, 7) Las dos llantas delanteras del coche sobre el piso y las dos llantas traseras en el aire, 8) Las dos llantas traseras sobre el piso y las dos llantas delanteras en el aire. La razón por la cual se toma en cuenta la posición final se debe a que, en el caso que de manera automática e inmediata se bajaran las ventanas, esto podría dañar a los pasajeros. Por ejemplo, supongamos que se presenta el escenario 2) o 3), que es cuando el auto se encuentra volteado sobre uno de sus costados y se diera la situación que un pasajero estuviera en ese costado, éste podría ser lastimado si su cabeza o cuerpo estuviera apoyado sobre el cristal. Otro ejemplo sería, si el auto estuviera en el escenario 4) y, si de manera inmediata se liberaran los cinturones de

seguridad, los pasajeros sufrirían daños al no tener un mecanismo que los mantenga unidos a los asientos. Por estos motivos, el sistema que se propone, bajará las ventanas de los autos después de ser liberados los seguros pero, antes de bajar las ventanas, el sistema identificará la posición del auto y con base a esta información, se determinará que ventana o ventanas se bajarán y de la misma manera para los otros dispositivos que manipulará el sistema.

4. En caso de que el sistema se encuentre sin energía para operar, nuestro sistema lo soluciona mediante una fuente de alimentación de respaldo. Esta situación tampoco es considerada en los inventos analizados. El invento que proponemos, incluye una fuente de alimentación secundaria que proporcionará energía, durante un tiempo determinado, para que éste pueda operar.

2.2. MARCO TEÓRICO

2.2.1. *ESTÁNDARES DE CALIDAD ISO*

Existen algunos estándares de calidad, normas para desarrollar software que han sido utilizados de manera importante, como es el caso de ISO 26262 y la ISO/IEC/IEEE 15288. Algunas de estos estándares están enfocadas al sector automotriz. Aparte de los estándares de calidad algunos autores de la ingeniería de software han propuesto diferentes recomendaciones en el desarrollo de software, dado que en pocos años las diferentes tecnologías continúan en una evolución constante a una gran velocidad, hace unos pocos años el principal objetivo era las computadoras personales, ahora se tienen diferentes plataformas en la cual pueden existir desarrollo de sistemas, como servicios web, la nube, dispositivos móviles, dispositivos embebidos, cliente servidor, etc. Con sus respectivas generalidades y particularidades.

2.2.1.1. **ISO 26262**

El estándar internacional ISO 26262 publicada en noviembre de 2011, es una adaptación del IEC 61508 para la industria automotriz, este aplica para los sistemas eléctricos y/o electrónicos relacionados a la seguridad, y aborda los peligros potenciales al malfuncionamiento. El estándar por sí mismo y toda su normativa y requerimientos están principalmente basados en el riesgo., Entre más probable pueda ser un riesgo potencial, mayor deberá ser la evaluación de seguridad. Las características de seguridad deberán ser instaladas y mantenidas adecuadamente.

Las partes del estándar ISO 26262 son: 1. Vocabulario, 2. Administración de la seguridad funcional, 3. Fase conceptual, 4. Desarrollo del producto, 5. Desarrollo del producto: Desarrollo de *Hardware*, 6. Desarrollo del producto: Desarrollo de *Software*, 7. Producción y operación, 8. Procesos de soporte, 9. Orientación ASIL y Análisis Orientado a la seguridad, 10. Guías del ISO 26262. [29].

Para el sistema de software, es de especial importancia ISO 26262-6 Desarrollo de Software automotriz. Ya que ésta es la que de manera específica aborda los aspectos del desarrollo de sistemas de software para el sector automotriz. En este apartado se describen 7 fases que, con el fin de garantizar la seguridad y confiabilidad del sistema, se deben contemplar en el momento de desarrollo; ver tabla 1. A continuación se describen cada una de estas fases.

Fase	Nombre de la fase
5	Inicio del desarrollo del software
6	Especificación de requerimientos de seguridad
7	Diseño de la arquitectura
8	Diseño e implementación
9	Pruebas unitarias
10	Pruebas de integración
11	Verificación de requerimientos de seguridad

Tabla 1. ISO 26262-6: Fases en el desarrollo de Software [15]

Fase 5. Inicio del desarrollo de software. Durante esta fase, se seleccionan las herramientas y el proceso de desarrollo de software. Adicionalmente se crea un plan en el que se incluye, entre otras cosas, el modelo que se utilizará en el desarrollo de software así como los planes para verificar el desarrollo. También se define el lenguaje que se usará para el modelado y para la implementación del software y las herramientas que se utilizarán en las diferentes fases de desarrollo.

Fase 6. Especificación de requerimientos de seguridad. El objetivo principal de esta fase es definir todas las funciones que proporcionará el software y que potencialmente podrían impactar en la seguridad del sistema ya sea por: a) generar un estado inseguro, por ejemplo un módulo de software que controla la tracción en el proceso de frenado o, b) No tratar correctamente un fallo en el hardware o software. Los requerimientos, como pueden ser: aspectos de tiempo de respuesta para cada componente de software, interfaces entre componentes de software, etcétera, deben ser definidos.

Fase 7. Diseño de la arquitectura. Durante esta fase, se desarrolla un diseño de alto nivel para cada componente de software. La conformidad entre el diseño de la arquitectura y los

requerimientos de seguridad deben ser verificada. Existen requerimientos específicos para el buen diseño de la arquitectura, por ejemplo: Uso de una notación apropiada, diseño modular, categorización de componentes, entre otros.

Fase 8. Diseño e implementación. En esta fase, cada subsistema se diseña e implementa. Los requerimientos para esta fase son: Definir la notación de diseño, hacer una especificación unitaria, desarrollar una verificación unitaria. El estándar recomienda algunos métodos para cumplir con los requerimientos de la fase.

Fase 9. Pruebas unitarias. El objetivo de las pruebas unitarias es evaluar de manera individual la correcta operación de cada módulo de software. Para las pruebas unitarias se deben cumplir los siguientes requerimientos: 1. Verificar componentes de seguridad, 2. Planificación y ejecución de pruebas, 3. Métodos de pruebas unitarias, 4. Generación de pruebas, 5. Métricas de pruebas, Las pruebas unitarias son útiles por que permiten identificar y corregir problemas en el código. Para llevar a cabo las pruebas unitarias se recomienda el uso de métodos definidos en el estándar.

Fase 10. Pruebas de Integración. Aunque en las pruebas unitarias se encontrarán la mayoría de los defectos de un sistema de software, es probable que surjan algunos defectos debido a la incorrecta comunicación entre los componentes del sistema. El objetivo de las pruebas de integración es demostrar que la integración de los diferentes componentes de software trabajan adecuadamente. Los métodos recomendados por el estándar para esta fase, son los mismos que para la fase de pruebas unitarias.

Fase 11. Verificación de requerimientos de seguridad. Esta fase se lleva a cabo con el fin de asegurar que el software embebido trabaja apropiadamente en el ambiente para el cual fue

creado. Los requerimientos que deben ser cubiertos en esta fase son: 1. Planeación y ejecución de la verificación, 2. Ambientes de prueba, 3. Hardware objetivo, 4. Resultados de evaluación, [15].

2.2.1.2. Modelo en V

El proceso de desarrollo de software adopta el modelo en V, como se describe claramente en el apartado 6 de la norma ISO 26262, donde se muestra: 6-6. Especificación de los requerimientos de seguridad de software, 6-7. Diseño de la arquitectura, 6-8. Diseño de unidad de software e integración, 6-9. Pruebas de unidad de software, 6-10. Diseño de integración y pruebas, 6-11. Verificación de requerimientos seguros de software. El modelo V es ampliamente adoptado en la industria automotriz en Taiwan, *The Industrial Technology Research Institute (ITRI)*, actualmente intentan desarrollar sistemas de control de motor utilizando ISO 26262, integrando ASIL C para sistemas de propulsión eléctrica en vehículos eléctricos [16], ver figura 6.

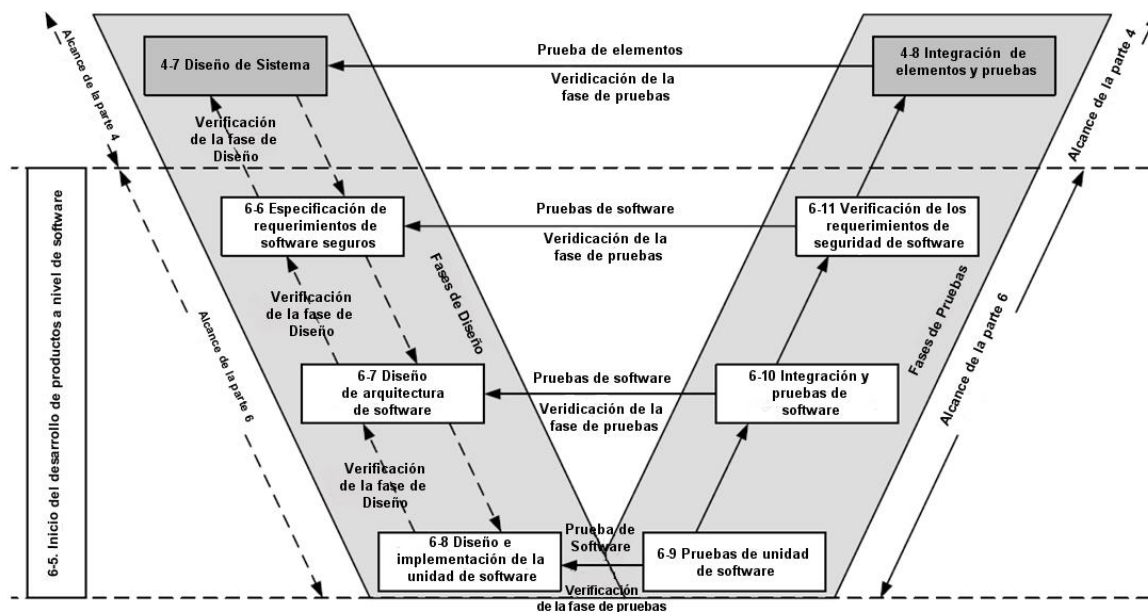


Figura 6. Modelo de desarrollo de software [16]

Las aplicaciones de alta calidad enfocadas en el ISO 26262 contribuyen en las unidades de software que son probadas, y a pesar de que la norma no especifica un estándar de código en particular, requiere que sean empleadas. Más allá de la adherencia a los estándares de código las herramientas de software pueden revisar y hacer cumplir las guías de calidad para el diseño de unidades de software y facilita su integración y pruebas acorde a lo definido en la arquitectura de software y a los requerimientos del sistema, aplicando y haciendo cumplir tales principios a un nivel de unidad de código, logrando trabajen juntas con la arquitectura definida. Idealmente una herramienta de integración deberá colaborar y facilitar automáticamente para que se puedan aplicar todas las etapas de desarrollo en el modelo en V y puedan coordinar los requerimientos con trazabilidad, análisis y pruebas en todas las etapas del producto desarrollado. Desde una amplia perspectiva las prácticas del ISO 26262 hacen un código más entendible, más confiable, menos propenso a los errores, fácil para probar y dar mantenimiento [17], ver figura 7.

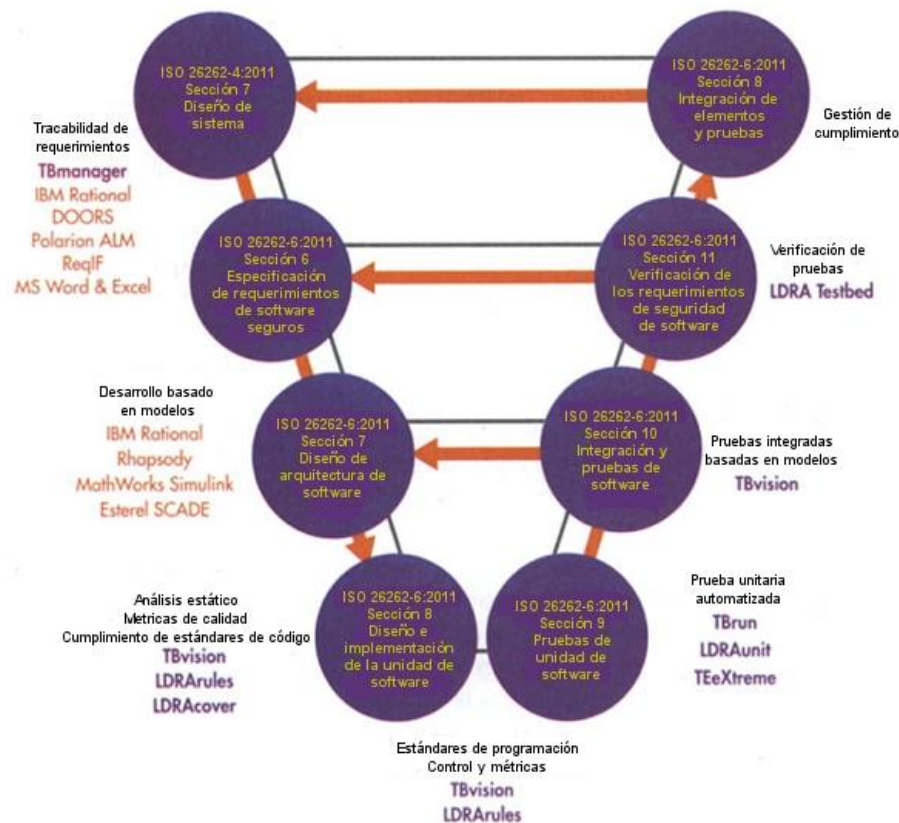


Figura 7. Capacidades de herramientas en el uso de guías de ISO 26262. [17]

2.2.1.3. ISO/IEC/IEEE 15288

Otro estándar es el ISO/IEC/IEEE 15288 (*Systems and Software Engineering – System life cycle Processes*). Que tiene como propósito proporcionar un grupo definido de procesos para facilitar la comunicación entre compradores, proveedores y otras partes interesadas en el ciclo de vida de los sistemas. Otro propósito es que este estándar pueda ser usado como una guía para establecer aspectos del negocio como: métodos, técnicas, procedimientos, herramientas y entrenamiento de personal.

El estándar ISO/IEC/IEEE 15288 establece un marco de referencia común de descripción de procesos para el ciclo de vida del desarrollo de sistemas. Define un grupo de procesos y terminología asociada desde el punto de vista de ingeniería. Estos procesos pueden ser aplicados a algún nivel en la jerarquía de la estructura de sistemas. Esto es llevado a cabo a través de la participación de todas las partes interesadas con el establecimiento de una meta para lograr la satisfacción de los clientes [18]. El ISO/IEC/IEEE 15288 agrupa las actividades que se pueden realizar durante el ciclo de vida del sistema en cuatro grupos de procesos: Proceso de Acuerdo, Proyecto Organizacional – proceso de habilitación, Proceso de Gestión Tecnológica y Proceso Tecnológico. Cada proceso incluye su propósito, resultados deseados, lista las actividades y las tareas que deben realizarse para lograr los resultados esperados. Los cuatro grupos de procesos y sus subprocesos se representan en la figura 8 [19].



Figura 8. Procesos del ciclo de vida del sistema [19]

Definición de Requerimientos de los *stakeholders*, tiene como objetivo, ayudar a definir los requerimientos del sistema de tal manera que éste cuente con los servicios que el usuario necesita o los servicios que un *stakeholder* requiera en su contexto de trabajo. Adicionalmente, en este proceso, se hace una identificación de *stakeholders*, o grupo(s) de *stakeholders*. En este proceso se analizan y transforman las necesidades, expectativas y deseos de los *stakeholders* en un conjunto de requerimientos que expresan la interacción que existirá entre el sistema y su ambiente operacional [20].

Los procesos de administración del proyecto para la norma ISO/IEC/IEEE 15288 se encuentran en el apartado *Technical Management Processes*. El proceso de gestión tecnológica por sus siglas en inglés, es usado para establecer y evolucionar los planes, ejecutar los planes, evaluar el logro real, comparar el progreso contra los planes, y la ejecución del control a través del cumplimiento. Este proceso puede ser invocado todo el tiempo en el ciclo de vida y en algún nivel en la jerarquía del proyecto, así como requerido por planes o eventos inesperados. Este proceso puede ser aplicado con el nivel de rigor y formalidad que dependa del riesgo y complejidad del proyecto.

El proceso de control de gestión tecnológica contempla las siguientes clausulas [27]:

6.3.1. Proceso de planeación del proyecto

6.3.2. Proceso de evaluación y evaluación del proyecto

6.3.3. Proceso de gestión de decisión

6.3.4. Proceso de gestión de riesgo

6.3.5. Proceso de gestión de la configuración

6.3.6. Proceso de gestión de la información

6.3.7. Proceso de medición

6.3.8. Proceso de aseguramiento de calidad

La referencia de la norma ISO/IEC/IEEE 15288 en el apartado 6.3.1 Proceso de planeación del proyecto y el apartado 6.3.2 Proceso de evaluación y evaluación del proyecto son descritos a continuación en cada una de sus secciones.

6.3.1. Proceso de planeación del proyecto.

Salidas

- Objetivos y planes
- Roles, responsabilidades u actividades son definidas
- Recursos y servicios necesarios para lograr los objetivos son formalmente solicitados y contenidos
- Planes para la ejecución del proyecto son activadas

Actividades y tareas

- Definir el proyecto
 - Identificar el objetivo y restricciones
 - Definir el alcance del proyecto establecimiento de acuerdos
 - Definir y mantener un modelo de ciclo de vida que este compuesto de niveles usando la definición de modelos de ciclo de vida de la organización.
 - Establecer una estructura de desglose de trabajo basado en la evolución de la arquitectura del sistema
 - Definir y mantener el proceso que deberá ser aplicado en el proyecto
- Plan de proyecto y administración técnica
 - Define y mantiene un calendario de proyecto basado en administración, objetivos técnicos y estimación de trabajo

- Define criterios de logro, etapas de decisión, fechas de entrega dependencias principales en entradas o salidas externas
- Define los costos y plan de presupuesto
- Define roles, responsabilidades y autoridades
- Define la infraestructura y servicios requeridos
- Plan de adquisición de materiales y habilita servicios de sistema subministrados desde la salida del proyecto
- Generar y comunicar un plan para el proyecto, manejo técnico y ejecución, incluyendo revisiones
- Activar el proyecto
 - Obtener autorización para el proyecto
 - Enviar petición y obtener compromiso para recursos necesarios para desempeños del proyecto
 - Implementar plan del proyecto

6.3.2. Proceso de evaluación y control de proyecto

Salidas

- Medición del desempeño o aseguramiento de resultados son disponibles.
- Adecuación de roles, responsabilidades
- La adecuación de los recursos es evaluada
- Revisiones del proceso técnico son desarrolladas
- Desviaciones en el desempeño del proceso desde los planes son investigadas y analizadas
- *Stakeholder* afectados son informados del estado del proyecto
- Acciones correctivas son definidas y dirigidas, cuando los logros del proyecto no estén cumpliendo el objetivo
- Re planeación del proyecto es iniciado cuando es necesario

- Acciones del progreso para el proyecto desde un calendario o evento para continuar es autorizado
- Objetivos del proyecto son logrados

Actividades y tareas

- Plan para responsabilidades del proyecto y control
 - Define las responsabilidades del proyecto y estrategias de control
- Evaluación del proyecto
 - Evaluar la alineación de los objetivos del proyecto y planes con el contexto del proyecto
 - Evalúa la gestión y planes técnicos contra los objetivos para determinar la adecuada viabilidad
 - Evaluar el proyecto y el estado técnico contra los planes apropiados para determinar los costos actuales y costos proyectados, calendarios y variaciones del desempeño
 - Evaluar la adecuación de roles, responsabilidades y autoridades
 - Evaluar la adecuación y disponibilidad de recursos
 - Evaluar el progreso usando medidas de logros y finalización del hito
 - Conducta requerida a la gestión y revisiones técnicas, auditorias e inspecciones
 - Monitoreo crítico del proceso y nuevas tecnologías
 - Analizar resultados medidos y hacer recomendaciones
 - Registro y proporcionar estado y búsqueda desde tareas estimadas
 - Monitoreo del proceso de ejecución con el proyecto
- Control del proyecto
 - Iniciar acciones necesarias direccionadas a problemas identificados
 - Iniciar la re planificación necesaria del proyecto
 - Iniciar los cambios de acción cuando se detecte un cambio contra actual, de costo, tiempo, calidad debida para el impacto adquiriente o solicitud del proveedor

- Autorizar el proyecto para proceder junto el siguiente hito o evento si es justificado

Las recomendaciones de la norma internacional ISO/IEC/IEEE 15288 para el proceso de planeación del proyecto, son muy valiosas, ya que sirven de guía para el desarrollo de los sistemas de software [28].

2.2.2. MODELO DE PROCESO DE SOFTWARE

Los modelos de proceso prescriptivo define un conjunto prescrito de elementos del proceso como acciones de ingeniería de software, aseguramiento de calidad, mecanismos de control de cambio para cada proyecto y un flujo predecible para el trabajo del proceso, es decir, la manera en la que los elementos del proceso se relacionan entre sí. Cada modelo de proceso de software puede incluir actividades estructurales generales pero pone distinto énfasis en cada una de ellas, definiendo un distinto flujo de proceso.

Los modelos de proceso especializado tienen muchas de las características de uno o más de los modelos tradicionales, sin embargo dichos modelos tienden a aplicarse cuando se elige un enfoque de ingeniería especializado o definido muy específicamente.

Entre los principales modelos de proceso de software se tienen los siguientes:

- Modelos de proceso prescriptivo
 - Modelo de la cascada (Ciclo de vida clásico)
 - Modelo en V
 - Modelos de proceso incremental
 - Modelos de proceso evolutivo

- Modelo espiral
- Modelos concurrentes
- Modelos de proceso especializado
 - Desarrollo basado en componentes
 - El modelo de métodos formales
 - Desarrollo de software orientado a aspectos

Desarrollo Basado en Componentes

El modelo de desarrollo basado en componentes incorpora muchas de las características del proceso en espiral, es de naturaleza evolutiva y demanda un enfoque iterativo para la creación de software. Este modelo se basa en construir aplicaciones a partir de fragmentos de software prefabricados. El modelo de desarrollo basado en componentes incorpora las siguientes etapas:

1. Se investigan y evalúan, para el tipo de aplicación de que se trate, productos disponibles basados en componentes.
2. Se consideran los aspectos de integración de componentes.
3. Se diseña una arquitectura de software para que reciba los componentes.
4. Se integran los componentes de la arquitectura.
5. Se efectúan pruebas exhaustivas para asegurar la funcionalidad apropiada.

Este modelo lleva a la reutilización de componentes, con esto el ingeniero de software tiene la posibilidad de reducir el ciclo de tiempo en el desarrollo como el costo del proyecto. [21].

2.2.3. ARQUITECTURA DE SOFTWARE

El diseño arquitectónico es un enlace crucial entre la ingeniería de requerimientos y el diseño, por que identifica los principales componentes estructurales del sistema y su relación. La arquitectura de software es importante porque afecta el desempeño, capacidad de distribución

y mantenimiento del sistema, los componentes individuales implementan los requerimientos funcionales, los requerimientos no funcionales dependen de la arquitectura del sistema, la forma en la que los componentes se organizan y comunican. La arquitectura de sistemas se modela con frecuencia usando diagramas de bloques simples (representan una imagen de alto nivel de la estructura del sistema) y cada recuadro representa un componente, los recuadros dentro de los recuadros indican que un componente se dividió en subcomponentes y las flechas significan que los datos y/o señales pasan de un componente a otro conforme la dirección de la flecha.

Es imposible representar toda la información relevante sobre la arquitectura de un sistema, por lo general se necesita representar múltiples vistas de la arquitectura de software, en el modelo de vista 4+1 se sugiere deben existir cuatro vistas arquitectónicas fundamentales:

- La vista lógica, indica las abstracciones clave en el sistema como objetos o clases de objeto. En este tipo de vista se tiene que relacionar los requerimientos del sistema con entidades.
- La vista de proceso, indica en tiempo de operación, los procesos en interacción que componen el sistema, esto es útil para hacer juicios acerca de las características no funcionales como el rendimiento y la disponibilidad
- La vista de desarrollo, indica la descomposición del software en elementos que se implementen mediante un solo desarrollador o equipo de desarrollo.
- La vista física, expone el hardware del sistema y cómo los componentes de software se distribuyen a través de los procesadores en el sistema.

Las vistas conceptuales casi siempre se desarrollan durante el proceso de diseño y se usan para apoyar la toma de decisiones arquitectónicas, son una forma de comunicar a diferentes participantes la esencia del sistema.

La arquitectura de software puede basarse en un patrón o estilo arquitectónico en particular, es necesario elegir la estructura más adecuada que permita satisfacer los requerimientos del sistema. Existen diferentes patrones arquitectónicos de software:

- Arquitectura por capas
- Arquitectura de repositorio
- Arquitectura cliente – servidor
- Arquitectura tubería y filtro
- Arquitectura de aplicación

Los sistemas embebidos utilizan diferentes patrones arquitectónicos:

- Observar y reaccionar
- Control ambiental
- Segmentación de proceso

El patrón de control ambiental es el más difundido, el software controla la operación del equipo con base en estímulos del entorno, cuando se detectan cambios ambientales detectados por el sensor, se envían señales de control a los actuadores del sistema [22].

2.2.4. PRUEBAS UNITARIAS

Para el uso de pruebas unitarias se utilizará la plantilla de reporte de pruebas, que está incluido en la metodología *Personal Software Process* PSP desde la versión 1, esta plantilla es usada para registrar datos de cada prueba. Los registros de las condiciones de prueba y resultados son de ayuda, cuando son incorporados los pequeños programas dentro de grandes programas identificando las modificaciones posteriores, correcciones de problemas, incluso re uso de programas, Se encontrará esta ayuda regresando a las pruebas completadas previamente, conociendo que la prueba fue realizada, que datos fueron usados y los resultados que fueron obtenidos. Haciendo esto típicamente puede ser de ayuda cuando un programa es modificado y

se incorpora nueva funcionalidad, como parte de la prueba, queremos asegurarnos que los cambios no han causado problemas con la funcionalidad previamente trabajada. [23].

2.2.5. PRUEBAS DE INTEGRACIÓN

El proceso de la integración del sistema se basa en la construcción a partir de sus componentes, y probar el sistema resultante para encontrar problemas que puedan surgir debido a la integración de los componentes. Estos componentes pueden ser adquiridos, reutilizados y adaptados a un sistema particular, o nuevos desarrollos. En muchos sistemas grandes, es probable que se utilicen los tres tipos de componentes., las pruebas de integración comprueban que estos componentes realmente funcionan juntos, son llamados correctamente y transfieren los datos correctos en el tiempo preciso a través de sus interfaces. [25].

2.2.6. PRUEBAS DE SISTEMA

Las pruebas de sistema no es un proceso de prueba de funciones del sistema completo, tienen como propósito particular el comparar el sistema con los objetivos originales, el diseño de estas pruebas deberá formular casos de prueba que analicen la documentación del usuario para analizar los objetivos y estos serán el propósito central de las pruebas de sistema. La razón para esto, es que el estado de los objetivos indica lo que el programa deberá hacer y que tan bien deberá hacerlo. Existen 15 categorías de tipos de pruebas que pueden ser aplicadas como: Pruebas de volumen, pruebas de estrés, pruebas de usabilidad, pruebas de seguridad, pruebas de desempeño, pruebas de almacenamiento, pruebas de configuración, entre otras. [26].

2.2.7. PRUEBAS DE ACEPTACIÓN

Las pruebas de aceptación certifican que el sistema de software satisface con los requerimientos originales., Estas pruebas deberán de no ser desempeñadas hasta que se tengan las pruebas del sistema satisfactoriamente completadas. El usuario final es el responsable de asegurar que toda la funcionalidad relevante ha sido probada. Los planes de pruebas de aceptación definen los procedimientos a ejecutar y deberán ser seguidos de cerca como sea posible. Algunos proyectos no requieren una prueba de aceptación formal cuando el cliente o el usuario están satisfechos con las pruebas del sistema, cuando en el momento oportuno los requerimientos han sido demandados o cuando los usuarios han sido involucrados continuamente a través del ciclo de desarrollo y han sido aplicadas las pruebas de aceptación implícitamente durante el desarrollo del sistema. [27].

CAPÍTULO 3. DESARROLLO DEL PROYECTO

3.1. ASPECTOS TÉCNICOS

El Automóvil para su funcionamiento está formado por una gran cantidad de sistemas como: a) Sistema de frenos, b) Sistema de comunicaciones, c) Sistemas de seguridad, d) Sistema de comodidad, por mencionar algunos y estos hacen uso de sensores, los cuales transforman magnitudes físicas o químicas (mayormente no eléctricas) en magnitudes eléctricas (a veces a través de etapas no eléctricas). Y actuadores los cuales son elementos de acoplamiento entre el procesamiento eléctrico de la señal (procesamiento de la información) y el proceso (la mecánica). Transforman las señales de poca energía portantes de la información del lugar de medición en la forma de energía requerida para intervenir en el proceso mediante señales con carga. Los transductores (transformadores) de señal combinan los elementos de amplificación, aprovechan los principios de transformación de la física entre varias formas de energía (eléctrica, mecánica, de fluidos, térmica). [6].

Los sensores y actuadores están conectados a la ECU (*Electronic Control Unit*) El ECU procesa la información recibida desde los sensores de acuerdo a procedimientos matemáticos definidos (algoritmos de control). Controlan a los actuadores principalmente por señales de salidas eléctricas. Adicionalmente las ECU actúan como una interface con otros sistemas. [24].

3.2. ANÁLISIS Y REQUERIMIENTOS

Dentro del estudio de la ingeniería de software, existe una serie de documentos que ayudan a tener un mejor entendimiento en la creación de sistemas, desde lo general hasta lo particular.

Estos documentos sirven como guía, identificando aspectos del negocio, diferentes escenarios, casos de uso e incluso funcionalidad importante para el desarrollo de los proyectos.

3.2.1. DOCUMENTO DE VISIÓN Y ALCANCE

3.2.1.1. REQUERIMIENTOS DEL NEGOCIO

3.2.1.1.1. BACKGROUND

En la actualidad existen sistemas de seguridad Activa y Pasiva. Los cuales son enfocados a prevenir accidentes antes de una colisión en el caso de los primeros y actuar durante de la colisión en el caso de los segundos. Sin embargo, hay pocos estudios de seguridad que estén enfocados en situaciones presentadas después de una colisión.

3.2.1.1.2. OPORTUNIDAD DE NEGOCIO

Generar propuestas de comportamiento en los automóviles para eventos post-colisión y promover su inclusión en los autos.

3.2.1.1.3. OBJETIVOS DE NEGOCIO Y CRITERIOS DE ÉXITO

Los criterios de éxito del sistema propuesto son:

- Minimizar el tiempo en que los ocupantes de un vehículo siniestrado reciben ayuda del exterior.
- Disminuir las estadísticas de pérdida de vidas humanas por no haber recibido auxilio de manera expedita, después de una colisión.

3.2.1.1.4. NECESIDADES DEL CLIENTE O DE MERCADO

Se requiere de más propuestas de comportamiento de sistemas de seguridad post-colisión, que minimicen el tiempo en que los pasajeros reciban asistencia después de un choque severo. Adicionalmente, estos sistemas deben ser desarrollados bajo el estándar automotriz ISO 26262 debido a que son sistemas de misión crítica.

3.2.1.1.5. RIESGOS DE NEGOCIO

- Que el sistema no sea de interés para las diferentes armadoras.
- Que el costo de implementación incremente significativamente el costo de los vehículos.

3.2.1.2. VISIÓN DEL PRODUCTO

Que los automóviles modernos estén dotados de sistemas que minimicen los tiempos en que los pasajeros de un vehículo siniestrado reciban ayuda del exterior cuando, por sus propios medios no sean capaces de salir del automóvil colisionado.

3.2.1.2.1. CARACTERÍSTICAS IMPORTANTES

- Detección de colisión.
- Lectura de Sensores (Impacto, acelerómetro, giroscopio, presión, velocidad).
- Manipulación de Actuadores (Luz interior, luces intermitentes, seguros eléctricos, elevadores de ventanas, desanclar cinturones de seguridad).
- Minimizar tiempo de asistencia a ocupantes del vehículo.

3.2.1.2.2. SUPOSICIONES Y DEPENDENCIAS

- Se asume que el automóvil tiene instalados el hardware necesario para poder ejecutar toda la funcionalidad.
- Se asume que el automóvil tiene una red CAN de comunicación, para transmisión de datos.
- Se asume que el sistema cuenta con alimentación eléctrica.

3.2.1.3. ALCANCE Y LIMITACIONES

- El sistema post-colisión se enfocará en acciones que permitan disminuir el tiempo en que los pasajeros reciben asistencia, después de una colisión.
- El sistema post-colisión funcionará de manera autónoma sin la intervención del usuario.

3.2.1.3.1. Limitaciones y Excepciones

El sistema post-colisión no funcionará en caso que tanto la fuente primaria de alimentación como la secundaria hayan sido deshabilitadas debido a la colisión. Los servicios que proporcionará el sistema no podrán llevarse a cabo por daños físicos de los componentes del auto, que impidan la acción de los actuadores.

3.2.1.4. CONTEXTO DEL NEGOCIO

El sistema post-colisión deberá estar preparado para su implementación, ante los rápidos cambios de necesidades del mercado automotriz.

3.2.1.4.1. AMBIENTE DE OPERACIÓN

El sistema post-colisión estará instalado en los automóviles y el usuario no intervendrá en su funcionamiento. El sistema responderá a las condiciones detectadas por los sensores y ejecutará sus funciones de acuerdo a un análisis de las entradas de los mismos.

3.2.2. PROCESO DEFINICIÓN DE REQUERIMIENTOS DE STAKEHOLDER

Dentro de las actividades y tareas del proceso de definición de requerimientos y necesidades del *stakeholder*. Se deberá implementar el seguimiento de actividades y tareas de acuerdo a las políticas y procedimientos de la organización mediante las siguientes actividades. [16]

- Prepararse para la definición de requerimientos y necesidades del *stakeholder*.
- Definir las necesidades del *stakeholder*.
- Desarrollar el concepto operativo y otros conceptos del ciclo de vida del desarrollo del sistema.
- Transformar las necesidades de *stakeholder* en requerimientos de *stakeholder*.
- Analizar los requerimientos de *stakeholder*.

Como se describió en la sección del Marco Teórico, el proceso de Definición de Requerimientos del *stakeholder*, sugiere registrar las necesidades de las partes interesadas en varios documentos, entre ellos, el Documento de Concepto (ConOps), que se presenta en la siguiente sección. En esta sección se cuenta con algunos aspectos importantes, los documentos completos de Visión y Alcance se encuentra en el Anexo A y el documento de Concepto de Operaciones en el Anexo B.

3.2.3. DOCUMENTO DE CONCEPTO

3.2.3.1. INTRODUCCIÓN

Propósito del documento de concepto.

Actualmente, uno de los principales medios de transporte es el automóvil, con 1.2 billones de autos en todo el mundo. La cantidad impresionante de vehículos implica un alto número de accidentes, por ejemplo, tan sólo en Canadá, cada año, entre 2000 y 2010, hubo un promedio de 2,632 personas fallecidas, y 1.5 millones lastimados [3]. Ante este tipo de escenarios, es necesario que las personas que se encuentran dentro de un vehículo siniestrado reciban ayuda del exterior de una forma rápida y sin complicaciones. Por tal motivo el sistema post-colisión propuesto, busca minimizar el tiempo en el que los ocupantes reciben ayuda, para ello facilitará el acceso al interior del auto, activando algunos servicios como: encender las luces del interior, encender las luces intermitentes, liberar seguros eléctricos, bajar automática y temporizadamente ventanas y desanclar cinturones de seguridad. Para ejecutar esta funcionalidad, el sistema verificará: a) Que el vehículo ya no se encuentre en movimiento, b) Cual es la posición final del auto. Este comportamiento contribuye a una asistencia expedita a las personas que se encuentran en el interior del auto siniestrado.

3.2.3.2. CONCEPTO DE PRODUCCIÓN

El sistema será fabricado mediante tarjetas electrónicas de control las cuales tendrán software embebido. Adicionalmente, estarán habilitadas para llevar a cabo una comunicación mediante el protocolo de comunicación automotriz CAN (*Control Area Network*).

3.2.3.3. CONCEPTO DE OPERACIONES (CONOPS)

Este sistema hace lectura de diferentes sensores que determinan el estado del vehículo después de una colisión. El software de control manipulará a los actuadores de acuerdo a los requerimientos establecidos para cada escenario.

Los escenarios visualizados determinarán el comportamiento del sistema y para ello se deberán realizar las siguientes tareas:

- Evaluar colisión.

El sistema post-colisión detectará una colisión y el grado de severidad de la misma.

- Detectar estado de reposo del vehículo.

Después del impacto, el sistema identificará el momento en el que el auto alcanza un estado de reposo.

- Rango de Tiempo seguro.

Cuando el vehículo ha dejado de moverse se determinará un rango de tiempo seguro en el cual el vehículo ha permanecido en reposo. Después de este rango de tiempo seguro, el sistema iniciará la manipulación de los diferentes actuadores.

- Posición del vehículo.

El comportamiento del sistema post-colisión depende de la posición final del vehículo. Por lo que se debe determinar.

- Activar la fuente secundaria de energía.

La fuente secundaria de energía se activa sólo en caso que la fuente primaria (batería) esté dañada o desconectada.

- Encender luces interiores.

En caso que la colisión sea en la noche, esta funcionalidad ayuda a los ocupantes a poder ver el interior del auto.

- Desactivar seguros de puertas.

Con esta funcionalidad se liberan los seguros de las puertas para que puedan ser abiertas desde el exterior.

- Bajar ventanas.

Existen escenarios que se deben analizar, para determinar si las ventanas se pueden bajar o no.

- Identificar presencia de ocupantes en el vehículo.

Es necesario conocer en que asientos hay ocupantes, para saber que ventanas deben ser bajadas.

- Identificar posición del Automóvil.

Es necesario saber en qué posición física quedó el auto. Las posiciones posibles son: a) Todas las llantas hacia arriba, b) Todas las llantas hacia abajo, c) Llantas traseras hacia arriba, d) Llantas delanteras hacia arriba, e) Llantas lado del conductor hacia arriba, f) Llantas lado del copiloto hacia arriba.

- Identificar presión de ventana.

Se identificará si algún pasajero se encuentra recargado en la ventana. En caso afirmativo, la ventana no se bajará.

- Desanclar cinturones de seguridad.

Los cinturones no se deberán liberar de forma automática. Para liberarlos, el sistema deberá evaluar los incisos a) al f) de la funcionalidad “Identificar posición del Automóvil”. Dependiendo de esta evaluación se determina si se liberan todos los cinturones o sólo los que apliquen.

En esta sección se cuenta con algunos aspectos importantes, el documento completo se encuentra en el Anexo B.

3.2.4. PROCESO ANÁLISIS DE REQUERIMIENTOS

El propósito del proceso de Análisis de Requerimientos del estándar ISO/IEC/IEEE 15288, es transformar las necesidades o servicios deseados de los *stakeholders* en requerimientos técnicos del sistema.

Las actividades de este proceso, generan una representación del sistema, que deberá satisfacer los requerimientos funcionales de los *stakeholders* y cumplir las restricciones o requerimientos no funcionales. En este documento sólo se presentan los requerimientos funcionales.

3.2.5. MODELADO DE REQUERIMIENTOS

La especificación de requerimientos se realizó mediante una combinación de lenguaje natural (notación informal) y lenguajes de modelado (notación semi-formal). En esta propuesta, se utiliza el lenguaje de modelado SysML (*Systems Modeling Language*). Para la representación de los requerimientos del sistema propuesto, se utilizó una notación semi-formal para cumplir con las recomendaciones del ISO 26262.

En la figura 9, se puede observar el requerimiento funcional: “El sistema post-colisión debe evaluar un evento de colisión”. Para lograr esta funcionalidad, el sistema contará con sensores

de impacto que detectarán una colisión frontal, trasera o lateral y el grado de severidad de la misma.

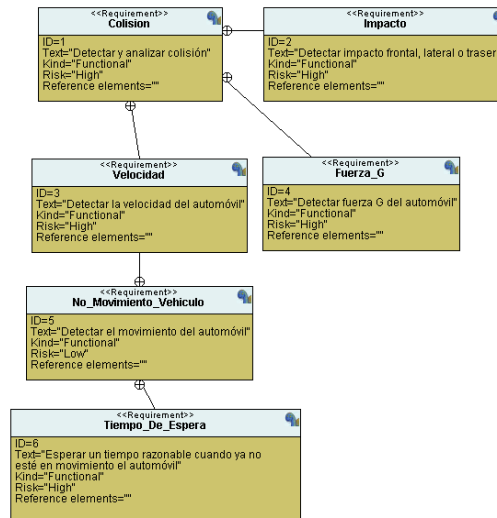


Figura 9. Diagrama de Requerimientos. Detección de impacto.

En la figura 10 se puede observar el requerimiento funcional “El sistema post-colisión deberá activar la fuente de energía secundaria”. Para lograr esta funcionalidad el sistema identificara si no existe energía primaria si esto es afirmativo, entonces, el sistema activará el funcionamiento de la fuente de energía secundaria.

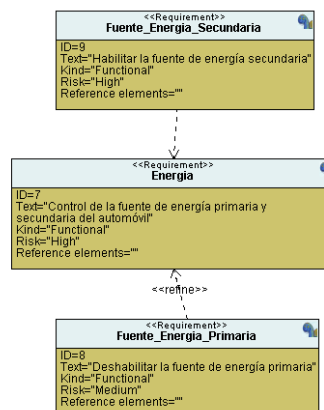


Figura 10. Diagrama de Requerimientos. Habilitar fuente de energía secundaria.

En la figura 11 se puede observar el requerimiento funcional “El sistema post-colisión deberá manipular actuadores dependiendo del estado del vehículo”. Para lograr esta funcionalidad el sistema analizará la información de los sensores que determinan la posición final física del vehículo y la presión en las ventanas. Con esta información, el sistema ejecutará las actividades de encendido de luces, desactivar los seguros eléctricos, bajar los cristales de las ventanas y desanclar los cinturones de seguridad.

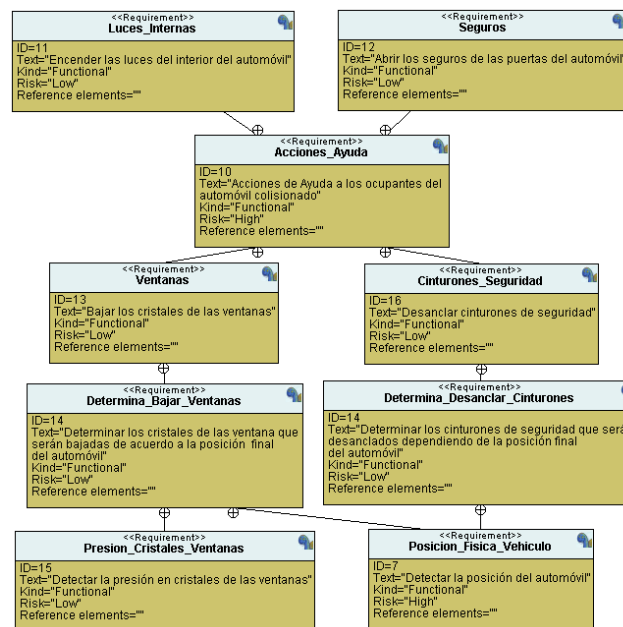


Figura 11. Diagrama de Requerimientos. Activación de actuadores.

3.2.6. DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS DE SOFTWARE SRS

3.2.6.1. ESPECIFICACIÓN DE REQUERIMIENTOS

El sistema deberá:

1. Detectar un impacto frontal, lateral o trasero.
2. Detectar la velocidad del automóvil.
3. Detectar la fuerza G del automóvil.

4. Detectar el movimiento del automóvil.
5. Esperar un tiempo razonable cuando ya no esté en movimiento el automóvil.
6. Detectar la posición del automóvil.
7. Deshabilitar la fuente primaria de energía.
8. Habilitar la fuente secundaria de energía.
9. Encender las luces del interior del automóvil.
10. Abrir los seguros de puertas del auto.
11. Detectar la presión en cristales de las ventanas.
12. Determinar los cristales de las ventanas que serán bajadas de acuerdo a la posición final del automóvil.
13. Bajar los cristales de las ventanas.
14. Determinar los cinturones de seguridad que serán desanclados dependiendo de la posición final del automóvil.
15. Desanclar cinturones de seguridad.

3.2.6.2. CASOS DE USO

Como apoyo adicional a los procesos de Definición y Análisis de Requerimientos, se desarrollaron casos de uso que describen los requerimientos mostrados en la sección anterior. En esa sección se agregan detalles de los requerimientos mediante el flujo básico y el flujo alternativo de cada caso de uso mostrado. Por espacio, sólo se presentan dos casos de uso completos. En el caso de uso 1, Evalúa colisión, cuando algunos de los sensores de impacto detectan alguna señal, inmediatamente se lee el sensor de velocidad y el sensor acelerómetro para identificar la velocidad a la que viaja el vehículo y para conocer la fuerza de desaceleración que ha tenido el vehículo con el impacto detectado, respectivamente, para poder realizar un análisis y determinar si la colisión ha sido severa, (Ver figura 12).

Caso de uso	CU1 Evalúa colisión
-------------	---------------------

Actores	Sensor de impacto frontal, Sensor de impacto lateral derecho, Sensor de impacto izquierdo, Sensor de impacto trasero, Sensor de velocidad, Sensor acelerómetro.
Descripción	Identificar un impacto por los diferentes sensores de impacto, con una velocidad mayor a los 100 kilómetros por hora y una fuerza de desaceleración mayor o igual a 3G.
Tipo	Primario
Incluye	Funcionalidad de los sensores.
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	N/A
Flujo básico	1.- El sistema lee entrada del sensor de impacto frontal. 2.- El sistema lee entrada del sensor de impacto lateral derecho. 3.- El sistema lee entrada del sensor de impacto lateral izquierdo. 4.- El sistema lee entrada del sensor de impacto trasero. 5.- El sistema lee el sensor de velocidad. 6.- El sistema lee el sensor de fuerza G. 7.- El sistema procesa entradas de los sensores y evalúa las condiciones de un impacto severo para iniciar el sistema que consisten en detección de un impacto en cualquiera de los sensores de impacto, velocidad > a 100 km por hora y fuerza G $\geq$ 3. Si no se cumplen las condiciones se quedarán los sensores alertas, se pasa al paso 1.
Flujo alternativo	1.1.- El sistema no puede leer el sensor de impacto frontal. 1.2.- El sistema muestra testigo en tablero. 2.1.- El sistema no puede leer el sensor de impacto lateral derecho.

	2.2.- El sistema muestra testigo en el tablero. 3.1.- El sistema no puede leer el sensor de impacto lateral izquierdo. 3.2.- El sistema muestra testigo en el tablero. 4.1.- El sistema no puede leer el sensor de impacto trasero. 4.2.- El sistema muestra testigo en el tablero. 5.1.- El sistema no puede leer el sensor de velocidad. 5.2.- El sistema muestra testigo en el tablero. 6.1.- El sistema no puede leer el sensor de fuerza G. 6.2.- El sistema muestra testigo en el tablero.
--	--

Figura 12. Caso de Uso 1. Evalúa colisión.

En el caso de uso 9 el sistema realizara un análisis identificando la información obtenida en la identificación de la posición del vehículo y detectando si se tiene presión en las ventanas del automóvil, para que se pueda determinar que ventanas es posible bajar los cristales de las ventanas. Ver figura 13.

Caso de uso	CU9 Bajar ventanas
Actores	Sensor de presión de cristales de ventanas, Actuador de elevadores.
Descripción	Analizar la posición en la que quedo el vehículo combinado con la presión de los diferentes cristales de las ventanas para así decidir que ventanas son libres para poder bajarlas.
Tipo	Primario.
Incluye	Funcionalidad de Actuadores.
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU3 y CU5

Flujo básico	1.- El sistema lee sensor de presión en cristal de la ventana del piloto. 2.- El sistema lee sensor de presión en cristal de la ventana del copiloto. 3.- El sistema lee sensor de presión en cristal de la ventana trasera del piloto. 4.- El sistema lee sensor de presión en cristal de la ventana trasera del copiloto. 5.- El sistema lee la información almacenada de la posición del automóvil. 6.- El sistema procesa entradas de los sensores de presión, así como la información de la posición del automóvil. 7.- El sistema baja ventanas.
Flujo alternativo	1.1.- El sistema no detecta presión en el cristal de la ventana del piloto. 1.2.- El sistema muestra testigo en el tablero. 2.1.- El sistema no detecta presión en el cristal de la ventana del copiloto. 2.2.- El sistema muestra testigo en el tablero. 3.1.- El sistema no detecta presión en el cristal de la ventana trasera del piloto. 3.2.- El sistema muestra testigo en el tablero. 4.1.- El sistema no detecta presión en el cristal de la ventana trasera del copiloto. 4.2.- El sistema muestra testigo en el tablero.

Figura 13. Caso de Uso 9. Bajar ventanas.

En el Anexo C se encuentra el documento completo de Especificación de Requerimientos SRS con los 10 casos de uso propuestos, en esta sección solo se mencionaron los casos de uso uno “Evalúa Colisión” y el caso de uso nueve “Bajar Ventanas”.

3.3. DISEÑO

3.3.1. ARQUITECTURA DEL SISTEMA

3.3.1.1. Vista lógica

En esta vista se tienen las diferentes abstracciones clave del sistema representadas mediante el diagrama de clases, éste hace referencia a los diferentes tipos de *Electronic Control Unit* (ECU), por sus siglas en inglés, los cuales a su vez contienen los diferentes sensores y actuadores que intervienen en el sistema, ver figura 14.

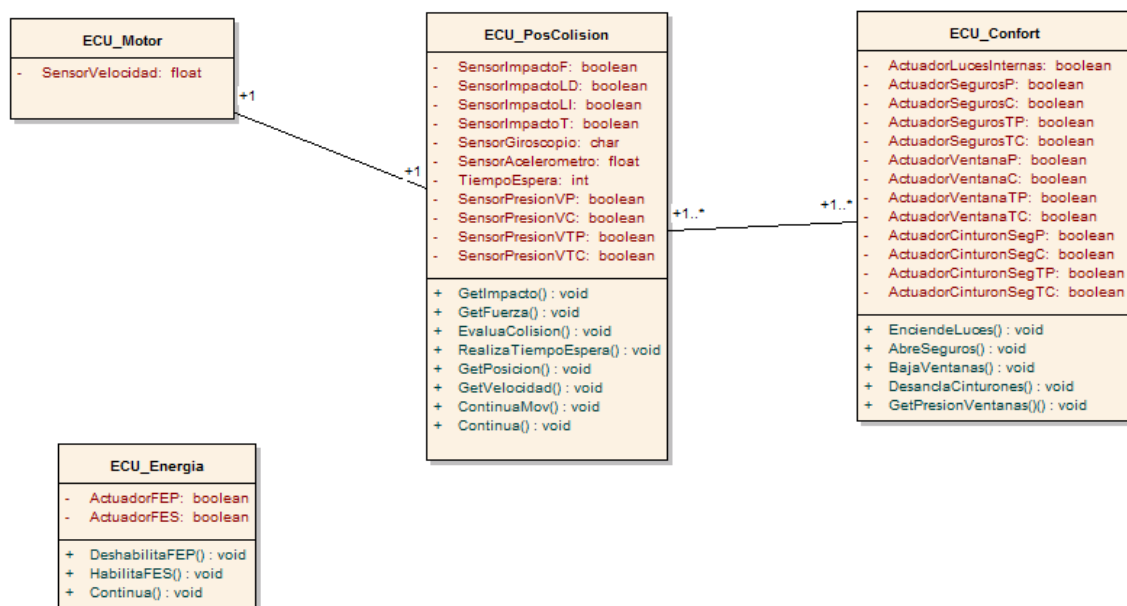


Figura 14. Diagrama de clases.

3.3.1.2. Vista de Proceso

Esta vista se representa mediante el diagrama de secuencia, caso de uso 1, evalúa colisión. En el cual se aprecia la interacción de los componentes del sistema en tiempo de operación, ver figura 15.

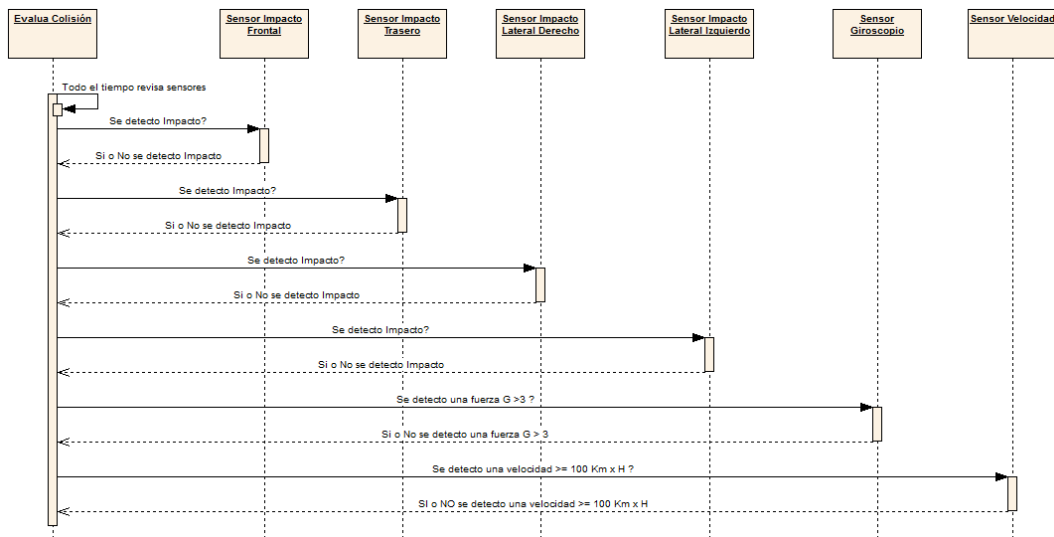


Figura 15. Diagrama de secuencia.

En el Anexo E se encuentran todos los diagramas de secuencia.

3.3.1.3. Vista de Desarrollo

El siguiente diagrama de despliegue representa la descomposición de elementos desde una vista general que los desarrolladores deberán conocer para la creación del sistema, ver figura 16.

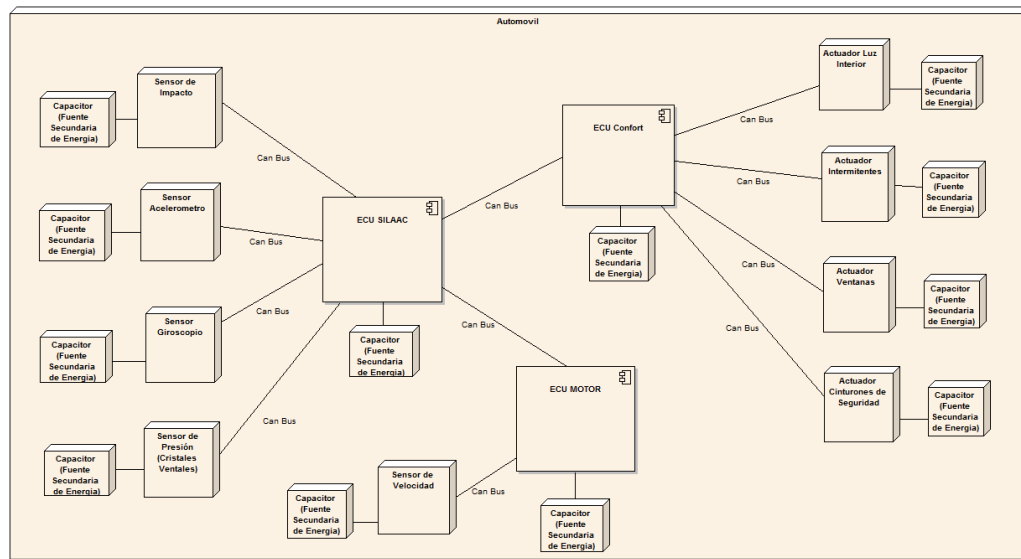


Figura 16. Diagrama de despliegue I.

3.3.1.4. Vista Física

En el siguiente diagrama de despliegue se representan los elementos y hardware del sistema así como los componentes de software, ver figura 17.

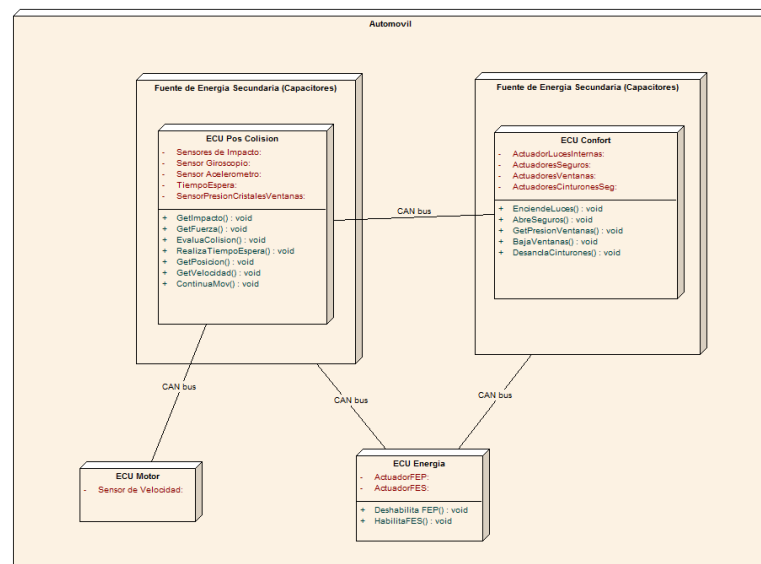


Figura 17. Diagrama de despliegue II.

3.3.2. DISEÑO DETALLADO

Para representar el diseño detallado se generó una plantilla que contiene información de control y seguimiento de los diferentes pseudocódigos que serán generados para el sistema de software de control automotriz., A continuación se presenta la plantilla de pseudocódigo. Ver figura 18.

Nombre		Fecha		Número de Pseudocódigo	
Número de Caso de Uso		Caso de Uso			
Pseudocódigo					

Figura 18. Plantilla pseudocódigo.

Se realizaron 10 pseudocódigos de los 10 casos de usos, los cuales están en el anexo F. A continuación se presenta el pseudocódigo del primer caso de uso, evalúa colisión. Ver figura 19.

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	1
Número de Caso de Uso	1	Caso de Uso	Evalúa Colisión		
Inicio Procedimiento Evalúa_Colisión Comentario: Declara variables. Colisión = verdadero tipo booleano Velocidad = falso tipo booleano Fuerza = falso tipo booleano SensorVelocidad = 0 tipo entero SensorFrontal = 0 tipo entero SensorLateralDerecho = 0 tipo entero SensorLateralIzquierdo = 0 tipo entero					

SensorFuerzaG = 0 tipo entero

Mientras (Colision == falso) y (Velocidad == verdadero) y (Fuerza == verdadero) hacer

Comentario: Verifica sensor de velocidad, ¿a qué velocidad va el vehículo?

SensorVelocidad = Lee sensor de velocidad

Si SensorVelocidad >= 100 entonces

Velocidad = Verdadero

Si no

Velocidad = Falso

Fin Si

Comentario: Verifica sensores de impacto

Lee sensor de impacto frontal

SensorFrontal = Lee sensor de impacto frontal

Comentario: ¿detecto colisión?

Si SensorFrontal == si entonces

Colision = verdadero

Si no

SensorTrasero = Lee sensor de impacto trasero

Comentario: ¿detecto colisión?

Si SensorTrasero == si entonces

Colision = verdadero

Si no

SensorLateralDerecho = Lee sensor de impacto lateral derecho

Comentario: ¿detecto colisión?

Si SensorLateralDerecho == si entonces

Colision = verdadero

Si no

Sensor LateralIzquierdo = Lee sensor de impacto lateral izquierdo

Comentario: ¿detecto colisión?

Si SensorLateralIzquierdo == si entonces

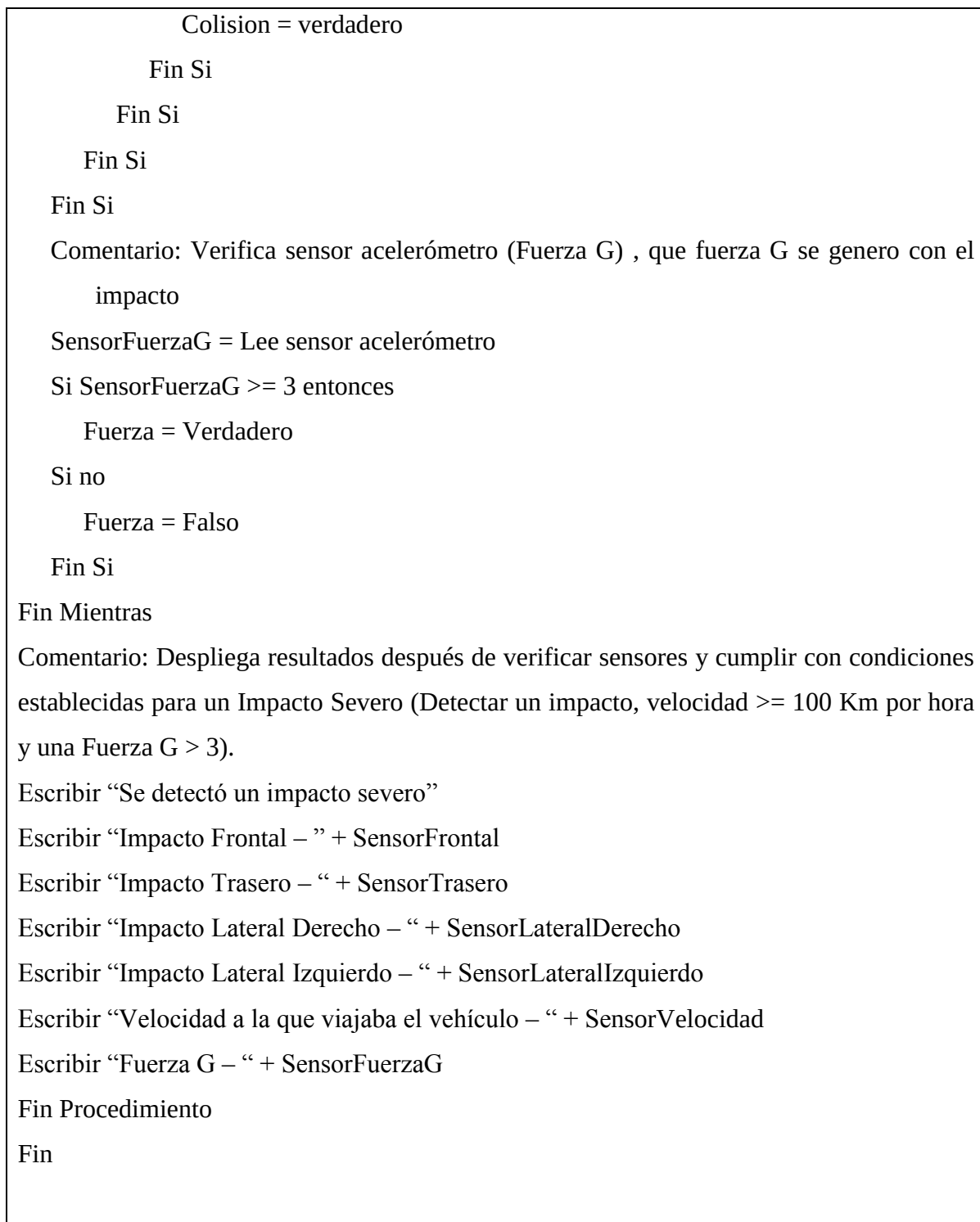


Figura 19. Pseudocódigo. Evalúa colisión.

3.4. DESARROLLO

3.4.1. ELEMENTOS UTILIZADOS

Para la creación del prototipo se utilizó la tarjeta de control Arduino la cual permite el manejo de sensores, se utilizó un giroscopio para identificar y determinar la posición en la cual quedo el vehículo después de la colisión y como actuador se utilizó un servomotor para simular el control de la ventana, así como un sensor de fuerza para poder simular la presión ejercida en la ventana y así determinar si es posible bajar los cristales de la ventana.

Los elementos que utilizaron son los siguientes:

- Arduino UNO
- Computadora de escritorio
- Sensor acelerómetro y giroscopio MPU6050
- Sensor de fuerza FSR400
- Servo motor micro Servo 9g SG90 Tower Pro
- Protoboard
- Mini protoboard
- Jumpers
- Cable USB
- Resistencia de 10 k
- Estructura metálica
- Esfera para hámster
- Pieza de acrílico
- Broche Baco
- Tornillo con tuerca
- Alfiler
- Liga

En la siguiente imagen se presenta los componentes de prueba, ver figura 20.

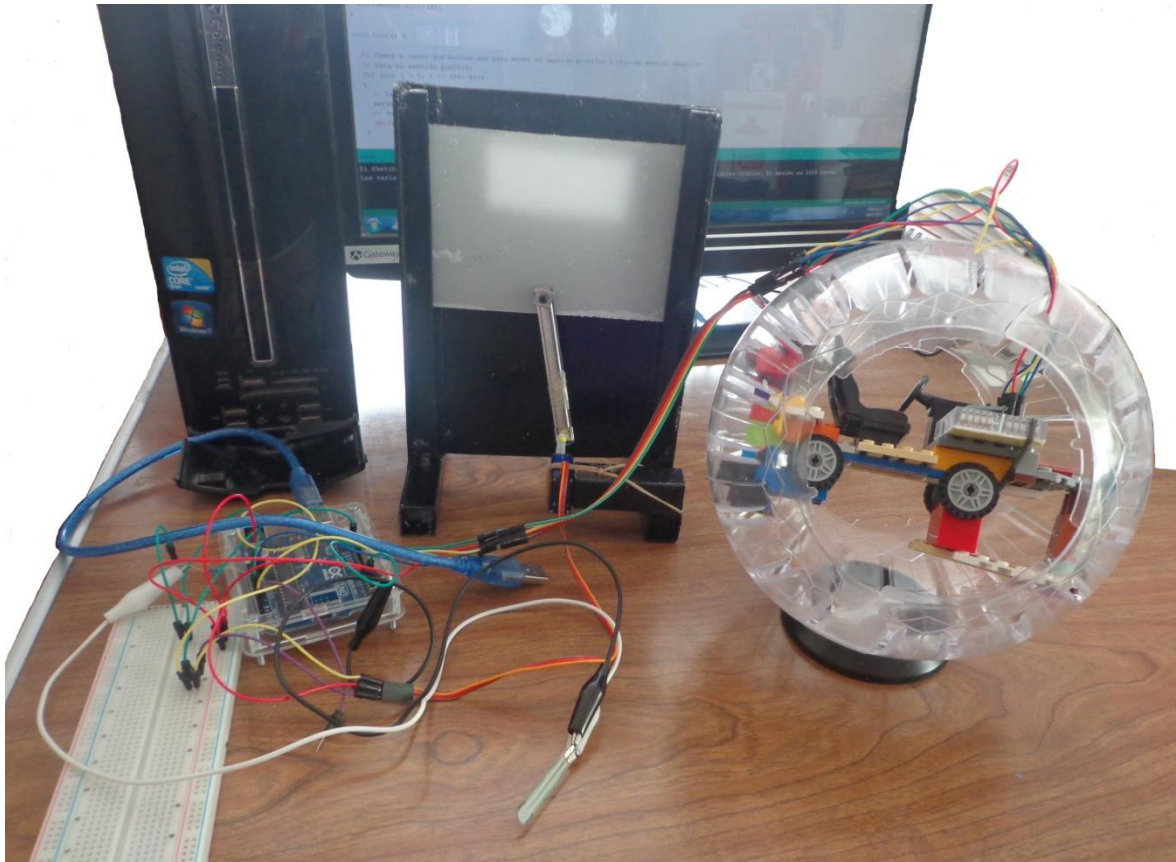


Figura 20. Escenario de pruebas.

Fig. 21. Escenario de pruebas

Para la programación se utilizó el IDE de Arduino el cual utiliza el lenguaje de programación C++.

3.4.2. CREACIÓN DEL SISTEMA

Se eligió el caso uso 4 Posición del Vehículo y el caso de uso 9 Bajar Ventanas por que este depende básicamente de la funcionalidad del caso de uso 4.

Programación de la Posición del Vehículo se investigó la referencia y programación del sensor Acelerómetro y Giroscopio MPU6050 del cual se descargó un *sketch* el cual tiene la funcionalidad básica y hace referencia a una librería que también se descargó y se configuró en el IDE, una vez conectado y ejecutando la funcionalidad base se identificó la funcionalidad y se modificó parte de la funcionalidad para evaluar las 6 diferentes posiciones que tienen especificadas en el caso de uso: 1.- Todas las llantas hacia arriba, 2.- Todas las llantas hacia abajo, 3.- Llantas traseras hacia arriba, 4.- Llantas delanteras hacia arriba, 5.- Llantas lado del piloto hacia arriba, 6.- Llantas lado del copiloto hacia arriba. Una vez identificada la posición que se lee del sensor giroscopio se grabó la información en la memoria EEPROM, la cual tiene la característica de que esta no se borra aunque se desconecte de la energía el Arduino UNO, lo cual no sucede con las variables declaradas en el *sketch*. Otra característica que se utilizó en esta programación es el uso de la función *loop* la cual una vez instalado el *sketch* en el Arduino UNO e instalado de manera correcta el sensor se puede visualizar la ejecución en el monitor de serie del IDE con una configuración a 11500 baudios.

Programación para bajar las ventanas del vehículo, esta funcionalidad se requiere tener previamente el conocimiento de la posición en la cual quedo el vehículo después de la colisión, también se requiere conocer si existe alguna presión sobre alguna ventana, para esto se utilizó un sensor de fuerza modelo FSR400, se requirió de simular el movimiento de la ventana, para lograr esto se necesitó usar un servomotor modelo Micro Servo 9g SG90 *Tower Pro*, que pudiese subir o bajar una pequeña ventana diseñada con una estructura metálica y acrílico. El funcionamiento a diferencia del *sketch* anterior es que en esta ocasión no será usada la función *loop* ya que se requiere que la lógica del programa sea ejecutada una sola vez y no un ciclo infinito que este leyendo en todo momento el sensor como en el caso del giroscopio. Por lo tanto la primera actividad del *sketch* es la lectura de la variable almacenada en la memoria EEPROM, con esto se conoce en qué posición quedo el vehículo y dependiendo de esto se procede a simular cada ventana del vehículo (piloto, trasera piloto, copiloto y trasera copiloto) posteriormente, dependiendo de la posición y simulando de igual forma la presión de cada ventana se decidirá si se manda señal al servomotor y se baja la ventana o se decidirá no bajar la ventana. Al igual que el *sketch* anterior es necesario tener la conexión correcta del

sensor de fuerza y servomotor a la tarjeta Arduino UNO. Para ser probada la funcionalidad se necesita abrir el Monitor serial con una configuración de 9600 baudios.

3.4.3. CÓDIGO FUENTE

Se generó una plantilla para la documentación del código generado, ver figura 21. El código completo de posición del vehículo y bajar ventanas se encuentra en el anexo G.

Nombre		Fecha		Programa	
Número de Caso de Uso		Caso de Uso			
Código					

Figura 21. Plantilla código.

3.4.3.1. CÓDIGO POSICIÓN DEL VEHÍCULO

En la generación de código de la posición del vehículo se utilizó un sensor acelerómetro y giroscopio MPU6050, del que se reusó código con el cual se tiene la configuración y control del giroscopio, y la generación de código se enfocó principalmente a crear la lógica necesaria para identificar y delimitar los diferentes valores necesarios en la determinación de las seis diferentes posiciones en las cuales se planeó, pudiese quedar el vehículo después de la colisión, determinadas desde el caso de uso y una vez detectado dicho valor se generó la codificación para almacenarse en la memoria EPROM de la tarjeta de control Arduino UNO, dicha información será fundamental para la funcionalidad del control de las ventanas y control de cinturones de seguridad, en otros módulos del sistema de software de control automotriz. A continuación se presenta un fragmento del código de la posición del vehículo, ver figura 22.

Nombre	GRG	Fecha	29/08/2017	Programa	PosicionVehiculo.ino
Número de Caso de Uso	4	Caso de Uso		Posición del Vehículo	

```

#include <EEPROM.h>
int ve;
int addr = 0;
void setup() {
    .....
}
void loop() {
    .....
# Código reusado
#ifndef OUTPUT_READABLE_YAWPITCHROLL
    // display Euler angles in degrees
    mpu.dmpGetQuaternion(&q, fifoBuffer);
    mpu.dmpGetGravity(&gravity, &q);
    mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);
    y = ypr[0] * 180 / M_PI;
    p = ypr[1] * 180 / M_PI;
    r = ypr[2] * 180 / M_PI;

    if ((p >= 6 && p <= 20) && (r >= -7 && r <= 7)) {
        ve = 1;
        EEPROM.write(addr, ve);
        Serial.print("(1) Todas las Llantas Arriba (Y: ");
        // Serial.println(F(" ..... Valores: ..... y : "));
        Serial.print(y);
        Serial.print(" P: ");
        Serial.print(p);
        Serial.print(" R: ");
        Serial.print(r);
        Serial.println(")");
    }
}

```

Figura 22. Fragmento de código. Posición del vehículo.

3.4.3.2. CÓDIGO BAJAR VENTANAS

Para la generación de código de esta funcionalidad, se requiere conocer la posición en la que quedo el vehículo después de la colisión, porque dependiendo de esta información la funcionalidad deberá de ser diferente, está información será enviada desde el módulo posición del vehículo y la primera instrucción que se realizara será el leer esta información., Este

módulo utiliza un sensor de fuerza y un servomotor, el sensor de fuerza simulará que algún ocupante se encuentra inconsciente y está ejerciendo presión sobre un cristal de la ventana, el sistema deberá de ser capaz de leer esta información y determinar si la ventana podrá ser bajada, entonces esta función es simulada con el servomotor el cual bajará o dejará arriba la ventana, dependiendo de las variables de posición del vehículo y de la presión de cada ventana.

3.4.4. PRUEBAS DEL SISTEMA

Para la documentación de las pruebas se generó una plantilla de pruebas tomando como base la plantilla propuesta en PSP, con algunas modificaciones, con esta plantilla se pretende tener una identificación y control de cada prueba, ver figura 23. En esta sección se presentará de forma parcial las pruebas del módulo de Posición del Vehículo y Bajar Ventanas, en el anexo H se tienen todas las pruebas realizadas.

Nombre		Fecha		Programa	
Nombre /Numero de Prueba					
Objetivo de la Prueba					
Descripción de la Prueba					
Condiciones de la Prueba					
Resultados Esperados					
Resultados Actuales					

Evidencia Fotográfica de la ejecución

Figura 23. Plantilla pruebas.

3.4.4.1. PRUEBAS POSICIÓN DEL VEHÍCULO

Nombre	GRG	Fecha	28/09/2017	Programa	PosicionVehiculo.ino
Nombre /Numero de Prueba	GRG/1				
Objetivo de la Prueba	Detectar posición del vehículo.				
Descripción de la Prueba	Identificar la posición del vehículo y almacenar esta información en un tipo de memoria que no se borre cuando se desconecte la tarjeta Arduino de la energía. Porque esta información será utilizada más adelante para otras funcionalidades del sistema.				
Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO. • Tener las conexiones correctas entre el sensor acelerómetro giroscopio MPU6050 a la tarjeta Arduino UNO y esta a su vez conectada a la computadora. • Abrir el monitor serial del IDE de Arduino para mostrar datos de la ejecución. • Configurar el monitor serial a 115200 baudios.				
Resultados Esperados	(1) Todas las Llantas Arriba (valores de Y, P y R)				
Resultados Actuales	(1) Todas las Llantas Arriba (Y: 68.26, P: 9.87, R: 6.53)				
Evidencia Fotográfica de la ejecución					

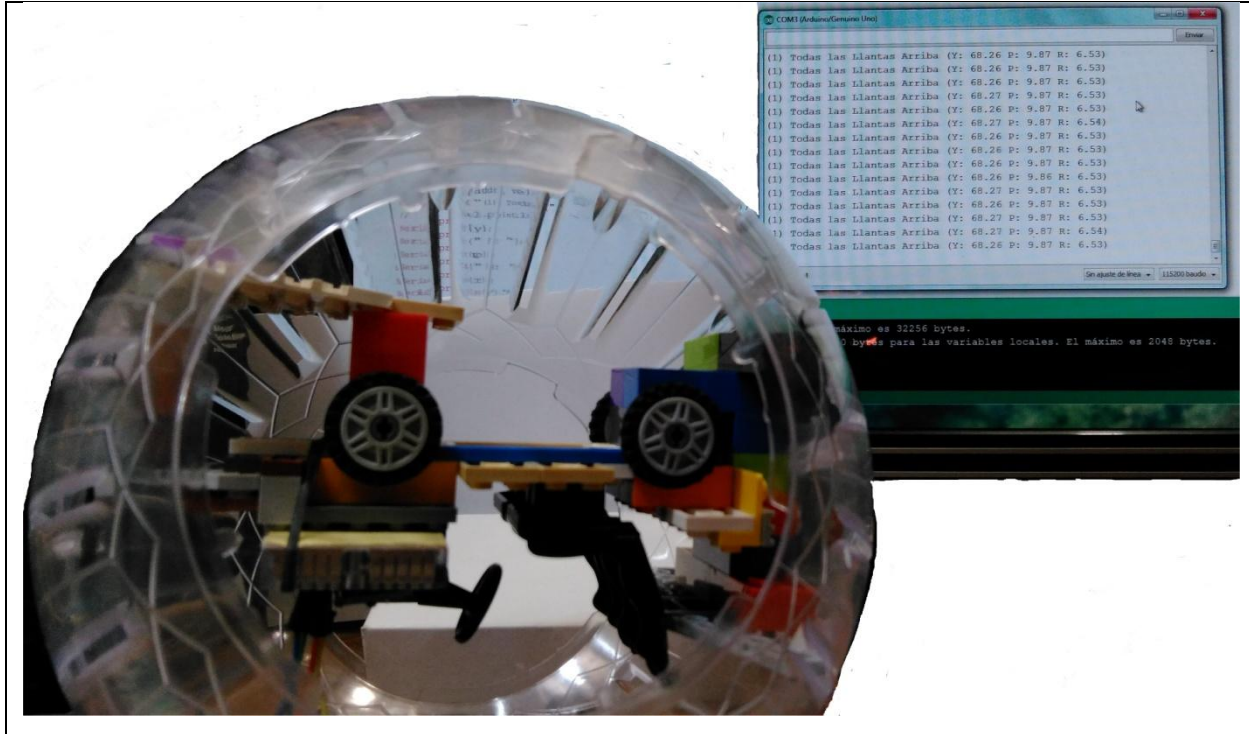


Figura 24. Prueba detectar posición del vehículo, todas las llantas arriba.

Nombre	GRG	Fecha	28/09/2017	Programa	PosicionVehiculo.ino
Nombre /Numero de Prueba	GRG/2				
Objetivo de la Prueba	Detectar posición del vehículo.				
Descripción de la Prueba	Identificar la posición del vehículo y almacenar esta información en un tipo de memoria que no se borre cuando se desconecte la tarjeta Arduino de la energía. (Porque esta información será utilizada más adelante para otras funcionalidades del sistema).				
Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO. • Tener las conexiones correctas entre el sensor acelerómetro giroscopio MPU6050 a la tarjeta Arduino UNO y esta a su vez conectada a la computadora. • Abrir el Monitor Serial del IDE de Arduino para mostrar datos				

	de la ejecución. • Configurar el Monitor Serial a 115200 baudios.
Resultados Esperados	(3) Llantas Traseras Arriba (valores de Y, P y R).
Resultados Actuales	(3) Llantas Traseras Arriba (Y: 104.46, P: 66.73, R: 5.64).

Evidencia Fotográfica de la ejecución.

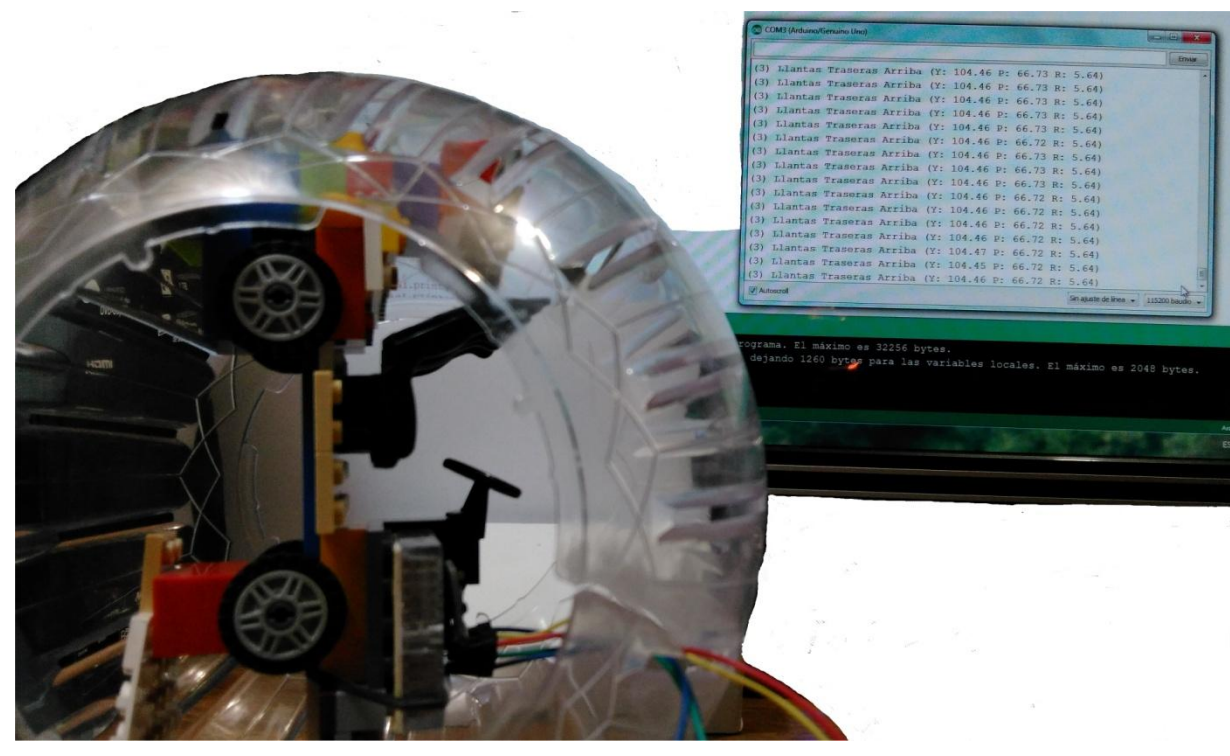


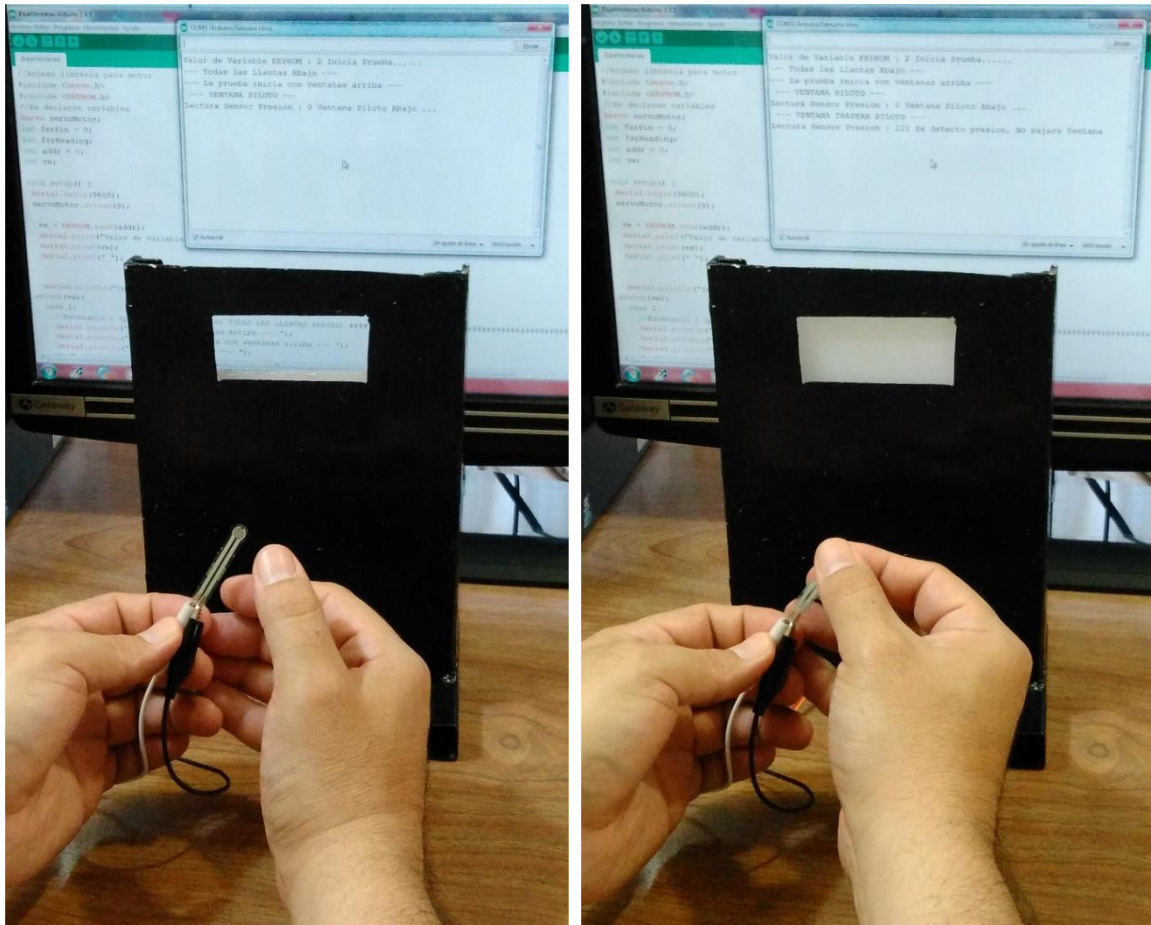
Figura 25. Prueba detectar posición del vehículo, llantas traseras arriba.

3.4.4.2. PRUEBAS BAJAR VENTANAS

Nombre	GRG	Fecha	01/10/2017	Programa	BajarVentanas.ino
Nombre /Numero de Prueba	GRG/7				
Objetivo de la Prueba	Simulación del bajado de ventanas del vehículo.				
Descripción de la Prueba	Bajar las ventanas del piloto, trasera piloto, copiloto y trasera copiloto dependiendo de la posición del vehículo previamente detectado y de la presión detectada en las ventanas.				
Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO • Tener las conexiones correctas entre el sensor de fuerza FSR400 y el Servo Motor Micro Servo 9g SG90 Tower Pro a la tarjeta Arduino UNO y esta a su vez conectada a la computadora. • Abrir el monitor serial del IDE de Arduino para mostrar la ejecución. • Configurar el monitor serial a 9600 baudios.				
Resultados Esperados	Todas las llantas abajo. Ventana piloto. No detecta presión, abajo. Ventana trasera piloto. Detecta presión, arriba. Ventana copiloto. No detecta presión, abajo. Ventana trasera copiloto. Detecta presión, arriba.				
Resultados Actuales	--- Todas las Llantas Abajo --- --- Ventana Piloto --- Lectura Sensor Presión: 0 Ventana Piloto Abajo ...				

	--- Ventana Trasera Piloto --- Lectura Sensor Presión: 221 Se detectó presión, No bajara Ventana. --- Ventana Copiloto --- Lectura Sensor Presión : 0 Ventana Copiloto Abajo ... --- Ventana Trasera Copiloto --- Lectura Sensor Presión: 119 Se detectó presión, No bajara Ventana.
--	--

Evidencia Fotográfica de la ejecución



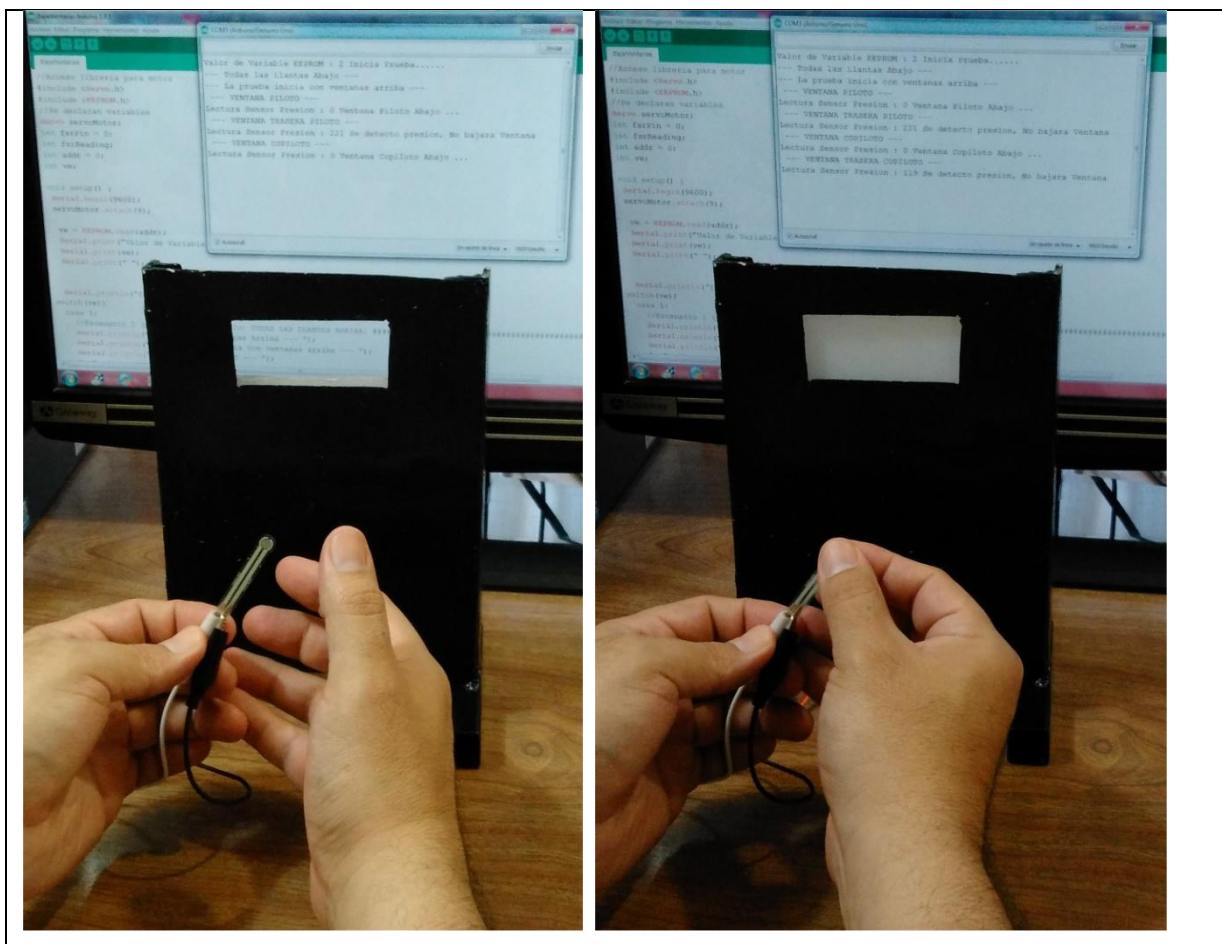


Figura 26. Prueba bajar ventanas, con posición de vehículo: todas las llantas abajo.

Nombre	GRG	Fecha	01/10/2017	Programa	BajarVentanas.ino
Nombre de Prueba	GRG/8				
Objetivo de la Prueba	Simulación del bajado de ventanas del vehículo.				
Descripción de la Prueba	Bajar las ventanas del piloto, trasera piloto, copiloto y trasera copiloto dependiendo de la posición del vehículo previamente detectado y de la presión detectada en las ventanas.				

Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO • Tener las conexiones correctas entre el sensor de fuerza FSR400 y el Servo Motor Micro Servo 9g SG90 Tower Pro a la tarjeta Arduino UNO y esta a su vez conectada a la computadora. • Abrir el Monitor Serial del IDE de Arduino para mostrar la ejecución. • Configurar el Monitor Serial a 9600 baudios.
Resultados Esperados	Llantas lado piloto arriba. Ventana piloto. Abajo. Ventana trasera piloto. Abajo. Ventana copiloto. Arriba. Ventana trasera copiloto. Arriba.
Resultados Actuales	--- Llantas Lado del Piloto hacia arriba --- --- Ventana Piloto --- Ventana Piloto Abajo ... --- Ventana Trasera Piloto --- Ventana Trasera Piloto Abajo ... --- Ventana Copiloto --- Ventana Copiloto Arriba ... --- Ventana Trasera Copiloto --- Ventana Trasera Copiloto Arriba ...
Evidencia Fotográfica de la ejecución.	



Figura 27. Prueba bajar ventanas, con posición de vehículo: llantas lado piloto arriba.
 En el Anexo G se encuentran todas las pruebas realizadas.

3.4.4.3. IMPLEMENTACIÓN

Los sistemas enfocados a la seguridad en el sector automotriz en un ambiente apegado a la realidad requieren de tener todos los componentes necesarios como sensores automotrices, *ECUS*, instalación de un *CAN BUS*, actuadores, etc. Para fines de investigación se simuló estos escenarios en un ambiente más manipulable, mediante tecnología Arduino UNO y sensores giroscopio, sensor de fuerza, *protoboard*, mini *protoboard*, resistencia, servo motor, *jumper*s, computadora personal y esfera de hámster, con todos estos elementos y a través de los documentos generados se desarrolló el código para instalar la aplicación a la tarjeta Arduino UNO y establecer la comunicación y control de los sensores, así como del servo motor.

3.5. ASPECTOS ADMINISTRATIVOS DEL SISTEMA

Haciendo referencia a las recomendaciones de la norma internacional ISO/IEC/IEEE 15288, mencionadas en el Marco Teórico de este documento de tesis y en específico al apartado 6.3.1 Proceso para la planeación del proyecto, esta sección contempla muchos de los aspectos administrativos del sistema que serán descritos a continuación, con las diferentes actividades realizadas en la elaboración del proyecto de tesis.

Proceso para planeación del proyecto.

- Salidas
 - Objetivo General, planteado en el capítulo uno de este documento que a la letra dice:

Construcción de un sistema de software de control automotriz que facilite el auxilio a los vehículos colisionados después de que suceda un accidente, mediante el uso de un sistema de software de control de misión crítica para eventos post-colisión,

con el fin de agilizar que las personas que se encuentren dentro puedan salir con facilidad y las personas que se encuentran fuera puedan auxiliar más rápido y con mayor facilidad.

- Los Roles y responsabilidades.

El tesista es el encargado del desarrollo de todas las actividades del proyecto con la guía y supervisión del asesor.

- Recursos y servicios necesarios para lograr los objetivos.

Se me permitió el acceso al laboratorio de Ingenierías de la UPAEP para tener sesiones de trabajo con el asesor y con compañeros estudiantes de diferentes disciplinas lo cual sirvió de mucha ayuda para el desarrollo del sistema.

Todos los componentes electrónicos fueron comprados para el desarrollo del sistema como La tarjeta Arduino UNO, los sensores giroscopio y de fuerza, así como el servo motor, *jumper*s, caiman *protoboard*, mini *protoboard*, resistencias y los demás componentes para la implementación del sistema en el prototipo.

- Planes de ejecución del proyecto.

Más adelante en este capítulo, en la sección 3.6.3 documento de planes contra realidad, de este capítulo se describe el plan que fue elaborado para la creación del sistema.

- Actividades y tareas, sección definición de proyecto.

- Identificación de objetivo y restricciones.

El objetivo general ya fue mencionado anteriormente, las restricciones, se mencionan en diferentes secciones del documento, en el capítulo uno, llamado Propósito y Organización, apartado 1.8 Alcances y Limitaciones, y en el capítulo tres, apartado 3.2.1.1.5 Riesgos del Negocio y 3.2.1.3. Alcance y Limitaciones, en específico del documento de Visión y Alcance.

- Definir alcance.

Para establecer el alcance del prototipo se realizó una reunión de trabajo con el asesor para establecer los casos de uso que se desarrollaría en el prototipo, se eligieron la posición del vehículo y el bajar las ventanas, cabe mencionar que una razón por la cual se tomó esta decisión es porque para desarrollar la funcionalidad de bajar ventanas se requiere saber la posición del vehículo después de la colisión.

- Definir y mantener un ciclo de vida.

El modelo de desarrollo que más se adapta a este tipo de proyecto es el mencionado en el capítulo dos, Basado en componentes ya que aparte de utilizar un ciclo en espiral, tiene como principal objetivo la integración de componentes y re uso.

- Establece una estructura de desglose de trabajo, basado en la evolución de la arquitectura del sistema.

En este capítulo en el apartado de Arquitectura del sistema se presenta el modelo de 4 vistas, con sus respectivos diagramas, y el diseño detallado con el pseudocódigo de cada caso de uso que fue desarrollado.

- Actividades y tareas, sección plan de proyecto y administración técnica.
 - Define y mantiene un calendario de proyecto, basado en administración de objetivos técnicos y estimación de trabajo.

En este capítulo más adelante se presenta en el apartado 3.6.1. La lista de Tareas y en el apartado 3.6.2. Estimación de esfuerzo para llevar a cabo las tareas, y en el apartado 3.6.5. El calendario de entregables.

- Define costos.

Para la elaboración del prototipo se presenta la siguiente lista de costos, ver tabla 2.

Artículo	Precio
Arduino UNO	330
Sensor acelerómetro y giroscopio MPU 6050	80
Sensor de fuerza FSR400	220
Servo motor micro Servo 9g SG90 Tower Pro	50
Protoboard	70
Mini protoboard	25
Jumpers	50
Resistencia 10 K	5
Caimanes	14
Estructura Metálica para simular ventana	250
Esfera para hámster	250
Pieza de acrílico	40
Broche baco	10
Tornillo con turca	5
Alfiler	1
Liga	1
Total	1401

Tabla 2. Tabla de costos

- Define infraestructura y servicios requeridos

En este capítulo en el apartado 3.4.1. Elementos utilizados se muestra la infraestructura donde se desarrolló el prototipo para probar la funcionalidad propuesta del sistema.

- Actividades y tareas, activar proyecto.
 - Obtener autorización del proyecto.

La aceptación del tema de tesis para desarrollar este documento y el sistema de software de misión crítica para seguridad automotriz en eventos post-colisión fueron otorgados por el Centro Interdisciplinario de Posgrados.

- Implementar plan del proyecto.

El conjunto de todas las actividades, investigación, documentos, diagramas, generación del prototipo, presentadas en este trabajo de tesis.

3.6.1. LISTA DE TAREAS

En esta sección se enlistan las principales tareas que realizaron en la elaboración del sistema

- Se crearon los siguientes documentos.
 - Visión y Alcance
 - Concepto de Operaciones
 - Especificación de Requerimientos
- Se crearon los siguientes diagramas
 - Requerimientos
 - Clases
 - Secuencia
 - Despliegue
- Investigación entorno Arduino UNO
- Investigación sensor giroscopio
- Investigación sensor de fuerza
- Investigación servo motor
- Creación de elementos para instalar los sensores en el prototipo
- Creación de Pseudocódigo
- Codificación.
- Pruebas.

3.6.2. ESTIMACIÓN DE ESFUERZO PARA LLEVAR A CABO LAS TAREAS

Para el desarrollo del sistema se realizaron una serie de tareas desde la elaboración de diferentes documentos, generación de diferentes tipos de diagramas, investigación previa para la generación del prototipo y el desarrollo del mismo. De estas actividades se estimó el esfuerzo en tiempo utilizado en estas tareas las cuales son descritas en la siguiente tabla, ver tabla 3.

No	Actividad	Estimación
1	Documento Visión y Alcance	3 semanas
2	Documento de ConOps (Identificación de 10 escenarios)	1 mes
3	Documento SRS (10 casos de uso)	1 mes, quince días
4	Generación de diagramas	1 mes
5	Pseudocódigo de cada caso de uso	1 semana
6	Establecer Alcance del sistema	1 semana
7	Investigación de los sensores disponibles y que pueden ser utilizados en el prototipo para Arduino	2 semanas
8	Desarrollo del sistema	3 semanas
9	Pruebas del sistema	1 semana

Tabla 3. Tabla de actividades.

3.6.3. DOCUMENTO DE PLANES CONTRA REALIDAD

Los planes son importantes para creación de sistemas, nos ayudan organizar las actividades a realizar, incluso nos ayudan a organizarnos de manera individual y de manera grupal cuando intervienen más personas en la creación del sistema, también forma parte de las recomendaciones de la norma ISO/IEC/IEEE 15288 para el llevar en control de los proyectos.

Los planes pueden ser simples o tan complejos como uno los requiera, aparte de organizar y controlar nuestro trabajo. A continuación se muestran los planes contra la realidad realizado en este trabajo de tesis, ver tabla 4.

Actividad	Orden de actividades del Plan	Orden de realización en la Realidad
Documento Visión y Alcance	1	5
Documento de ConOps	2	2
Documento SRS	3	3
Investigación y avance del documento de patente	4	1
Creación del artículo presentado en CONISOFT '17	5	6
Investigación de las tecnologías para la creación del prototipo	6	4
Arquitectura del Sistema	7	7
Diseño detallado	8	9
Creación de los componentes físicos necesarios para el prototipo	9	8
Desarrollo del prototipo	10	10
Pruebas del prototipo	11	11

Tabla 4. Tabla de plan contra realidad.

3.6.4. TIEMPOS DE DESARROLLO

Actualmente el desarrollo de sistemas tiene diferentes entornos y particularidades, existe desarrollo para desktop, web, la nube, dispositivos móviles, embebido y otros, por esta razón se tiene que realizar cierta investigación previa para poder conocer las diferentes

especificaciones necesarias en el ambiente requerido según el tipo de desarrollo en el cual se programe.

Para un ambiente embebido, se tienen diferentes actividades a considerar, en un ambiente de investigación, existen diferentes tarjetas que permiten controlar sensores e introducir código como Arduino, Raspberry, PLC, entre otras, para elaborar el prototipo presentado en este trabajo de tesis, se eligió Arduino. Independientemente de la tarjeta, se requiere instalar el IDE y establecer la conexión con la tarjeta Arduino, después de esto se requiere investigar los diferentes sensores que se necesitan para el proyecto, en este caso se necesita de un giroscopio, un sensor de fuerza y como actuador se utilizó un servo motor de 0 a 180 grados, estos dispositivos adicionales deberán ser instalados en la tarjeta Arduino para eso se requiere de otros componentes de conexión como *protoboard*, mini *protoboard*, *jumper*s, caimanes e incluso resistencias específicas si son requeridas, después de tener estos componentes se requiere de conocer las diferentes especificaciones de conexión entre la tarjeta y el sensor en particular, para poder desarrollar el código necesario para después instalarlo en la tarjeta, cabe mencionar que algunos sensores requieren de librerías propias, con lo cual es necesario descargarlas del proveedor y hacer la debida referencia y también si se requiere utilizar re uso de código. Adicionalmente a las especificaciones descritas anteriormente, hace falta un entorno en particular para instalar y probar los diferentes sensores y simulación de actuadores como fue la adaptación de la bola de hámster para instalar dentro el giroscopio y probar las diferentes posiciones del vehículo y el diseño de la estructura metálica con el acrílico y la adaptación del servomotor para la simulación de las ventanas. Después de explicar las diferentes particularidades encontradas y consideradas para la generación del desarrollo se mencionan los siguientes tiempos utilizados, ver Tabla 5.

No	Actividad	Tiempo
1	Investigación (Tipo de sensor o actuador, conexión, especificaciones particulares)	1 mes
2	Diseño de entorno para adaptar sensores y actuadores	1 semana
3	Pseudocódigo para identificar la posición del vehículo	2 días

4	Desarrollo para identificar la posición del vehículo	3 días
5	Pseudocódigo para bajar ventanas	2 días
6	Desarrollo para bajar ventanas	3 días

Tabla 5. Tiempos de desarrollo.

3.6.5. CALENDARIO DE ENTREGABLES

En la siguiente tabla se enlistan los diferentes entregables generados en este trabajo de tesis, ver tabla 6.

Entregable	Fecha
Documento de Visión y Alcance	5 de febrero de 2017
Documento de ConOps	1 de marzo de 2017
Documento SRS	25 de abril de 2017
Diagrama de secuencia	22 de julio de 2017
Pseudocódigo	31 de julio de 2017
Desarrollo del sistema	27 de septiembre de 2017
Pruebas del sistema	8 de octubre de 2017

Tabla 6. Tabla de entregables.

Cabe comentar que paralelo a este documento se trabajó en el desarrollo de un Artículo que fue presentado en el *5th edition of the International Conference in Software Engineering Research and Innovation* (CONISOFT'17).

CAPÍTULO 4. RESULTADOS

Mediante el desarrollo de la presente tesis se crearon diferentes documentos, diagramas casos de uso e incluso un Artículo el cual fue presentado en CONISOFT'17.

Entre los diferentes documentos creados, los cuales tienen gran importancia en el estudio de la ingeniería de software, partiendo desde un enfoque general, hasta un enfoque específico, mismos que fueron creados para documentar el Sistema de Software de control, de misión crítica, para eventos Post-colisión. Se tienen los siguientes documentos:

- Visión y Alcance.
- Concepto.
- Especificación de Requerimientos.
- Patente (avance parcial).
- En especial y con gran satisfacción, al contar con una orientación adecuada se presentó el **Artículo “Software Requirement Specification for the Automotive Sector: The case of a Post-Collision Event Control System”** el cual fue aceptado en el *Track 1: Scientific papers* para su publicación en la IEEE en CONISOFT'17, ver Anexo D.

Se crearon los siguientes casos de uso:

- CU1. Evalúa colisión.
- CU2. Detecta movimiento.
- CU3. Tiempo razonable sin movimiento.
- CU4. Posición del vehículo.
- CU5. Deshabilita fuente primaria de energía.
- CU6. Habilita fuente secundaria de energía.
- CU7. Enciende luces internas.
- CU8. Abre seguros de puertas.
- CU9. Bajar ventanas.

- CU10. Desancla cinturones de seguridad.

Se crearon diferentes tipos de diagramas:

- De requerimientos.
- Secuencia, para cada caso de uso antes mencionado.
- Despliegue.
- Clases.

Se realizó un prototipo con el fin de implementar parcialmente la funcionalidad del sistema, utilizando la tecnología Arduino para probar toda la lógica desarrollada mediante el uso de sensor de giroscopio, sensor de fuerza y servo motor para hacer la simulación del control de las ventanas. La funcionalidad que se decidió desarrollar en el prototipo es la del caso de uso 4 Posición del vehículo y la del caso de uso 9 Bajar ventanas, todo en un ambiente de investigación, quedando pendiente un ambiente apegado a la realidad mediante el uso de componentes automotrices para trabajo a futuro.

Toda la información mencionada en este capítulo que forma parte del trabajo de tesis se encuentra en la sección de Anexos.

CAPÍTULO 5. ANÁLISIS Y DISCUSIÓN

5.1. CONCLUSIONES Y RECOMENDACIONES

Actualmente los sistemas de seguridad automotriz son de gran importancia y han sido estudiados, sobre todo en sus clasificaciones activos y pasivos, pero los sistemas de seguridad post-colisión, han sido poco estudiados.

Los sistemas de misión crítica como es el caso de los sistemas de seguridad automotriz, cuentan con diferentes recomendaciones de los diferentes estándares de calidad ISO 26262 e ISO/IEC/IEEE 15288, mediante el enfoque basado en procesos.

La utilidad de realizar estudios de seguridad automotriz para eventos de post-colisión se busca contribuir en la investigación y mejora en las diferentes acciones que faciliten la ayuda y auxilio a las personas que han sufrido un accidente y con esto se disminuya el índice de fatalidad por no contar con los mecanismos necesarios para el auxilio, de manera inmediata.

5.2. TRABAJO A FUTURO

Como trabajo a futuro se requiere continuar con el desarrollo de la funcionalidad del Sistema de Software de control, de misión crítica, para eventos Post-colisión., Considerando el resto de los casos de uso: a) CU1. Evalúa colisión, b) CU2. Detecta movimiento, c) CU3. Tiempo razonable sin movimiento, d) CU5. Deshabilita fuente primaria de energía, e) CU6. Habilita fuente secundaria de energía, f) CU7. Enciende luces internas, g) CU8. Abre seguros de puertas, h) CU10. Desancla cinturones de seguridad. Los casos de uso anteriores y los ya desarrollados en este trabajo de tesis agrupan a los requerimientos funcionales.

Se considera la definición de los requerimientos no funcionales.

Otra actividad que se debe considerar realizar es la migración del Sistema de Software de control, de misión crítica, para eventos Post-colisión, esté requerirá un ambiente de operación

apegado a la realidad, elaborado con piezas, usadas en un auto convencional, con el fin de que la funcionalidad sea probado en un auto real.

REFERENCIAS

- [1] México. INEGI. Estadísticas a propósito de la industria automotriz. 2016. Print.
- [2] México. INEGI. La Industria automotriz en México 2014, Serie estadísticas sectoriales. 2014. Print.
- [3] Canada. Transport Canada. Canadian Motor Vehicle Traffic Collision Statistics 2015 Collected in Cooperation with the Canadian Council of Motor Transport Administrators, 2015, Print.
- [4] United States of America. U. S. Departament of Transportation National Highway Traffic Safety Administration. Traffic Safety Facts Crash Stats, October 2016, Print.
- [5] INEGI. 24/06/2017. Banco de Indicadores (Estados Unidos Mexicanos). Recuperado de: <http://www.beta.inegi.org.mx/app/indicadores/?ag=01&ind=6200009443#>, última visita 25 de noviembre de 2017.
- [6] GmbH. V. V. (1999), *Manual de la técnica del Automóvil 3rd ed*, Barcelona, España: Reverté
- [7] Argentina. Agencia Nacional de Seguridad Vial, Ministerio de Transporte Presidencia de la Nación, Manual del Conductor para el Curso de Seguridad Vial, Print.
- [8] Adrian Parvelescu, F. E. (2000). Patente no 6,081,758. Estados Unidos.
- [9] Gardner. N. J. (1969), Patente no 3,479,767. Estados Unidos.
- [10] Weber. H. J. (1996), Patente no 5,574,315. Estados Unidos.
- [11] Balaraju. S. Riyazuddin. K. “Car Post Crash Analysis & Emergency Rescue Alert System”, International Journal of Advanced Research in Computer Science and Software Engineering, June 2016.
- [12] Augenstein. J. F. Bahouth. G. T. “Pos-crash Support Systems”, Springer, 2012.
- [13] Findlay Volkswagen, <http://www.findlayvw.com/blog/what-does-vw-automatic-post-collision-braking-system-do/>, última visita 25 de noviembre de 2017.
- [14] Hassan. M. T. Z. Meguid. S. A. “Effect to seat belt and head restraint on occupant’s response during rear-end collision”, Springer Science+Business Media Dordrecht, April 2017.

- [15] Cisneros. J. R. A. Fernandez. C. A. F. “Especificación de Requerimientos para el Desarrollo de Software Automotriz en México”, Revista Latinoamericana de Ingenieria de Software, 2015.
- [16] Lee. C. S. Lin. C. C. “Specification of Software Safety Requirements Acording to ISO 26262 Standar for an Electric Vehicle Motor Control System”. The 14th IFToMM World Congress, Taipei, Taiwan, October 25-30, 2015.
- [17] Pitchford. M. “Cost-Effective Unit Testing and Integration in Accordance with ISO 26262”, Electronic Desing, Engineering Essentials, May 2017.
- [18] “ISO/IEC/IEEE 15288, System and Software engineering – System life cycle processes”, International Standar Mayo 2015.
- [19] “ISO/IEC/IEEE 15288, System and Software engineering – System life cycle processes – Processes in this standard”, International Standar Mayo 2015.
- [20] “ISO/IEC/IEEE 15288, System and Software engineering – 6.4.2.3 Activities and tasks”, International Standar Mayo 2015.
- [21] Pressman, R. S (2010), Ingenieria de Software un enfoque práctico Séptima edición, México, Mc Graw Hill.
- [22] Sommerville, I (2011), Ingenieria de Software Novena edición, México, Addison-Wesley.
- [23] Humphrey, W. S. (1995), A Discipline for Software Engineering, United States of America, Addison-Wesley, Print.
- [24] GmbH. R. B. (2007). Bosch Automotive Electrics and Automotive Electronics, Systems and Components, Networking and Hibrid Drive 5th Edition, Germany, Springer Vieweg. Print.
- [25] Sommerville, I (2005), Ingenieria de Software Séptima edición, España, Addison-Wesley.
- [26] Myers, G (2004), The Art of Software Testing Second edition, United States of America, John Wiley & Sons, Inc.
- [27] Lewis, W (2005), Software Testing and Continuous Quality Improvement Second edition, United States of America, Aurebach Publications.

- [28] “ISO/IEC/IEEE 15288, System and Software engineering – 6.3 Technical Management Process”, International Standar Mayo 2015.
- [29] Kafka. P. “The Automotive Standar ISO 26262 the innovative driver for enhanced safety assessment & technology for motor cars”, 2012 International Symposium on Safety Science and Technology, 2012.
- [30] México. AMIA. Cifras de febrero, Primer Semestre, 2017.
- [31] Pérez. J. M. (1998), Técnicas del Automóvil, equipo eléctrico, séptima edición, España, Paraninfo.
- [32] Halderman. J. D. (2012), Diagnosis and Troubleshooting of automotive, electrical, electronic and computer systems, sixth edition. United States of America, Prentice Hall.
- [33] Pérez. D. I. (Octubre de 2009), Simulación de un choque frontal de un vehículo contra diferentes tipos de barrera. Universidad Carlos III Madrid, Escuela Politécnica Superior.
- [34] Contreras. Y. J. (Febrero de 2013), Sistema de Seguridad en cabeceras de asiento de automóvil para disminuir efecto Whiplash en colisiones traseras. Instituto Politécnico Nacional.

Anexo A. DOCUMENTO DE VISIÓN Y ALCANCE

A.1. Requerimientos del Negocio

A.1.1. *Background*

En la actualidad existen sistemas de seguridad Activa y Pasiva. Los cuales son enfocados a prevenir accidentes antes de una colisión en el caso de los primeros y actuar durante de la colisión en el caso de los segundos. Sin embargo, hay pocos estudios de seguridad que estén enfocados en situaciones presentadas después de una colisión.

A.1.2. Oportunidad de Negocio.

Generar propuestas de comportamiento en los automóviles para eventos post-colisión y promover su inclusión en los autos.

A.1.3. Objetivos de Negocio y Criterios de Éxito.

Los criterios de éxito del sistema propuesto son:

- Minimizar el tiempo en que los ocupantes de un vehículo siniestrado reciben ayuda del exterior.
- Disminuir las estadísticas de pérdida de vidas humanas por no haber recibido auxilio de manera expedita, después de una colisión.

A.1.4. Necesidades del Cliente o de Mercado.

Se requiere de más propuestas de comportamiento de sistemas de seguridad post-colisión, que minimicen el tiempo en que los pasajeros reciban asistencia después de un choque severo. Adicionalmente, estos sistemas deben ser desarrollados bajo el estándar automotriz ISO 26262 debido a que son sistemas de misión crítica.

A.1.5. Riesgos de Negocio.

- Que el sistema no sea de interés para las diferentes armadoras.
- Que el costo de implementación incremente significativamente el costo de los vehículos.

A.2. Visión de solución

A.2.1. Visión del Producto

Que los automóviles modernos estén dotados de sistemas que minimicen los tiempos en que los pasajeros de un vehículo siniestrado reciban ayuda del exterior cuando, por sus propios medios no sean capaces de salir del automóvil colisionado.

A.2.2. Características Importantes

- Detección de colisión.
- Lectura de Sensores (Impacto, acelerómetro, giroscopio, presión, velocidad).
- Manipulación de Actuadores (Luz interior, luces intermitentes, seguros eléctricos, elevadores de ventanas, desanclar cinturones de seguridad).
- Minimizar tiempo de asistencia a ocupantes del vehículo.

A.2.3. Suposiciones y dependencias.

- Se asume que el automóvil tiene instalados el hardware necesario para poder ejecutar toda la funcionalidad.
- Se asume que el automóvil tiene una red CAN de comunicación, para transmisión de datos.
- Se asume que el sistema cuenta con alimentación eléctrica.

A.3. Alcance y Limitaciones.

- El sistema post-colisión se enfocará en acciones que permitan disminuir el tiempo en que los pasajeros reciben asistencia, después de una colisión.
- El sistema post-colisión funcionará de manera autónoma sin la intervención del usuario.

A.3.1. Limitaciones y Excepciones

El sistema post-colisión no funcionará en caso que tanto la fuente primaria de alimentación como la secundaria hayan sido deshabilitadas debido a la colisión. Los servicios que proporcionará el sistema no podrán llevarse a cabo por daños físicos de los componentes del auto, que impidan la acción de los actuadores.

A.4. Contexto del negocio

El sistema post-colisión deberá estar preparado para su implementación, ante los rápidos cambios de necesidades del mercado automotriz.

A.4.1. Ambiente de operación.

El sistema post-colisión estará instalado en los automóviles y el usuario no intervendrá en su funcionamiento. El sistema responderá a las condiciones detectadas por los sensores y ejecutará sus funciones de acuerdo a un análisis de las entradas de los mismos.

Anexo B. DOCUMENTO CONCEPTO DE OPERACIONES

B.1. Introducción

Propósito del documento de concepto.

Actualmente, uno de los principales medios de transporte es el automóvil, con 1.2 billones de autos en todo el mundo. La cantidad impresionante de vehículos implica un alto número de accidentes, por ejemplo, tan sólo en Canadá, cada año, entre 2000 y 2010, hubo un promedio de 2632 personas fallecidas, y 1.5 millones lastimados [3]. Ante este tipo de escenarios, es necesario que las personas que se encuentran dentro de un vehículo siniestrado reciban ayuda del exterior de una forma rápida y sin complicaciones. Por tal motivo el Sistema post-colisión propuesto, busca minimizar el tiempo en el que los ocupantes reciben ayuda, para ello facilitará el acceso al interior del auto, activando algunos servicios como: encender las luces del interior, encender las luces intermitentes, liberar seguros eléctricos, bajar automática y temporizadamente ventanas y desanclará cinturones de seguridad. Para ejecutar esta funcionalidad, el sistema verificará: a) Que el vehículo ya no se encuentre en movimiento, b) Cual es la posición final del auto. Este comportamiento contribuye a una asistencia expedita a las personas que se encuentran en el interior del auto siniestrado.

B.2. Concepto de producción.

El sistema será fabricado mediante tarjetas electrónicas de control las cuales tendrán software embebido. Adicionalmente, estarán habilitadas para llevar a cabo una comunicación mediante el protocolo de comunicación automotriz CAN (*Control Area Network*).

B.3. Concepto de operaciones (ConOps).

Este sistema hace lectura de diferentes sensores que determinan el estado del vehículo después de una colisión. El software de control manipulará a los actuadores de acuerdo a los requerimientos establecidos para cada escenario.

Los escenarios visualizados determinarán el comportamiento del sistema y para ello se deberán realizar las siguientes tareas:

- Evaluar colisión.

El sistema post-colisión detectará una colisión y el grado de severidad de la misma.

- Detectar estado de reposo del vehículo.

Después del impacto, el sistema identificará el momento en el que el auto alcanza un estado de reposo.

- Rango de Tiempo seguro

Cuando el vehículo ha dejado de moverse se determinará un rango de tiempo seguro en el cual el vehículo ha permanecido en reposo. Después de este rango de tiempo seguro, el sistema iniciará la manipulación de los diferentes actuadores.

- Posición del vehículo.

El comportamiento del sistema post-colisión depende de la posición final del vehículo. Por lo que se debe determinar.

- Activar la fuente secundaria de energía.

La fuente secundaria de energía se activa sólo en caso que la fuente primaria (batería) esté dañada o desconectada.

- Encender luces interiores.

En caso que la colisión sea en la noche, esta funcionalidad ayuda a los ocupantes a poder ver el interior del auto.

- Desactivar seguros de puertas.

Con esta funcionalidad se liberan los seguros de las puertas para que puedan ser abiertas desde el exterior.

- Bajar ventanas.

Existen escenarios que se deben analizar, para determinar si las ventanas se pueden bajar o no.

- Identificar presencia de ocupantes en el vehículo.

Es necesario conocer en que asientos hay ocupantes, para saber que ventanas deben ser bajadas.

- Identificar posición del Automóvil.

Es necesario saber en qué posición física quedó el auto. Las posiciones posibles son: a) Todas las llantas hacia arriba, b) Todas las llantas hacia abajo, c) Llantas traseras hacia arriba, d) Llantas delanteras hacia arriba, e) Llantas lado del conductor hacia arriba, f) Llantas lado del copiloto hacia arriba.

- Identificar presión de ventana.

Se identificará si algún pasajero se encuentra recargado en la ventana. En caso afirmativo, la ventana no se bajará.

- Desanclar cinturones de seguridad.

Los cinturones no se deberán liberar de forma automática. Para liberarlos, el sistema deberá evaluar los incisos a) al f) de la funcionalidad “Identificar posición del Automóvil”. Dependiendo de esta evaluación se determina si se liberan todos los cinturones o sólo los que apliquen.

B.4. Concepto de soporte.

El proyecto propuesto se planea implementar mediante el uso de componentes electrónicos con tecnología de Arduino, tarjeta CAN Shield, así como diferentes sensores, actuadores, capacitores y ECU automotrices.

B.5. Concepto de Eliminación.

Para desinstalar el sistema propuesto es necesario en primera instancia poder quitar todo el hardware instalado previamente, en principio ya que mucha de la programación generada en el proyecto se encuentra estrechamente vinculada a los diferentes componentes electrónicos instalados.

Anexo C. DOCUMENTO DE ESPECIFICACIÓN DE REQUERIMIENTOS SRS

C.1. Estado de la Técnica.

Se conoce el proceso de funcionamiento de sistemas de seguridad para la liberación automática del “seguro de niños”, instalado en las puertas traseras de algunos autos. Este sistema desbloquea el seguro automáticamente. Se compone de un sensor de emergencia, como puede ser el sensor de colisión de las bolsas de aire, un mecanismo de seguro de la puerta y un mecanismo para liberar el seguro de la puerta. Cuando el sensor detecta una emergencia, como puede ser una colisión, el sensor produce una señal que es enviada al mecanismo de liberación. El mecanismo de liberación desbloquea el “seguro de niños” de la puerta [8]. El sensor también puede enviar una señal para desconectar la batería del circuito eléctrico del automóvil e interrumpir la línea de alimentación de combustible del tanque hacia el motor. El sensor de emergencia, adicionalmente, puede incluir un sensor de temperatura o humo.

Adicionalmente, se conoce el proceso de un sistema automático integral, para la protección y rescate de los ocupantes de un vehículo siniestrado. El sistema está conformado por un elemento mecánico-eléctrico agregado a una Unidad de Control Electrónico (ECU). Esta ECU está conectada a la batería de reserva e iniciará comandos temporizados, tales como la activación del bloqueo de ignición de combustible, incluyendo la apertura eléctrica de las ventanas, la apertura centralizada de las puertas y la separación de los cinturones de seguridad de los anclajes. El sistema está destinado a prevenir los riesgos que puedan surgir después de un accidente tiene lugar y también facilita la extracción de los ocupantes del vehículo accidentado [10].

Se conoce también un sistema para la apertura de emergencia para puertas automotrices que incluye una manija retráctil que se extiende para que manualmente se abra o cierre una

ventana del vehículo, un primer piñón de engranaje está conectado a la manija retráctil, un segundo piñón de engranaje que se engancha con el primer piñón [9].

Existe otro sistema automotriz que en un evento de choque, de manera automática e inmediata se desbloquearán los seguros y se bajarán las ventanas. El objetivo es que el personal médico pueda ayudar a los pasajeros sin pérdida de tiempo. Alternativamente, después de un accidente, las ventanas eléctricas del auto también podrían ser operadas mediante una manija exterior. Los vehículos automotores que tienen dispositivos que pueden ser configurados para ser activados o desactivados, tales como “seguro de niños”, son factibles de ser controlados automáticamente con el fin de permitir la liberación de puertas de un vehículo, evitando el peligro de quedar atrapado dentro de vehículos dañados.

Las deficiencias que se encuentran en los inventos actuales es que ninguna contemplan el momento oportuno para que el sistema que controla las -ventanas, seguros, cinturones de seguridad, ignición del motor, entre otros-, entre en operación, este aspecto no considerado podría poner en riesgo de daño o muerte en los pasajeros de los vehículos automotores. La mayoría de los inventos exponen que la apertura de seguros, liberación de cinturones de seguridad, bajar las cristales de las ventanas, desbloquear el seguro de niños, se hará de manera automática e inmediata. Al activarse el sistema de manera inmediata puede poner en riesgo a los pasajeros porque, puede existir el escenario que inmediatamente después de ocurrir el accidente, el auto todavía se sigue moviendo, en situaciones, incluso, el auto puede girar y voltearse, por lo que, dado que los sistemas expuestos en las patentes actuales no contemplan este escenario, podría ocurrir que, por ejemplo, si los cinturones de seguridad se liberaron y las ventanas se bajaron, los pasajeros podrían salir disparados a través de las ventanas, por lo que podrían resultar con lesiones más graves, incluso la muerte.

Una primera diferencia en el invento que se propone, se encuentra en que se toma en consideración el estado del auto en el momento posterior al impacto y se determina cuál es el momento oportuno para que el sistema inicie su operación.

Se activa el inicio del sistema en el momento en el que ya no detecta movimiento, durante un espacio de tiempo razonable. En otras palabras, se asignará un intervalo de tiempo desde que el carro dejó de moverse, cuando se cumpla el intervalo definido, el sistema entrará en funcionamiento.

La segunda diferencia consiste en que el sistema que se propone considera la posición final del vehículo siniestrado. Las posiciones finales contempladas son: cuatro llantas en el piso, vehículo sobre lado derecho, vehículo sobre lado izquierdo, cuatro llantas en el aire, cuatro llantas en el aire con frente sobre el piso y cuatro llantas en el aire con parte trasera sobre el piso. La razón por la cual se toma en cuenta la posición final se debe a que, en el caso que de manera automática e inmediata se bajaran las ventanillas, esto podría dañar a los pasajeros en caso de que el auto estuviera volteado sobre uno de sus costados y en ese costado estuviera un pasajero, éste podría ser lastimado en el escenario de que su cabeza o cuerpo estuviera apoyado sobre el cristal. Adicionalmente, si el auto estuviera con las cuatro ruedas arriba, y si de manera inmediata se liberaran los cinturones de seguridad, los pasajeros sufrirían daños al no tener un mecanismo que los mantenga unidos a los asientos. Por estos motivos, el sistema que se propone, bajará las ventanillas de los autos después de ser liberados los seguros pero, antes de bajar las ventanillas, el sistema identificará la posición del auto y con base a esta información, se determinará que ventanilla o ventanillas se bajaran.

La tercera diferencia, consiste en que los inventos previos no consideran una fuente de alimentación de respaldo en caso de que la fuente principal quede inutilizable a causa de la colisión. Bajo este escenario los sistemas propuestos analizados, se quedarían sin energía para

entrar en funcionamiento. El invento que proponemos, incluye una fuente de alimentación secundaria que proporcionará energía al sistema para que pueda operar sin problema.

C.2. INTRODUCCIÓN

La industria automotriz represento el 16.9% de las manufacturas en el 2014, por lo que ella aporta, se ubica entre las más importantes después de la industria alimentaria. Mencionado en un informe que proporciona INEGI en combinación con la AMIA (Asociación Mexicana de la Industria Automotriz. AC., En un boletín de prensa de Febrero de 2017, comenta que la venta nacional de vehículos ligeros registró los mejores niveles históricos para cualquier mes de febrero y para cualquier bimestre del año. Durante el segundo mes del año se vendieron 117,976 unidades, 6.5% por encima de las ventas en febrero de 2016 [30]. Esta industria es de gran importancia para nuestro país y no tan solo en lo económico, ha sido objeto de estudios en el tema de la seguridad, las cuales representan mucho capital humano y mucho tiempo en estos. Existen algunas investigaciones a través de las cuales generaron dispositivos como los cinturones de seguridad o las bolsas de aire que han sido implementados en los automóviles. Algunas otras investigaciones se enfocan en la disminución de efectos *whiplash* (latigazo cervical) en colisiones traseras mediante sistemas de seguridad en cabeceras del asiento de automóvil [34]. Existen otras investigaciones que por los altos índices de accidentes ocurridos en las carreteras y la incorporación de nuevas tecnologías aparte de incluir confortabilidad puedan también incluir mayor seguridad de los ocupantes y los posibles peatones implicados en la colisión, mediante el estudio de simulaciones de choque frontal contra diferentes tipos de barreras [33]. Por mencionar solo algunos.

El sistema Pos Colisión propuesto en este documento está enfocado a la ayuda de las personas que se encuentran en el vehículo el cual ha tenido una colisión severa, desde el momento de la colisión y con propuestas diferentes a los estudios mencionados anteriormente, dejando un tiempo de espera razonable después de detectar que ya no está en movimiento en vehículo, para activarse la funcionalidad, ya que si el vehículo colisionado, inmediatamente realizara

acciones como bajar ventanas o desanclando los cinturones de seguridad mientras este siga en movimiento o a lo mejor continúe girando por las características de la colisión, sería muy peligroso porque aparte de la colisión pondría en un riesgo mayor a los ocupantes. Otra característica del sistema propuesto es la fuente de energía secundaria, porque supongamos que por características de la colisión el acumulador resultó averiado o se desconectó, en este supuesto la fuente secundaria de energía permitirá que la funcionalidad de ayuda continúe sin problemas de energía.

C.2.1. Propósito

El propósito del sistema Pos Colisión es el auxiliar a las personas que están dentro de un automóvil que ha tenido una colisión. La funcionalidad del sistema se enfoca antes, durante y después de la colisión. Entre las acciones que se tienen antes de la colisión es mediante un grupo de sensores que determinan si el impacto es leve o es severo, esto con la finalidad de que el sistema no se inicie en falso o solo por un impacto leve. Durante la colisión se refiere a determinar situaciones como la posición física en que ha quedado el vehículo colisionado, así como determinar si el vehículo continua en movimiento y cuando se detecta que esta inmóvil, empieza a contar un tiempo razonable de inamovilidad. Finalmente después de la colisión y después del tiempo razonable de espera lo cual nos hace diferentes a otras propuestas, es habilitar una red secundaria de energía en caso de haberse averiado la red primaria en la colisión, Subir seguros de puertas, bajar los vidrios de las ventanas de las puertas y desanclar los cinturones de seguridad.

En este documento se especifica los diferentes requerimientos que detallan la funcionalidad del sistema Pos Colisión. Este sistema de seguridad en un entorno automotriz a pesar de hacer uso de diferentes componentes del vehículo como el CAN bus, Unidades de control, sensores y actuadores, es un sistema independiente ya que todas las acciones realizadas son controladas por esté.

C.2.2. Alcance

El sistema Pos Colisión detectará una colisión de un auto, después de esto, determinará cuando el automóvil se encuentre en estado de reposo durante un tiempo razonable, finalmente, el sistema determinará la posición final física. Con base a esta información, el sistema ejecutará una serie de acciones entre las que se encuentran: deshabilitar la fuente de alimentación primaria (batería del automóvil), habilitar fuente secundaria de energía (circuito de capacitores), encender las luces del interior, abrir seguros de las puertas, bajar ventanas y desanclar cinturones de seguridad, las últimas 2 acciones mediante un análisis de la situación física final del automóvil.

C.2.3. Definiciones, acrónimos y abreviaciones

Termino	Definición
SRS	Documento <i>Software Requirements Specification</i>
JAC	Jorge R. Aguilar Cisneros
GRG	Genaro de la Rosa Garcia
ECU	<i>Electronic Control Unit</i> . El ECU procesa la información recibida desde los sensores de acuerdo a procedimientos matemáticos definidos (algoritmos de control) [24]
CAN Bus	En 1991 CAN bus (<i>Controller-Area Network</i>) fue el primer sistema bus en ser introducido en la producción en masa de vehículos a motor. Este ha sido establecido como un sistema estándar en el sector automotriz. [24]
Sensor	Transforman magnitudes físicas o químicas

	(mayormente no eléctricas) en magnitudes eléctricas (a veces a través de etapas intermedias no eléctricas) [6]
Actuador	Son elementos de acoplamiento entre el procesamiento eléctrico de la señal (procesamiento de la información) y el procesamiento (la mecánica) [6]
Capacitor ó Condensador	Se llama condensador a dos electrodos separados por un dieléctrico. La capacidad del condensador depende de la forma geométrica de los electrodos, de su separación y de la constante dieléctrica del dieléctrico [6]
Elevadores de seguros eléctricos ó Cerraduras electromagnéticas de las puertas	Las cerraduras electromagnéticas de las puertas, permiten efectuar el bloqueo de cierre o apertura de las mismas a partir de una sola de las puertas delanteras girando la cerradura con la llave correspondiente. Igualmente puede lograrse este resultado desde el interior del vehículo accionando un interruptor situado en el tablero de mandos [31].
Elevador de ventana ó Control eléctrico de ventana	Un dispositivo mecánico que trasforma el movimiento giratorio de manivela manual de la ventana o del motor eléctrico en un movimiento vertical para subir o bajar una de las ventanas del vehículo [32].
Relevador, Relé o Relay	Un relevador es un interruptor magnético que usa una armadura móvil para controlar un circuito de alto amperaje para usarlo como un interruptor eléctrico de bajo amperaje [32].

C.3. Descripción General

El sistema Pos Colisión permite la asistencia a los ocupantes de un automóvil que sufrió una colisión, dado que lleva a cabo las acciones de: Detectar el impacto, identifica que el vehículo ya no se encuentra en movimiento y esperar un tiempo razonable para continuar con la funcionalidad, Detectar la posición física en la que quedó el automóvil, desactivar la energía primaria, habilitar una fuente secundaria de energía, encender luces internas del vehículo, subir seguros de las puertas, bajar las ventanas y desanclar cinturones de seguridad. Estas acciones se llevan a cabo antes, durante y después de un accidente.

C.3.1. Perspectiva del producto

El sistema Pos Colisión es un producto independiente, aunque hace uso de algunos componentes del automóvil el control de los sensores, actuadores y demás componentes son controlados por el sistema propuesto.

C.3.2. Funciones del sistema

Las principales funciones que realizará el sistema son:

- Detectar la colisión
- Detectar estado de reposo del automóvil

- Esperar un tiempo razonable después de que el automóvil se encuentra en reposo
- Identificar la posición del automóvil
- Deshabilitar fuente principal de energía
- Habilitar fuente secundaria de energía
- Encender la luz del interior del automóvil
- Quitar los seguros de las puertas
- Identificar la presión en los cristales de las ventanas
- Identificar la posición física final del automóvil
- Bajar los cristales de las ventanas de acuerdo a la presión detectada y a la posición del automóvil
- Desanclar los cinturones de seguridad de acuerdo a la posición del automóvil

C.3.3. Características del Usuario

El usuario no interactúa directamente con el sistema, el sistema actuará de manera independiente sin necesidad de recibir una orden del usuario.

C.3.4. Limitaciones

El sistema Pos Colisión no funcionará en caso que tanto la fuente primaria de alimentación como la secundaria hayan sido deshabilitadas por la colisión. Los servicios que proporcionará el sistema no podrán llevarse a cabo por daños físicos de los componentes del auto que

impidan, por ejemplo, que los vidrios bajen, que los seguros se desactiven o que los cinturones de seguridad se liberen.

C.3.5. Asunciones y Dependencias

Se asume que el automóvil cuenta con seguros y cristales eléctricos

Se asume que el automóvil tiene instalados todo el hardware descrito en el documento para que físicamente este posibilitado de ejecutar toda la funcionalidad

Se asume que el automóvil tiene anclaje eléctrico que permite desanclar de manera eléctrica los cinturones de seguridad

Se asume que el automóvil tiene una red-CAN bus

C.4. Especificación de Requerimientos

El sistema deberá:

1. Detectar un impacto frontal, lateral o trasero
2. Detectar la velocidad del automóvil
3. Detectar la fuerza G del automóvil
4. Detectar el movimiento del automóvil
5. Esperar un tiempo razonable cuando ya no esté en movimiento el automóvil
6. Detectar la posición del automóvil
7. Deshabilitar la fuente primaria de energía
8. Habilitar la fuente secundaria de energía
9. Encender las luces del interior del automóvil
10. Abrir los seguros de puertas del auto

11. Detectar la presión en cristales de las ventanas
12. Determinar los cristales de las ventanas que serán bajadas de acuerdo a la posición final del automóvil
13. Bajar los cristales de las ventanas
14. Determinar los cinturones de seguridad que serán desanclados dependiendo de la posición final del automóvil
15. Desanclar cinturones de seguridad

C.5. Modelado de Sistema

C.5.1. Diagrama de Requerimientos

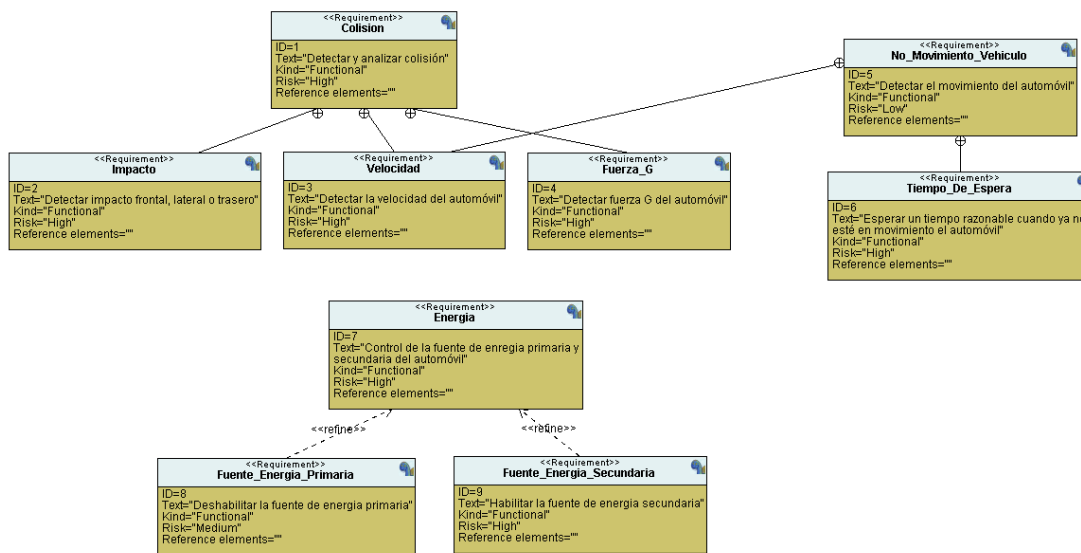


Figura 1. Diagrama de Requerimientos. Evalúa colisión y fuente secundaria de energía

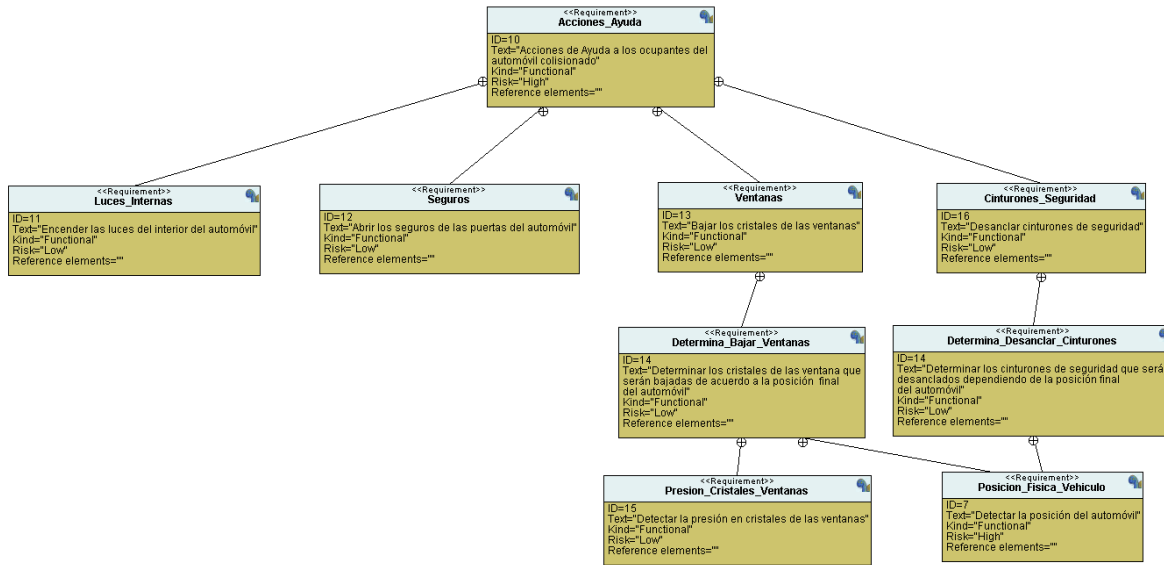


Figura 2. Diagrama de Requerimientos. Acciones de ayuda.

C.5.2. Diagrama General

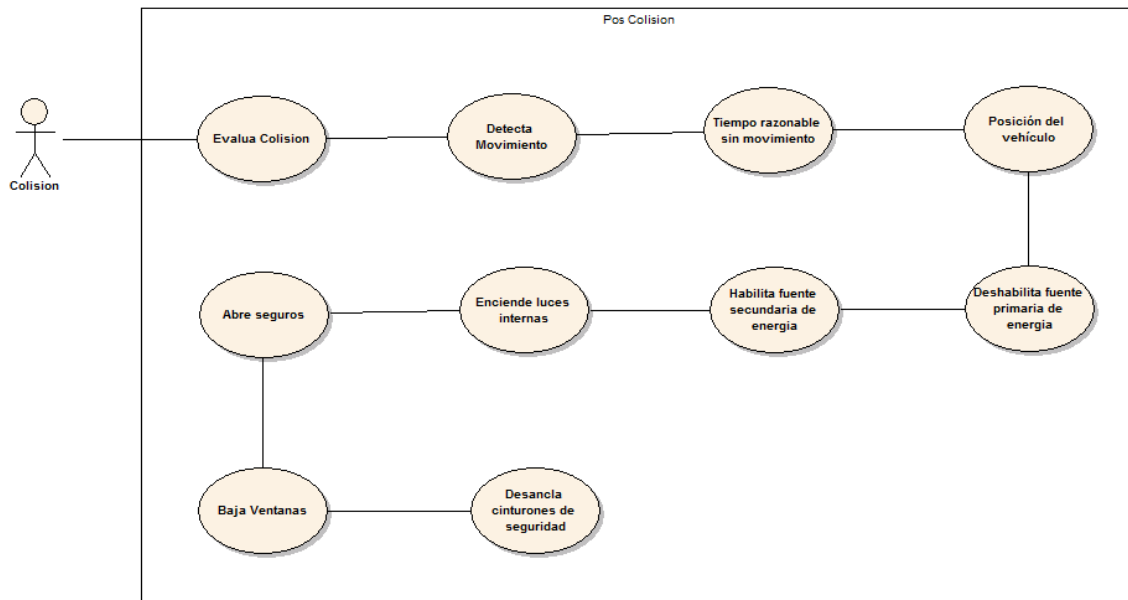


Figura 3. Diagrama general del sistema Pos Colisión

C.5.3. Casos de uso del Sistema Pos Colisión

El evaluar la colisión en una tarea primordial para determinar las condiciones que son requeridas para dar inicio al sistema pos colisión.

Caso de uso	CU1 Evalúa colisión
Actores	Sensor de impacto frontal, Sensor de impacto lateral derecho, Sensor de impacto izquierdo, Sensor de impacto trasero, Sensor de velocidad, Sensor acelerómetro.
Descripción	Identificar un impacto por los diferentes sensores de impacto, con una velocidad mayor a los 100 kilómetros por hora y una fuerza de desaceleración mayor o igual a 3G
Tipo	Primario
Incluye	Funcionalidad de los sensores
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	N/A
Flujo básico	1.- El sistema lee entrada del sensor de impacto frontal 2.- El sistema lee entrada del sensor de impacto lateral derecho 3.- El sistema lee entrada del sensor de impacto lateral izquierdo 4.- El sistema lee entrada del sensor de impacto trasero 5.- El sistema lee el sensor de velocidad 6.- El sistema lee el sensor de fuerza G 7.- El sistema procesa entradas de los sensores y evalúa las condiciones de un impacto severo para iniciar el sistema que consisten en detección de un impacto en cualquiera de los sensores de impacto, velocidad > a 100 km por hora y fuerza G >= 3. Si no se cumplen las condiciones se quedarán los sensores alertas, se pasa al paso 1
Flujo alternativo	1.1.- El sistema no puede leer el sensor de impacto frontal

	1.2.- El sistema muestra testigo en tablero 2.1.- El sistema no puede leer el sensor de impacto lateral derecho 2.2.- El sistema muestra testigo en el tablero 3.1.- El sistema no puede leer el sensor de impacto lateral izquierdo 3.2.- El sistema muestra testigo en el tablero 4.1.- El sistema no puede leer el sensor de impacto trasero 4.2.- El sistema muestra testigo en el tablero 5.1.- El sistema no puede leer el sensor de velocidad 5.2.- El sistema muestra testigo en el tablero 6.1.- El sistema no puede leer el sensor de fuerza G 6.2.- El sistema muestra testigo en el tablero
--	---

Figura 4. Caso de Uso 1. Evalúa Colisión

Una actividad que debe de ser monitoreada es la inmovilidad del auto colisionado, en la figura 5 se muestra el caso de uso 2 Detecta movimiento, en donde se monitorea el vehículo hasta identificar que ya no se encuentra en movimiento.

Caso de uso	CU2 Detecta movimiento
Actores	Sensor de velocidad
Descripción	Monitorear el sensor de velocidad hasta detectar que no envía ninguna señal, lo cual indica que el vehículo se ha detenido.
Tipo	Primario
Incluye	Funcionalidad del sensor
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU1

Flujo básico	1.- El sistema lee el sensor de velocidad 2.- El sistema procesa la entrada del sensor y evalúa. Si el vehículo continua moviéndose, se regresa al paso 1
Flujo alternativo	1.1.- El sistema no puede leer el sensor de velocidad 1.2.- El sistema muestra testigo en el tablero

Figura 5. Caso de Uso 2. Detecta movimiento

Cuando el vehículo ha tenido una colisión, existe una serie de eventos que suceden en fracciones de segundo, y una diferencia que se tiene en esta propuesta es esperar un tiempo razonable sin movimiento del vehículo, una vez que se detectó la inmovilidad ya que si se comenzara con algunas actividades de auxilio como bajar cristales o desanclar cinturones de seguridad, por mencionar algunas y el automóvil continua en movimiento o probablemente incluso girando, pondría en mayor riesgo a los ocupantes. Ver figura 6.

Caso de uso	CU3 Tiempo razonable sin movimiento
Actores	Tarjeta de control
Descripción	Esperar un tiempo razonable que probablemente será de 45 segundos
Tipo	Primario
Incluye	Funcionalidad de la Tarjeta Arduino
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU2
Flujo básico	1.- El sistema inicia cronometro con una duración de 45 2.- Cuando termina cronometro, se termina esta etapa
Flujo alternativo	N/A

Figura 6. Caso de Uso 3. Tiempo razonable sin movimiento

Es necesario conocer en qué posición física ha quedado el vehículo, ver figura 7, después de la colisión y cuando este ha dejado de moverse, ya que en base a esta información el sistema podrá analizar otras acciones de ayuda a los ocupantes. Se tienen identificadas varias posiciones indicadas en los escenarios mencionados en la sección anterior del documento Concepto de operaciones.

Caso de uso	CU4 Posición del vehículo
Actores	Sensor Giroscopio
Descripción	Detectar la posición en que quedo el vehículo
Tipo	Primario
Incluye	Funcionalidad del sensor
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU3
Flujo básico	1.- El sistema lee sensor de posición 2.- El sistema procesa la entrada del sensor y detecta la posición del automóvil 3.- El sistema almacena la información en la tarjeta de control
Flujo alternativo	1.1.- El sistema no puede leer el sensor de posición 1.2.- El sistema muestra testigo en el tablero

Figura 7. Caso de Uso 4. Posición del vehículo

En el caso de uso 5 se menciona la desconexión de la fuente primaria de energía, ver figura 8, ya que todos los componentes que tiene el sistema funcionan con energía y este es un insumo principal para que el sistema funcione, los tipos de colisiones pueden ser muy variados con características particulares de colisión a colisión.

Caso de uso	CU5 Deshabilita fuente primaria de energía
Actores	switch de encendido
Descripción	Deshabilitar la fuente primaria de energía.
Tipo	Primario
Incluye	Funcionalidad de componentes automotrices utilizados
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU4
Flujo básico	1.- El sistema apaga el switch de encendido
Flujo alternativo	N/A

Figura 8. Caso de Uso 5. Deshabilita fuente primaria de energía

En el caso de uso 6, se habilita la fuente secundaria de energía para asegurarnos que los componentes tengan este insumo para su funcionamiento, ver figura 9.

Caso de uso	CU6 Habilita fuente secundaria de energía
Actores	Condensadores
Descripción	Habilitar el funcionamiento de los diferentes condensadores instalados en los diferentes sensores, actuadores y ecu's instalados en el vehículo, con una capacidad de 12 volts y de 3 a 5 ampers con una duración de al menos 5 minutos, para lograr la funcionalidad necesaria del resto del sistema
Tipo	Primario
Incluye	Funcionalidad de los condensadores
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU 5

Flujo básico	1.- El sistema activa los capacitores de la tarjeta principal de control 2.- El sistema activa capacitores para el encendido de las luces del interior del automóvil 3.- El sistema activa los capacitores para los elevadores de seguros eléctricos 4.- El sistema activa los capacitores para los reles que complementan los elevadores de seguros eléctricos 5.- El sistema activa los capacitores para los sensores de presión de ventanas 6.- El sistema activa los capacitores para los elevadores de los cristales de las ventanas. 7.- El sistema activa los capacitores para los reles que complementan los elevadores de los cristales de ventanas 8.- El sistema activa los capacitores para el sistema de anclaje de los cinturones de seguridad
Flujo alternativo	N/A

Figura 9. Caso de Uso 6. Habilita fuente secundaria de energía

En el caso de uso 7 se encienden las luces internas del vehículo para alertar a los pasajeros que se tiene una situación de emergencia, si la hora en que ocurrió la colisión es de noche esta acción será de mayor ayuda, también puede suceder que el conductor quedo aturdido por el impacto y se encuentra desorientado o incluso inconsciente y en esos instantes no pueda activar esta funcionalidad. Ver figura 10.

Caso de uso	CU7 Enciende luces Internas
Actores	Luces internas, Interruptor de las luces
Descripción	Encender las luces internas del vehículo para alertar a los ocupantes de que sucedió una colisión

Tipo	Primario
Incluye	Funcionalidad de los componentes físicos usados
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU 5
Flujo básico	1.- El sistema enciende luces internas del automóvil
Flujo alternativo	1.1.- No enciende luces internas del automóvil 1.2.- El sistema muestra testigo en el tablero

Figura 10. Caso de Uso 7. Enciende luces Internas

En el caso de uso 8 el sistema abre los seguros de puertas, esto para que las personas que se encuentran en el vehículo, puedan ser auxiliadas de manera pronta por personas que se encuentran fuera del automóvil. Ver figura 11.

Caso de uso	CU8 Abre seguros de puertas
Actores	Actuadores de seguros, relevadores
Descripción	Abrir todos los seguros de las puertas del vehículo
Tipo	Primario
Incluye	Funcionalidad de Actuadores y relevadores
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU6
Flujo básico	1.- El sistema envía al elevador y rele para abrir el seguro de puerta del piloto 2.- El sistema envía al elevador y rele para abrir el seguro de puerta del copiloto 3.- El sistema envía al elevador y rele para abrir el seguro de

	puerta trasera del piloto 4.- El sistema envía al elevador y rele para abrir el seguro de puerta trasera del copiloto
Flujo alternativo	1.1.- El sistema detecta el mal funcionamiento en la apertura del seguro de la puerta del piloto 1.2.- El sistema muestra testigo en el tablero 2.1.- El sistema detecta el mal funcionamiento en la apertura del seguro de la puerta del copiloto 2.2.- El sistema muestra testigo en el tablero 3.1.- El sistema detecta el mal funcionamiento en la apertura del seguro de la puerta trasera del piloto 3.2.- El sistema muestra testigo en el tablero 4.1.- El sistema detecta el mal funcionamiento en la apertura del seguro de la puerta trasera del copiloto 4.2.- El sistema muestra testigo en el tablero

Figura 11. Caso de Uso 8. Abre seguros de puertas

En el caso de uso 9 el sistema realizará un análisis identificando la información obtenida en la identificación de la posición del vehículo y detectando si se tiene presión en las ventanas del automóvil, para que se pueda determinar que ventanas es posible bajar los cristales de las ventanas. Ver figura 12.

Caso de uso	CU9 Bajar ventanas
Actores	Sensor de presión de cristales de ventanas, Actuador de elevadores
Descripción	Analizar la posición en la que quedo el vehículo combinado con la presión de los diferentes cristales de las ventanas para así decidir que ventanas son libres para poder bajarlas

Tipo	Primario
Incluye	Funcionalidad de Actuadores
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU3 y CU5
Flujo básico	1.- El sistema lee sensor de presión en cristal de la ventana del piloto 2.- El sistema lee sensor de presión en cristal de la ventana del copiloto 3.- El sistema lee sensor de presión en cristal de la ventana trasera del piloto 4.- El sistema lee sensor de presión en cristal de la ventana trasera del copiloto 5.- El sistema lee la información almacenada de la posición del automóvil 6.- El sistema procesa entradas de los sensores de presión, así como la información de la posición del automóvil 7.- El sistema baja ventanas
Flujo alternativo	1.1.- El sistema no detecta presión en el cristal de la ventana del piloto. 1.2.- El sistema muestra testigo en el tablero 2.1.- El sistema no detecta presión en el cristal de la ventana del copiloto 2.2.- El sistema muestra testigo en el tablero 3.1.- El sistema no detecta presión en el cristal de la ventana trasera del piloto 3.2.- El sistema muestra testigo en el tablero 4.1.- El sistema no detecta presión en el cristal de la ventana

	trasera del copiloto 4.2.- El sistema muestra testigo en el tablero
--	--

Figura 12. Caso de Uso 9. Bajar ventanas

Caso de uso 10 al igual que el caso de uso bajar ventanas, al momento de desanclar cinturones de seguridad, el sistema deberá realizar un análisis previo antes de ejecutar la funcionalidad ya que es importante conocer la posición en la cual quedo el vehículo para determinar los posibles cinturones de seguridad que pueden ser desanclados. Ver figura 13.

Caso de uso	CU10 Desancla Cinturones de seguridad
Actores	Actuadores de cinturones de seguridad
Descripción	Analizar la posición en la que quedo el vehículo y decidir que cinturones de seguridad pueden ser desanclados
Tipo	Primario
Incluye	Funcionalidad de actuadores
Extiende	N/A
Referencia Cruzada	N/A
Depende de Caso de Uso	CU3 y CU5
Flujo básico	1.- El sistema lee la información almacenada de la posición del automóvil 2.- El sistema Analiza la posición del automóvil y determina que cinturones de seguridad serán des anclados 3.- El sistema desancla cinturón de seguridad del piloto 4.- El sistema desancla cinturón de seguridad del copiloto 5.- El sistema desancla cinturón de seguridad trasera del piloto 6.- El sistema desancla cinturón de seguridad trasera del copiloto
Flujo alternativo	3.1.- El sistema detecta el mal funcionamiento al desanclar el cinturón de seguridad del piloto

	3.2.- El sistema muestra testigo en el tablero 4.1.- El sistema detecta el mal funcionamiento al desanclar el cinturón de seguridad del copiloto 4.2.- El sistema muestra testigo en el tablero 5.1.- El sistema detecta el mal funcionamiento al desanclar el cinturón de seguridad trasero del piloto 5.2.- El sistema muestra testigo en el tablero 6.1.- El sistema detecta el mal funcionamiento al desanclar el cinturón de seguridad trasero del copiloto 6.2.- El sistema muestra testigo en el tablero
--	--

Figura 13. Caso de Uso 10. Desancla cinturones de seguridad

C.5.4.Diagrama de Clases

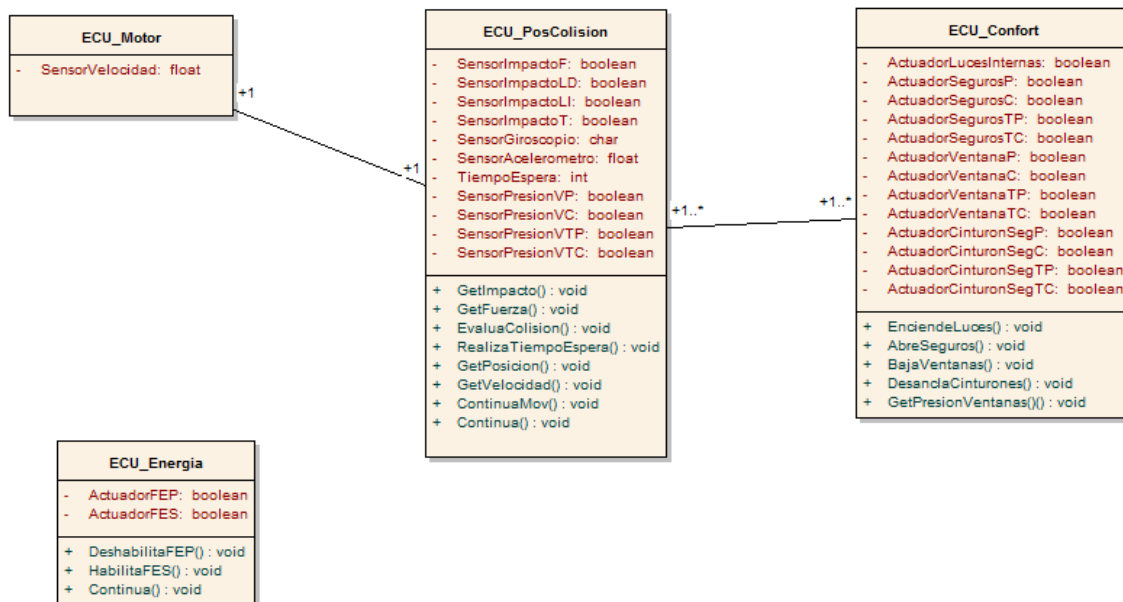


Figura 14. Diagrama de clases.

ECU_PosColision	Clase con la cual se identificarán los diferentes sensores los cuales proporcionan datos que componen la colisión así como su análisis. También proporciona información del automóvil durante y después de la colisión	
Atributos	SensorImpactoF:boolean	Es true si se detecta un impacto frontal
	SensorImpactoLD:boolean	Es true si se detecta un impacto lateral derecho
	SensorImpactoLI:boolean	Es true si se detecta un impacto lateral izquierdo
	SensorImpactoT:boolean	Es true si se detecta un impacto trasero
	SensorGiroscopio:char	Proporciona posición del automóvil
	SensorAcelerometro:float	Proporciona la fuerza G
	TiempoEspera:int	Variable que almacena el tiempo razonable de espera en que el automóvil está sin moverse
	SensorPresionVP:boolean	Es true si detecta presión en el cristal de la ventana del piloto
	SensorPresionVC:boolean	Es true si detecta presión en el cristal de la ventana del copiloto
	SensorPresionVTP:boolean	Es true si detecta presión en el cristal de la ventana trasera del piloto
	SensorPresionVTC:boolean	Es true si detecta presión en el cristal de la ventana trasera del copiloto
Operaciones	getImpacto():void	Revisa si algún sensor de impacto (Frontal, lateral o trasero) detectó un impacto.
	getVelocidad(): void	Detecta a la velocidad que viajaba el automovíl al momento de la colisión, mediante el sensor de velocidad.

	getFuerza(): void	Detecta la fuerza G de desaceleración vehículo en 1 segundo, mediante el sensor acelerómetro.
	EvaluaColision():void	Evalua la información recabada de las operaciones anteriores de esta clase y las compara con valores predeterminados para definir que la colisión dará inicio al sistema. Si no es así se queda en espera de las condiciones predeterminadas.
	ContinuaMov():void	Monitorea si el automóvil continúa en movimiento y detecta cuando éste queda parado después de la colisión.
	RealizaTiempoEspera():void	Inicia un cronometro con un tiempo de espera razonable.
	GetPosicion():void	Identifica la posición en la cual quedo el automóvil después de la colisión.
	Continua():void	Instrucción para continuar con la siguiente etapa del sistema.
Relaciones	El sistema Pos Colision tiene conectado varios tipos de sensores (Impacto, velocímetro y acelerómetro), los cuales envían información del vehículo y están conectados a una red CAN bus del automóvil	
Extención UML		

Figura 15. Descripción de la clase ECU_PosColision.

ECU_Motor	Clase con la cual se tendrá acceso al sensor de velocidad el cual se encuentra controlado por la ECU del Motor	
Atributos	SensorVelocidad:float	Detecta la velocidad del producto
Operaciones		

Relaciones	El sistema Pos Colision tiene conectado varios tipos de sensores (velocímetro), el cual envía información del vehículo y están conectados a una red CAN bus del automóvil
Extención UML	

Figura 16. Descripción de la clase ECU_Motor.

ECU_Energia	Clase con la cual se controla los asuntos de energía con los cuales trabajaran los componentes del sistema	
Atributos	ActuadorFEP():boolean	Es true si la fuente de energía primaria esta deshabilitada
	ActuadorFES():boolean	Es true si la fuente de energía secundaria esta habilitada
Operaciones	DeshabilitaFEP():void	Deshabilita la fuente de energía primaria.
	HabilitaFES():void	Habilita fuente de energía secundaria.
	Continua():void	Instrucción para continuar con la siguiente etapa del sistema.
Relaciones	El sistema Pos Colision tiene conectado varios tipos de condensadores o capacitores los cuales tienen la cualidad de de almacenar energía y estarán conectados a los diversos sensores y actuadores del automóvil	
Extención UML		

Figura 17. Descripción de la clase ECU_Energia.

ECU_Confort	Clase con la cual se determina las diferentes acciones de ayuda a los ocupantes del automóvil colisionado	
Atributos	ActuadorLucesInternas:boolean	Es true si las luces internas fueron encendidas
	ActuadorSegurosP:boolean	Es false si el seguro de la puerta del

		piloto está quitado
	ActuadorSegurosC:boolean	Es false si el seguro de la puerta del copiloto está quitado
	ActuadorSegurosTP:boolean	Es false si el seguro de la puerta trasera del piloto está quitado
	ActuadorSegurosTC:boolean	Es false si el seguro de la puerta trasera del copiloto está quitado
	ActuadorVentanaP:boolean	Es false si el cristal de la ventana de la puerta del piloto está abajo
	ActuadorVentanaC:boolean	Es false si el cristal de la ventana de la puerta del copiloto está abajo
	ActuadorVentanaTP:boolean	Es false si el cristal de la ventana de la puerta trasera del piloto está abajo
	ActuadorVentanaTC:boolean	Es false si el cristal de la ventana de la puerta trasera del copiloto está abajo
Operaciones	EnciendeLuces():void	Enciende luces del interior del automóvil
	AbreSeguros():void	Abre seguros de todas las puertas del automóvil.
	getPresionVentanas():void	Detecta presión en cristales de las ventanillas de las puertas del automóvil.
	BajaVentanas():void	Recupera la posición del automóvil y con el dato de la presión de las ventanillas analizar que ventanas pueden bajarse.
	DesanclaCinturones():void	Detecta posición del automóvil y determina que cinturones de seguridad se pueden desanclar.

Relaciones	El sistema Pos Colision tiene conectado varios tipos de sensores (Presión y giroscopio) así como actuadores (seguros eléctricos, elevadores y anclaje de cinturones de seguridad), los cuales envían información del vehículo y están conectados a una red CAN bus del automóvil
Extensión UML	

Figura 18. Descripción de la clase ECU_Confort.

C.5.5. Diagrama de Secuencia

Los diagramas de secuencia representan los escenarios del sistema, las cajas de arriba representan los objetos del sistema, las líneas verticales representan las líneas de vida de los objetos, el tiempo se encuentra en el eje de las y, las flechas solidas representan las invocaciones de los métodos de los objetos, mientras que las flechas punteadas significan el valor de regreso.

Como ya se mencionó anteriormente una característica particular de este sistema es que la interacción del sistema prácticamente se realiza con los sensores del mismo automóvil, y obviamente la funcionalidad diseñada, por tal razón al momento de tener una colisión que cumpla con los parámetros establecidos, se iniciará la una detección previa a las acciones de ayuda a los ocupantes, ya que se debe detectar que el automóvil ya no se encuentre en movimiento y así permanezca durante algún tiempo razonable, conocer la posición de cómo quedo el automóvil después de la colisión. Posteriormente a esto las acciones de ayuda inician en el sistema.

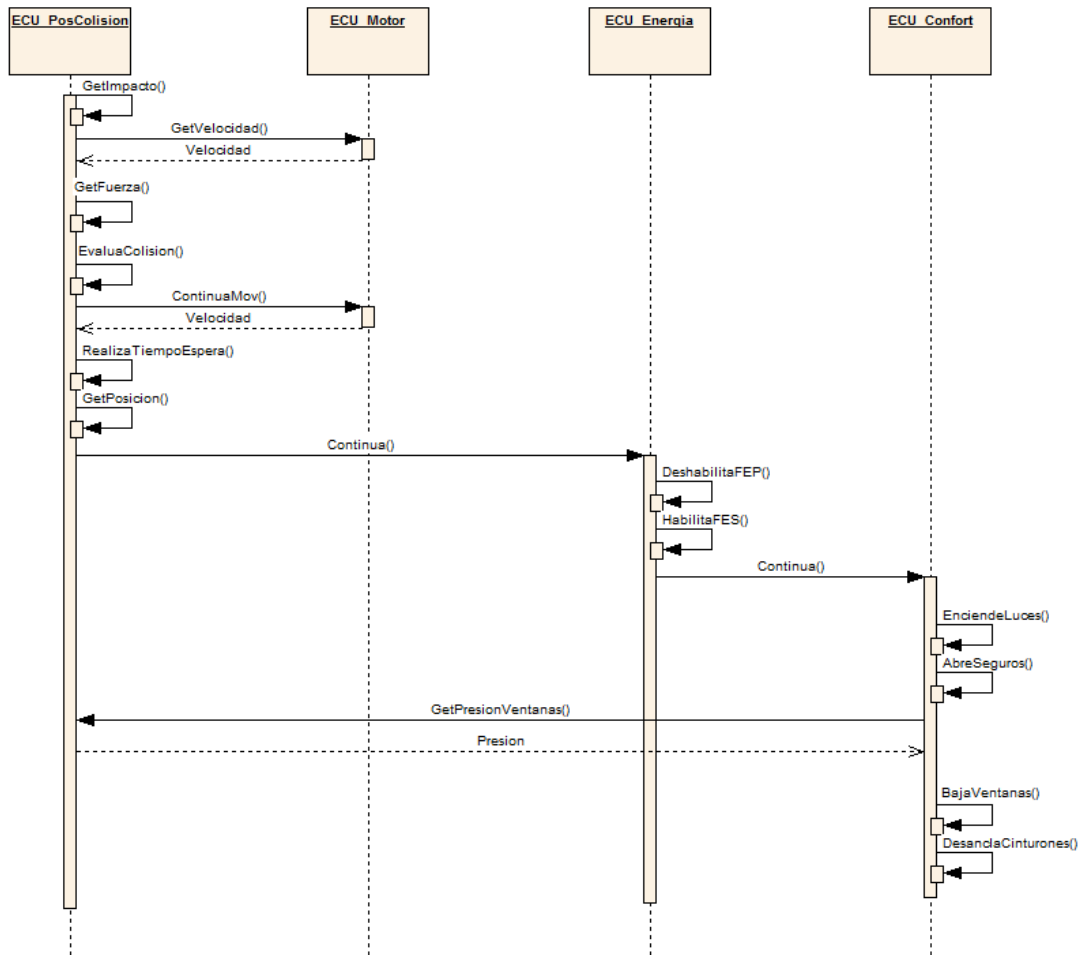


Figura 19. Diagrama de secuencia del sistema Pos Colisión.

C.5.6. Diagrama de Estado

Este tipo de diagrama ilustra el comportamiento del sistema como una colección de estados y las condiciones ocurren en el orden en el que se mueve el sistema de un estado a otro. Cada estado es una abstracción de un sistema como un todo. La transición son las flechas que indican el flujo de control desde un estado a otro.

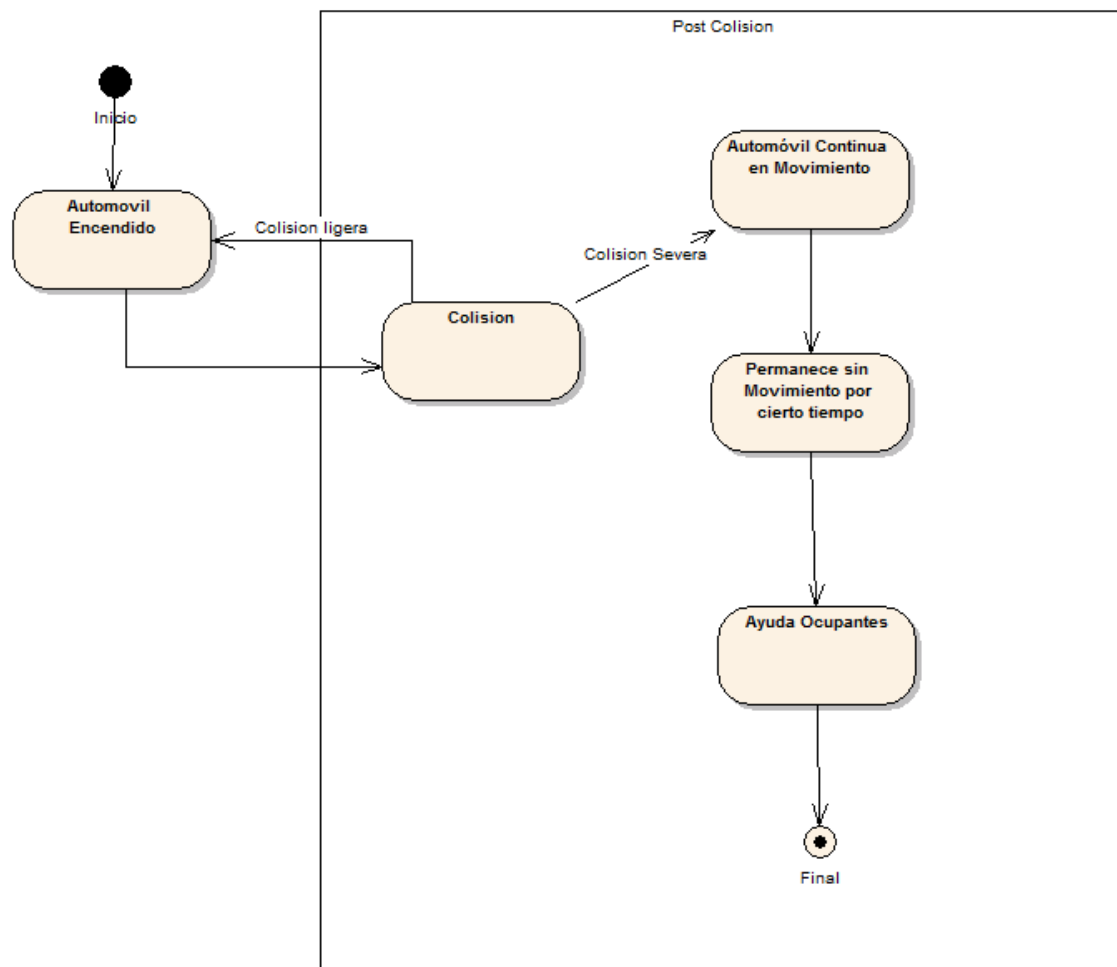


Figura 20. Diagrama de estado del sistema Pos Colisión.

C.6. Prototipo

En este documento el prototipo será representado mediante el siguiente diagrama de despliegue.

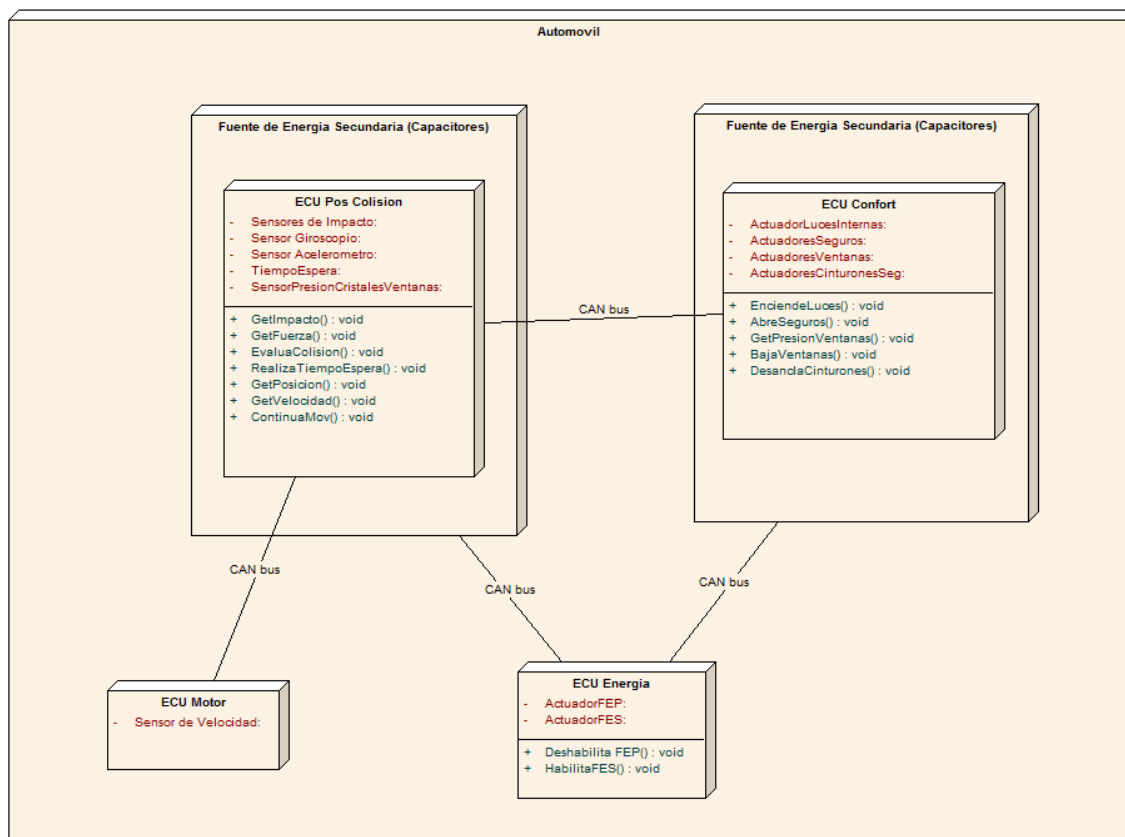


Figura 21. Diagrama de despliegue del sistema Pos Colisión.

Anexo D. ARTÍCULO PRESENTADO EN CONISOFT 2017

2017 2017 5th International Conference in Software Engineering Research and Innovation Software Requirement Specification for the Automotive Sector: The case of a Post-Collision Event Control System

Jorge Rafael Aguilar Cisneros, Genaro de la Rosa
García

Engineering Department, Graduate program in Software
Engineering
UPAEP 21 sur #1103 Col. Santiago C.P. 72310
Puebla Pue., México
jorge.aguilar@upaep.mx, genaro.delarosa@upaep.edu.mx

Carlos Alberto Fernández-y-Fernández
Computer Science Institute

U. T. M. Km 2.5 carretera Huajuapán - Acatlilma, C.P.
69000
Huajuapán de León, Oaxaca, México
caff@mixteco.utm.mx

Abstract—Safety systems in the automotive sector are very important. Among these are ABS brake systems, stability control system, and air bag systems. One group of these safety systems is oriented toward collisions and are activated at different times, having specific objectives, for example 1) systems activated before a collision have the objective of preventing impact, 2) systems activated during a collision have the objective of minimizing the physical harm to occupants, and 3) after a collision, their objective is to minimize the time between the collision and arrival of outside assistance for the vehicle's occupants. However, while type 1 systems (passive) and type 2 systems (active) are currently in many vehicles, type 3 systems have not been thoroughly researched. Therefore, this article describes a proposal of the behavioral logic of a type 3 system, which is represented through a software requirement specification document (SRS). The SRS presented satisfies some of the recommendations of the requirements of the automotive standard ISO 26262 and two of the processes used relate to the requirements of the standard ISO/IEC/IEEE 15288.

Keywords—Automotive safety systems, SRS, ISO 26262, ISO/IEC/IEEE 15288, Requirements.

I. INTRODUCTION

Automobiles have many types of systems, such as brake systems, comfort systems (operation of window regulators, seat and steering wheel adjustment, central locking), communication and information systems (parking systems, navigation systems, trip log) and safety system, just to mention a few. [1].

Within the safety systems, there are two types- active safety systems and passive safety systems. The active safety systems are those that help prevent accidents, such as steering stability systems, ABS brake systems, among others. The ABS brake system has the functions of- a) improving steering stability when braking, b) improving handling during braking, and c) optimizing braking distance. [2] See Figure 1.

On the other hand, the passive safety systems are activated without intervention by the occupants; these are, for example, seatbelt sensors, rolling protection systems, and air bag systems, among others.

The systems presented belong to category 1 and category 2, passive and active, respectively. In this article, the Software Requirement Specification (SRS) document of a system that belongs to category 3, systems that are activated after a collision and that determine the behavior of a car in post-collision events, are presented.

The article is organized as follows. In section II the Theoretical Framework is presented, in which the different elements that have been taken into account to write the SRS is presented. In section III the behavior of the post-collision system is described. This behavior is described taking into account two processes of the ISO/IEC/IEEE 15288 and the modeling of the requirements is presented taking into account the recommendations of the automotive standard ISO 26262. In section IV the conclusions are presented and finally, in section V future tasks are presented.

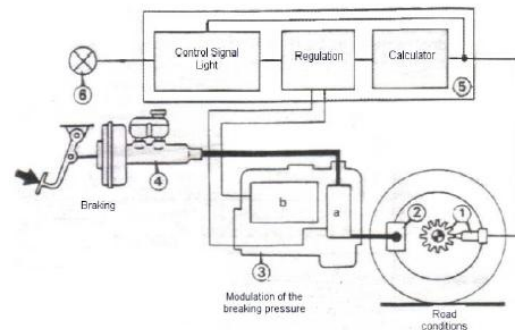


Fig. 1. Brake structure with an ABS system. [2].

II. THEORETICAL FRAMEWORK.

A. ISO 26262

The standard ISO 26262 (ISO: *International Organization for Standardization*) has as its objective to guarantee the functional security of the electrical/electronic systems of

automobiles [3]. This standard is made up of 9 sections, among which are the following:

- Section 5. Development of the product at system level.
- Section 6. Development of the product at hardware level.
- Section 7. Development of the product at software level.

In the various sections of the standard, the development and the integration of the functionality of an automotive system is considered. This standard is important since it establishes guides for development of critical mission automotive systems which are necessary due to the increase in the complexity and use of software/hardware systems in modern cars.

With the increase in the technological complexity of the automotive systems [4], features of the software and the implementation of hardware, there are growing risks of failures in the system. In this sense, the standard ISO 26262 includes a guide to decrease these risks. For such it establishes requirements and processes that should be satisfied to minimize these risks, among which are the following.

- It describes a lifecycle of automotive safety (management, development, production, operation, service and dismantling).
- It gives an approach based on the automobile-specific risk to determine the ASIL (*Automotive Safety Integrity Level*).
- It uses ASIL to specify and represent the requirements applicable of the standard ISO 26262 in order to avoid an unreasonable residual risk.
- It establishes requirements for the validation and confirmation metrics to guarantee a sufficient and acceptable level of safety.
- It establishes the requirements for the relationship with suppliers.

Functional safety is managed through the process of development (including the specification of the requirements, design, implementation, integration, verification, validation and configuration), the production processes, service and management processes. ISO 26262 is based on model V, as a model of reference for the different phases of product development.

The standard ISO 26262 is meant to be applied to systems related to safety that include one or more electrical/electronic systems (E/E) and that are installed in passenger automobiles produced in series with maximum vehicular mass of 3,500 kg. [5].

B. ISO/IEC/IEEE 15288

Another standard used in the proposal is ISO/IEC/IEEE 15288 (*Systems and Software Engineering – System life cycle Processes*), whose purpose is to give a defined group of processes to facilitate communication among buyers, suppliers and other interested parts in the lifecycle of the systems. Another purpose is that this standard may be used as a guide to

establish aspects of the business, such as methods, techniques, procedures, tools and staff training.

ISO/IEC/IEEE 15288 establishes a common framework of reference of process description for the lifecycle of the system development. It defines a group of associated processes and terminology from the engineering point of view. These processes may be applied at some level in the hierarchy of the system structure. This is carried out through the participation of all the interested parties with the establishment of a goal to achieve customer satisfaction [6]. ISO/IEC/IEEE 15288 groups the activities that can be carried out during the lifecycle of the system into four groups of processes: *Organizational Project-Enabling Processes*, *Agreement Processes*, *Project Processes* and *Technical Processes*. Each process includes its purpose, desired results, list of activities and tasks that should be carried out to achieve the results expected. The four groups of processes and their subprocesses are shown in Figure 2 [7].

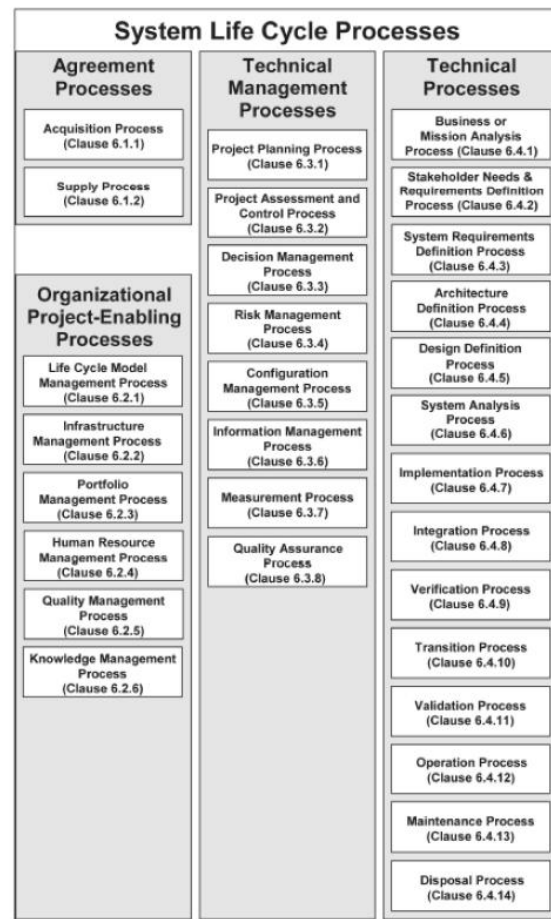


Fig. 2. Processes of the System Lifecycle. [7].

The first process, the Definition of Stakeholders' Requirements, has as its objective to help define the requirements of the system in such a way that it has the services the user needs or the services a stakeholder requires in his work context. Additionally, in this process, an identification of stakeholder, or group(s) of stakeholders, is made. Likewise, the needs, expectations and desires of the stakeholders are analyzed and transformed into a group of requirements to express the interaction that will exist between the system and their operational environment [8].

The second process of ISO/IEC/IEEE 15288 which was taken into account to structure the behavior of the system presented in this article is the Requirements Analysis process, whose purpose is to transform the stakeholders' desired or required services into the technical requirements with which the product should comply. Within this process a representation of the system that should fulfill the stakeholders' expectations is constructed.

In this article we are basing ideas on two international standards (ISO/IEC/IEEE 15288 and ISO 26262) since the development of critical mission software for the automotive sector requires the use of specific standards, architectures and models for this sector.

C. RELATED WORKS

Some studies focus on systems that are activated after the collision, such as the analysis, emergency and rescue alert system for automobiles after an accident. This study proposes the use of black boxes, technology from the aviation industry, which should be used in the automotive sector. These devices will serve to analyze when reviewing what causes were important factors in the collision. For this, the system will record images, which will allow researchers to know, among other things, how the vehicle was being driven and the road conditions. Another function, besides the analysis is that they also serve as assistance for a rapid emergency rescue operation, using the GPS, sending the location of the accident, sending text messages, and using cellular equipment, notifying the emergency service centers, such as hospitals, firemen and police. [9]

Others study the support system after the collision, also called Automatic Crash Notification, or ACN. This system uses the information from the sensors and communication technology, immediately calling for help, sending the exact location and magnitude of the impact to the rescue centers, with the characteristics of the accident, so rescue equipment can be sent for assistance. In 1995, some manufacturers began to equip the vehicles with basic ACN technology to transmit assistance signals in moderate to strong collisions, making use of the functionality of the airbag and sending exact coordinates from the GPS, establishing communication through the 911 emergency service. In more recent years, the detection of the address of the impact, presence of passengers, knowledge of three points of the use of the seatbelts, occurrence of a rollover and recognition of multiple collisions is used [10]. In recent years, some car models specifically include Post-Collision functionality.

Models of the 2016 Volkswagen, such as the Beetle, Jetta or Golf, have a Post-Collision braking system, intended to keep the occupants safe if the vehicle is involved in a collision. This driving assistance system may be of help avoiding potential collisions. When the airbag's impact sensor detects a collision, the vehicle slows down, applying the brakes. [11]

These studies are important since the occupants may have whiplash when there is a rear-end collision. The passengers' injuries occur due to several factors, including the severity of the impact, the design of the seatbelt, and the occupant's posture. To decrease the severity of these injuries, studies have been done that show the importance of the use of the seatbelt as well as the seat and headrest control, which keep the occupants safe. In the United States in 2014, approximately 24% of car accidents were rear-end collisions. [12].

III. DEVELOPMENT OF THE SOFTWARE REQUIREMENTS SPECIFICATION

In this section the proposal for behavior of a post-collision system whose objective is to minimize the time between a collision and the time in which the occupants of the vehicle receive external assistance is presented.

A. Document of Vision and Scope.

1) Business Requirements

a) Background

Currently, there are active safety systems, which are focused on preventing accidents before a collision, and passive safety systems, which act during the collision. However, there are few studies on safety focused on situations presented after a collision.

b) Business Opportunity.

To create proposals for automobile behavior in post-collision events and promote their inclusion in cars.

c) Business Objectives and Criteria of Success.

The criteria for success of the proposed system are:

- To minimize the time in which the occupants of a wrecked vehicle receive external assistance.
- To decrease the statistics of loss of human lives due to not having received prompt assistance after a collision.

d) Client or Market Needs

More proposals for behavior of post-collision safety systems that minimize the time in which passengers receive assistance after a serious accident are necessary. Also, these systems should be developed under the automotive standard ISO 26262 since they are critical mission systems.

e) Business risks.

- That the system is not of interest for the different assembly plants.
- That the cost of implementation increases the cost of vehicles substantially.

2) Vision of the product

That modern automobiles be equipped with systems that minimize the times in which passengers of the vehicle receive external assistance when they are not able to get out of the vehicle by themselves.

a) Important traits

- Detection of collision.
- Sensor reading (Impact, accelerometer, gyroscope, pressure, speed).
- Manipulation of Actuators (Inside lights, hazard lights, electrical locks, window regulators, seatbelts).
- Minimize assistance response time for vehicle occupants.

b) Assumptions and dependencies.

- It is assumed that the automobile has the necessary hardware installed to be able to carry out all features.
- It is assumed that the automobile has a CAN communication network for data transmission.
- It is assumed the system has electrical energy.

3) Scope and Limitations.

- The post-collision system will be focused on actions that allow a decrease in time in which passengers receive assistance after a collision.
- The post-collision system will function autonomously, without the intervention of the user.

a) Limitations and Exceptions

The post-collision system will not function if both the primary energy source and secondary energy source have been disabled due to the collision. The services the system will give cannot be carried out due to physical damage of the car's components that impede the actuators from working.

4) Business context

The post-collision system should be prepared for its implementation before the rapid change of needs in the automotive market.

a) Operation environment.

The post-collision system will be installed in automobiles and the user will not intervene in its functioning. The system will respond to the conditions detected by the sensors and will carry out their functions according to an analysis of their input.

B. Process of Definition of Stakeholder Requirements.

Among the activities and tasks of the process of definition of the Stakeholder's requirements and needs, the follow-up of activities and tasks should be implemented according to the policies and procedures of the organization through the following activities. [13].

- Be prepared for the definition of requirements and needs of the stakeholder.
- Define the needs of the stakeholder.

- Develop the operational concept and other concepts of the life cycle of the system development.
- Transform the stakeholder's needs into stakeholder's requirements.
- Analyze the stakeholder's requirements.

As described in the Theoretical Framework, the process of Definition of the Stakeholder's Requirements suggests recording the needs of the interested parties on various documents, among them, the Document of Concept of Operations (ConOps), which is presented in the following section.

1) ConOps Document.

a) Introduction

Purpose of the document of concept.

Currently, one of the main means of transportation is the automobile, with 1.2 billion cars in the entire world. The impressive amount of vehicles implies a high number of accidents. In Canada alone, every year between 2000 and 2010, there was an average of 2,632 people who died and 1.5 million who were injured [14]. Faced with this scenario, it is necessary that passengers involved in an accident receive external assistance quickly and without complications. For this reason, the post-collision system proposed seeks to minimize the time in which the occupants receive such assistance. For this, it will allow access to the inside of the car, activating some of the services, such as turning on the inside lights, turning on hazard lights, releasing electrical locks, automatically or temporarily lower windows and unbuckle seatbelts. To carry out this feature, the system will verify, a) that the vehicle is no longer in movement, and b) the final position of the car. This behavior contributes to a prompt assistance of the driver and/or passengers.

b) Concept of production

The system will be manufactured through electronic control card which will have embedded software. Additionally, they will be enabled to carry out communication through automotive communication protocol CAN (*Control Area Network*).

2) Concept of operations (ConOps)

The system reads the different sensors that determine the status of the vehicle after a collision. The control software will work the actuators according to the requirements established for each scenario.

The scenarios seen will determine the system's behavior and the following tasks should be carried out:

- Assess collision.

The post-collision system will detect a collision and its degree of severity.

- Detect the vehicle's state of rest.

After the impact, the system will identify the moment in which the car comes to a stop.

- Range of safe time

When the vehicle has stopped moving, a range of safe time will be determined, in which the vehicle has remained at rest. After this range of safe time, the system will begin the working of the different actuators.

- Position of the vehicle.

The behavior of the post-collision system depends on the final position of the vehicle, which is why this should be determined.

- Activate the secondary energy source.

The secondary energy source is only activated if the primary source (battery) is damaged or disconnected.

- Turn on interior lights.

If the collision happens at night, this feature helps the occupants see the inside of the car.

- De-activate door locks.

With this feature the door locks are released so they may be opened from outside.

- Roll down windows

There are scenarios that should be analyzed to determine if the windows can be rolled down or not.

- Identify the presence of occupants in the vehicle.

It is necessary to know in what seats there are occupants to know what windows should be rolled down.

- Identify position of the automobile

It is necessary to know in what physical position the car stopped. The possible positions are: a) All tires upward, b) All tires down, c) Rear tires upward, d) Front tires upward, e) Driver side tires upward, f) Passenger side tires upward.

- Identify window pressure

It will be identified if any passenger is leaning against the window. If so, the window will not roll down.

- Unbuckle seatbelts

The seatbelts should not automatically be released. To release them, the system should assess positions a) to f) of the feature "Identify position of the automobile". Depending on this assessment, it will be decided if all the seatbelts are released or only those that apply.

C. Requirement Analysis Process.

The purpose of the Requirement Analysis process of the standard ISO/IEC/IEEE 15288 is to transform the desired needs or services of the stakeholders in technical requirements of the system.

The activities of this process create a representation of the system, which should satisfy the functional requirements of the stakeholders and fulfill the non-functional restrictions or requirements. In this article, only the functional requirements are presented.

1) Requirement Modelling.

The specification of requirements was carried out through a combination of natural language (informal notation) y modeling languages (semi-formal notation). In this proposal, the SysML (*Systems Modeling Language*) modeling language is used. For the representation of the requirements of the proposed system, a semi-formal notation was used to comply with the recommendations of ISO 26262.

In Figure 3, the functional requirement can be seen: "The post-collision system should assess a collision event." To achieve this feature, the system will have impact sensors that will detect a front, rear, and side collision and its degree of severity.

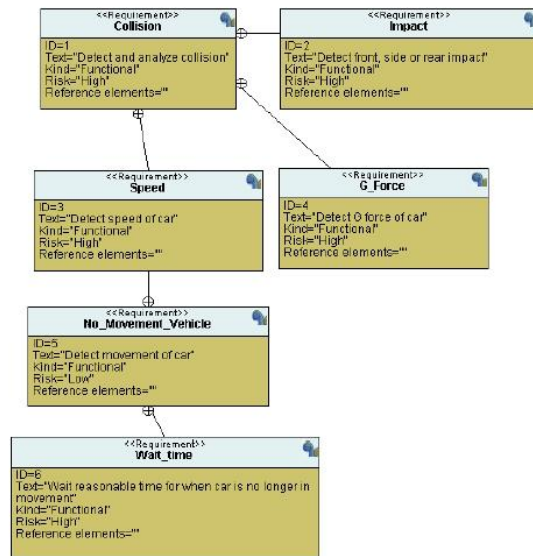


Fig. 3. Impact detection

In Figure 4, the functional requirement "The post-collision system should activate the secondary energy source" can be seen. To achieve this feature, the system will identify if there is no primary energy source. If the primary energy source is not present, the system will activate the secondary energy source.

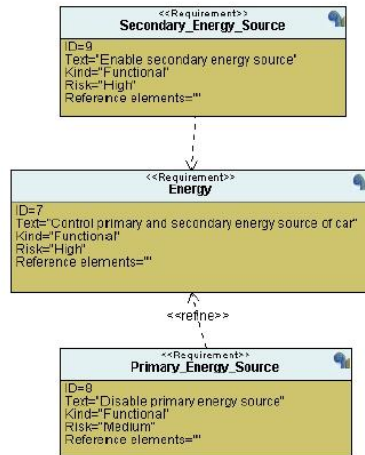


Fig. 4. Enabling secondary energy source

In Figure 5, the functional requirement “The post-collision system should work actuators depending on the status of the vehicle” can be seen. To achieve this functionality, the system will analyze the information from the sensors that determine the final physical position of the vehicle and the pressure on the windows. With this information, the system will carry out the activities of turning on lights, de-activating electrical locks, rolling down the windows and unbuckling seatbelts.

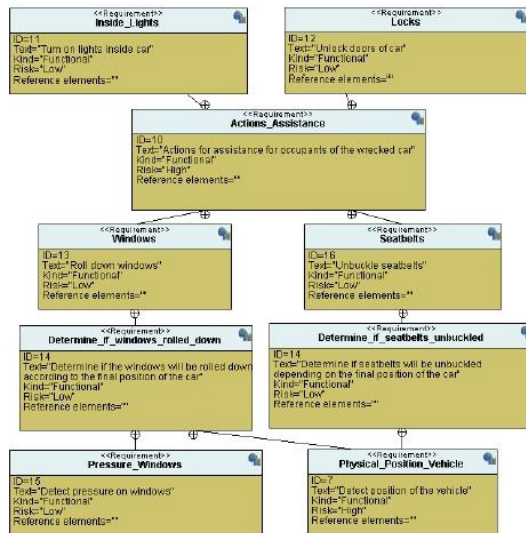


Fig. 5. Activation of actuators.

2) Use cases.

As additional support for the processes of Requirement Definition and Analysis, use case were developed that describe the requirements shown in the previous section. In that section, the details of the requirements are added through the basic flow

and the alternative flow of each use case shown. To save space, only one full use case is shown. In the use case 1, it assesses collision, when some of the impact sensors detect a signal, the speed sensor and the accelerometer sensor are immediately read to identify the speed at which the vehicle was traveling and to know the force of de-acceleration the vehicle had with the impact detected, respectively, to be able to carry out an analysis and determine if the collision was severe. (see Figure 6). For the other use case, only some details of the cases are given.

Use Case	CUI Assess collision
Those involved	Front impact sensor, right side impact sensor, left impact sensor, rear impact sensor, speed sensor, accelerometer sensor.
Description	Identify impact in those in which the speed is greater than 100 kilometers per hour and a de-acceleration force greater than or equal to 3g
Pictogram	
Type	Primary
Includes	Feature of the sensors
Basic flow	1.- The system reads front impact sensor input 2.- The system reads right side impact sensor input 3.- The system reads left side impact sensor input 4.- The system reads rear impact sensor input 5.- The system reads speed sensor 6.- The system reads the G force sensor 7.- The system processes sensor input 8. The system detects a speed > than 100 km per hour 9. The system detects a G force >= 3. 10. The system determines there has been a collision.
Alternative flow	1.1.- The system cannot read the front impact sensor 1.2.- The system will show a warning light on the dashboard. 2.1.- The system cannot read the right side impact sensor 2.2.- The system will show a warning light on the dashboard. 3.1.- The system cannot read the left side impact sensor 3.2.- The system will show a warning light on

	the dashboard.
	4.1.- The system cannot read the rear impact sensor
	4.2.- The system will show a warning light on the dashboard.
	5.1.- The system cannot read the speed sensor
	5.2.- The system will show a warning light on the dashboard.
	6.1.- The system cannot read the G force sensor
	6.2.- The system will show a warning light on the dashboard.

Fig. 6. CU1. Detection of impact

In Figure 7, the use case 8 is shown, “Unlock doors”, where the interaction with the different elements that allow access to the inside of the vehicle after a collision is shown.

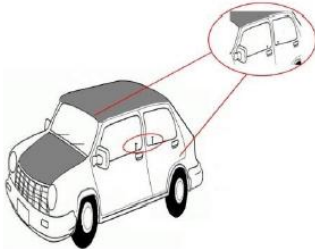
Use Case	CU8 Unlock doors
Description	Open all the locks on the vehicle's doors.
Pictogram	
Basic flow	1.- The system releases the driver's door lock. 2.- The system releases the front passenger door lock. 3.- The system releases the rear left passenger door lock. 4.- The system releases the rear right passenger door lock.
Alternative flow	1.1.- The system detects malfunction in opening the driver's door lock. 1.2.- The system will show a warning light on the dashboard. 2.1.- The system detects malfunction in opening the front passenger door lock. 2.2.- The system will show a warning light on the dashboard. 3.1.- The system detects malfunction in opening the rear left passenger door lock. 3.2.- The system will show a warning light on the dashboard. 4.1.- The system detects malfunction in opening the rear right passenger door lock. 4.2.- The system will show a warning light on the dashboard.

Fig.7. CU8. Unlock doors (partial).

In Figure 8, the basic and alternative flows of use case 9 “Roll down windows” are shown. It should previously be determined if there is pressure on the glass before carrying out that feature.

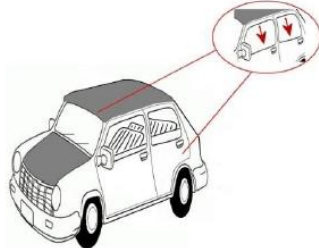
Use Case	CU9 Roll down windows
Description	Analyze the position in which the vehicle came to rest, combined with the pressure on the windows, to decide if the windows may be wound down.
Pictogram	
Basic flow	1.- The system reads the glass pressure sensor of the driver's window 2.- The system reads the glass pressure sensor of the front passenger window 3.- The system reads the glass pressure sensor of the left rear passenger window 4.- The system reads the glass pressure sensor of the right rear passenger window 5.- The system reads the information stored of the position of the automobile 6.- The system processes pressure sensor input, as well as the information on the position of the automobile 7.- The system rolls down the windows
Alternative flow	1.1.- The system detects malfunctioning of the sensor. 1.2.- The system will show a warning light on the dashboard. 2.1.- The system detects malfunctioning of the sensor. 2.2.- The system will show a warning light on the dashboard. 3.1.- The system detects malfunctioning of the sensor. 3.2.- The system will show a warning light on the dashboard. 4.1.- The system detects malfunctioning of the sensor. 4.2.- The system will show a warning light on the dashboard.

Fig. 8. CU9. Roll down windows (partial).

IV. CONCLUSIONS

With the specification of the system requirements, taking into account the standards ISO 26262 and ISO/IEC/IEEE 15288, this project attempts to create the quality requirements that allow a critical mission post-collision automotive system to be properly developed.

The passive and active safety systems, oriented toward collision events have been broadly studied; however, there is not much research nor many commercial implementations of post-collision safety systems.

The usefulness of researching and implementing post-collision systems in automobiles currently is due to the fact that they could reduce the percentage of deaths or injuries due to external assistance not being received promptly.

Development of this type of critical mission system should be done through a process-based approach. In our proposal, we base ourselves on the international standard ISO 26262 and the standard ISO/IEC/IEEE 15288 in order to guarantee their quality.

V. FUTURE TASK

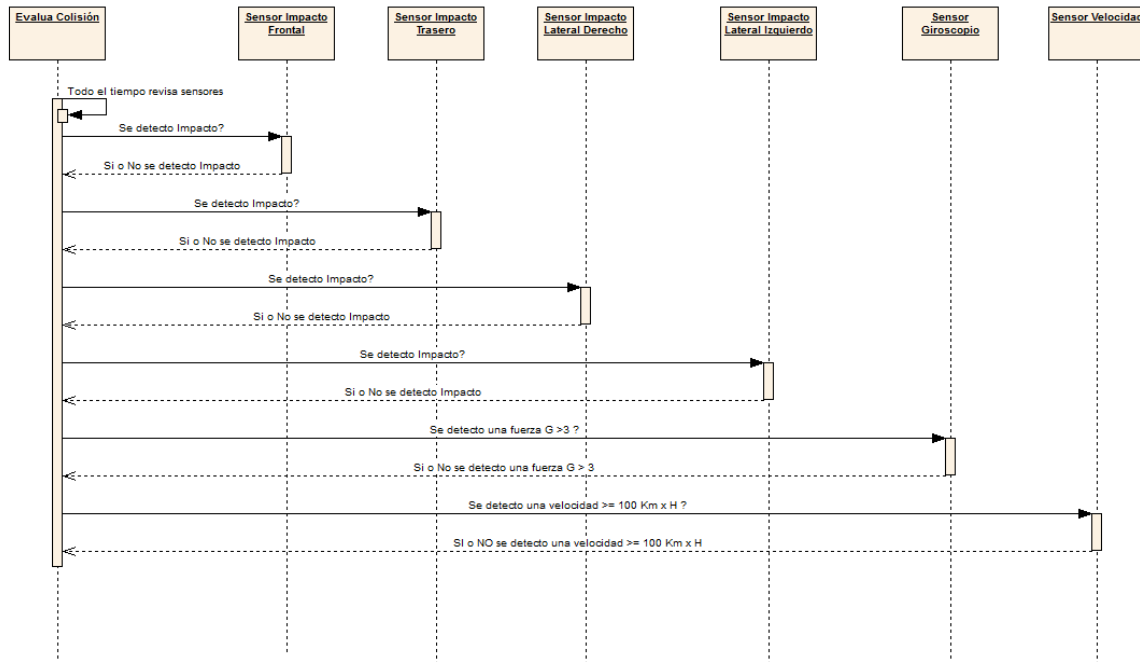
After defining and documenting the functional requirements of the post-collision system, the non-functional requirements will be developed. The design and architecture of the system will be the next step and will be done abiding by the recommendations of the international standard ISO 26262 and the international standard ISO/IEC/IEEE 15288. Finally, a functional prototype of the post-collision system will be developed which will be tested on a real car.

REFERENCES

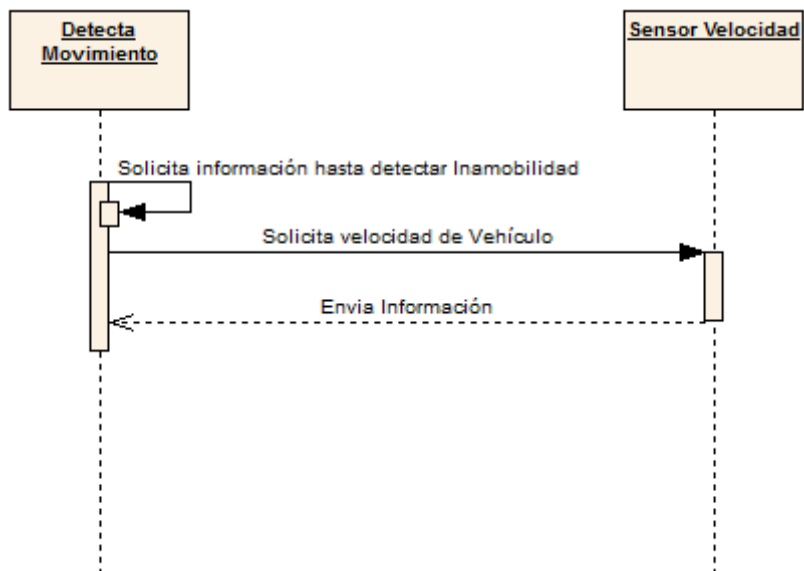
- [1] GmbH. V. V. (1999), *Manual de la técnica del Automóvil 3rd ed.*, Barcelona, España: Reverté.
- [2] Tenesca. M.G.E. Ramos. M.A.N. (2012). Implementación de un tablero de frenos ABS (Antiblock Brake System) para el laboratorio de la escuela de Ingeniería Automotriz. (Ingeniero Automotriz). Retrieved from <http://dspace.esPOCH.edu.ec/handle/123456789/2286#sthash.rNNzrT8.dpuf>
- [3] "ISO 26262, Functional safety for road vehicles", International Standard, November 2011.
- [4] Rivero. A.A.L. (2012), De sistema mecánico a sistema tecnológico completo, El sistema de los automóviles, Universidad Autónoma Metropolitana, Retrieved from <http://scielo.unam.mx/pdf/cya/v59n2/v59n2a2.pdf>
- [5] "ISO 26262, Functional safety for road vehicles - Scope", International Standard November 2011.
- [6] "ISO/IEC/IEEE 15288, System and Software engineering – System life cycle processes", International Standar Mayo 2015.
- [7] "ISO/IEC/IEEE 15288, System and Software engineering – System life cycle processes – Processes in this standard", International Standar Mayo 2015.
- [8] "ISO/IEC/IEEE 15288, System and Software engineering – System life cycle processes – 6.4.2 Stakeholder needs and requirements definition process", International Standar Mayo 2015.
- [9] Hassan. M. T. Z. Meguid. S. A. "Effect to seat belt and head restraint on occupant's response during rear-end collision", Springer Science+Business Media Dordrecht, April 2017.
- [10] Balaraju. S. Riyazuddin. K. "Car Post Crash Analysis & Emergency Rescue Alert System", International Journal of Advanced Research in Computer Science and Software Engineering, June 2016.
- [11] Augenstein. J. F. Bahouth. G. T. "Pos-crash Support Systems", Springer, 2012.
- [12] Findlay Volkswagen, <http://www.findlayvw.com/blog/what-does-vw-automatic-post-collision-braking-system-do/>, consulted on August 14, 2017.
- [13] "ISO/IEC/IEEE 15288, System and Software engineering – System life cycle processes – 6.4.2.3 Activities and tasks", International Standar Mayo 2015.
- [14] Canada. Transport Canada. Canadian Motor Vehicle Traffic Collision Statistics 2014 Collected in Cooperation with the Canadian Council of Motor Transport Administrators, 2014, Print.

Anexo E. DIAGRAMAS DE SECUENCIA

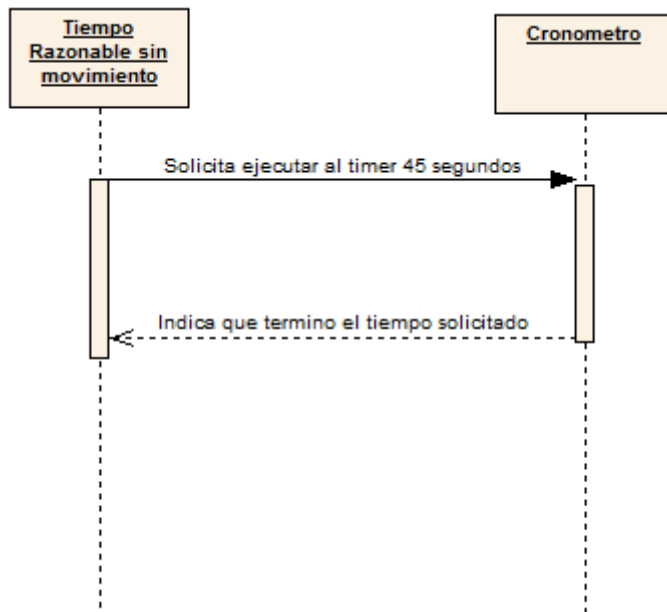
E.1. Diagrama de secuencia del caso de uso 1, “Evalúa Colisión”.



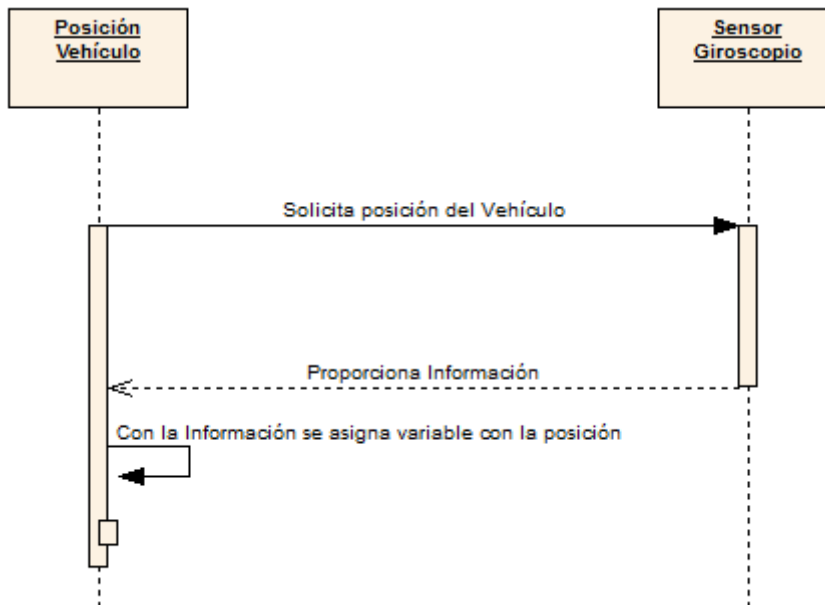
E.2. Diagrama de Secuencia del caso de uso 2, “Detecta Movimiento”.



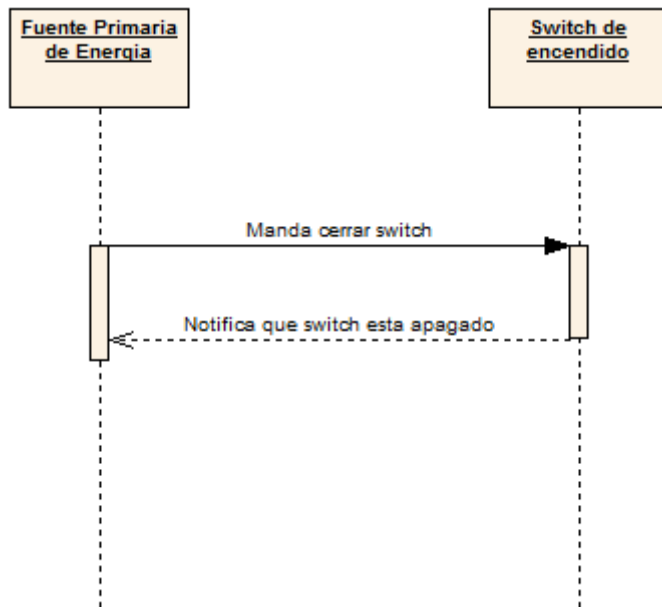
E.3. Diagrama de Secuencia de caso de uso 3, “Tiempo razonable sin movimiento”



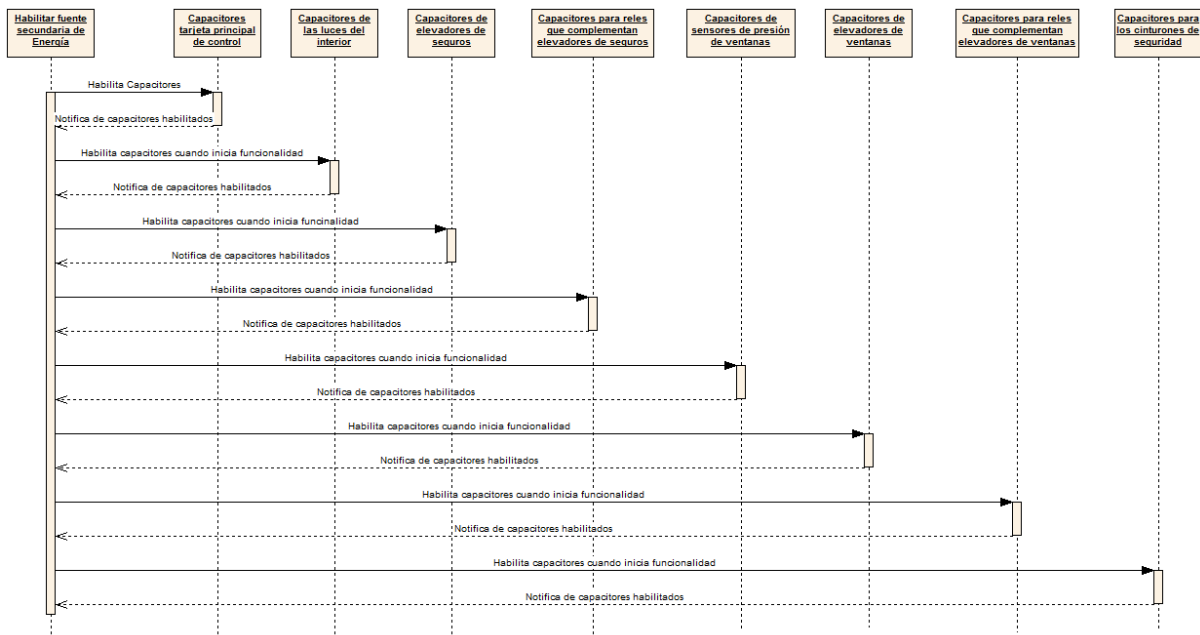
E.4. Diagrama de Secuencia de caso de uso 4: “Posición del Vehículo”



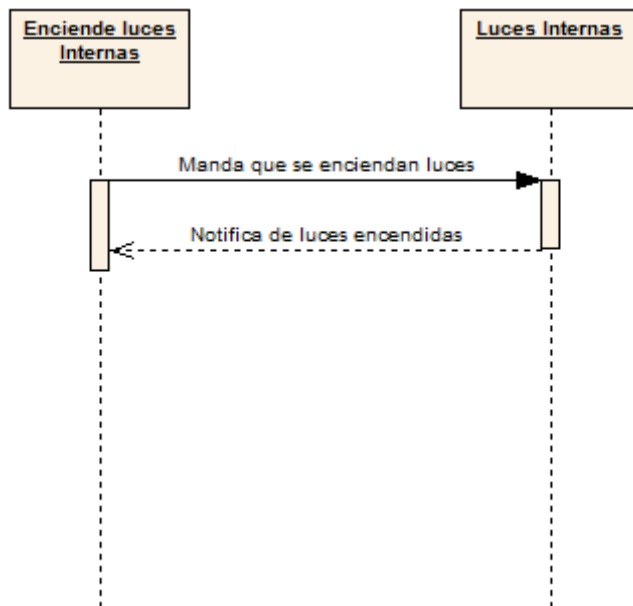
E.5. Diagrama de secuencia de Caso de Uso 5: “Deshabilita fuente primaria de energía”



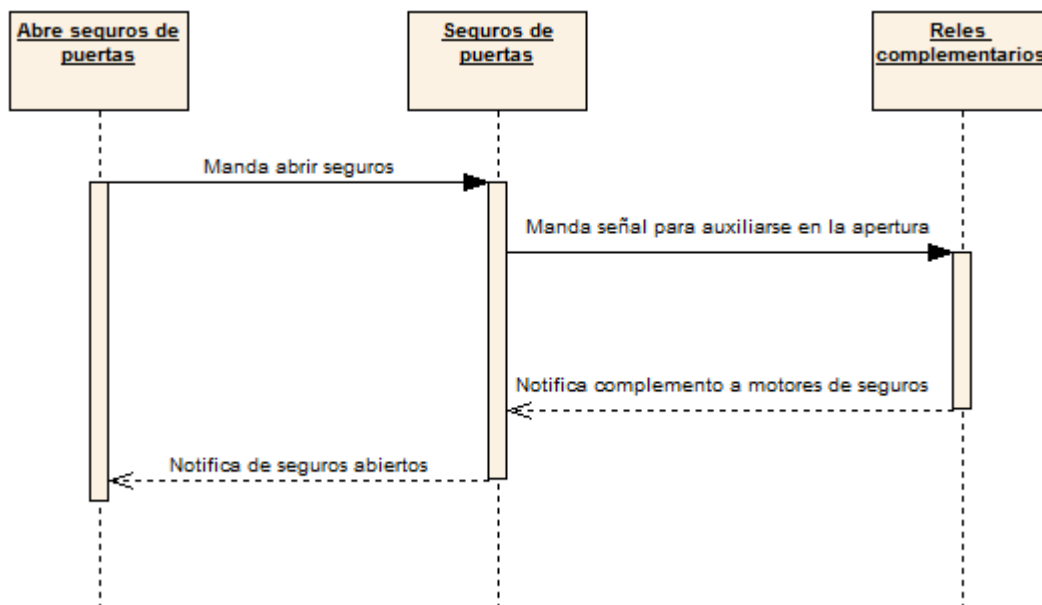
E.6. Diagrama de Secuencia de caso de uso 6: “Habilita fuente secundaria de energía”



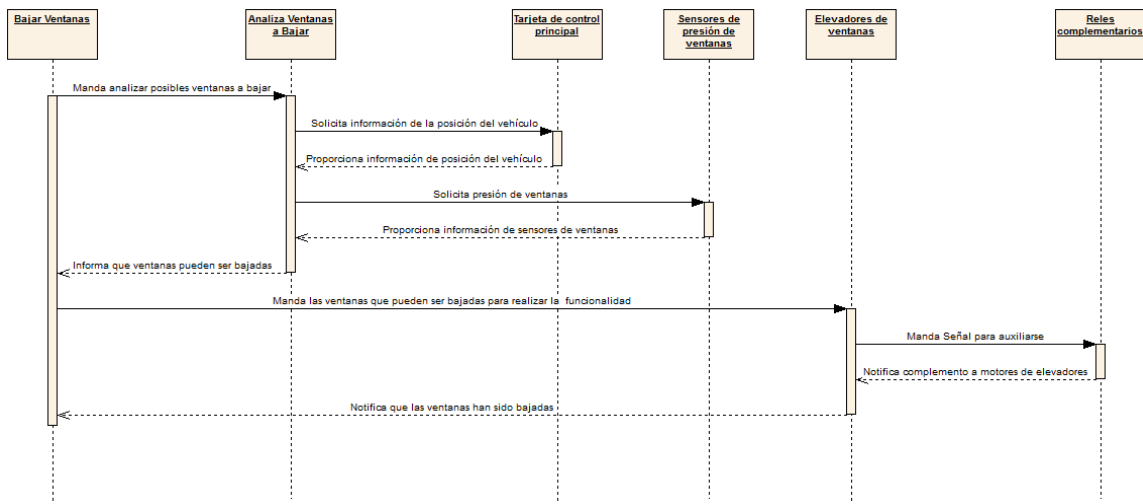
E.7. Diagrama de Secuencia de caso de uso 7: “Enciende luces Internas”



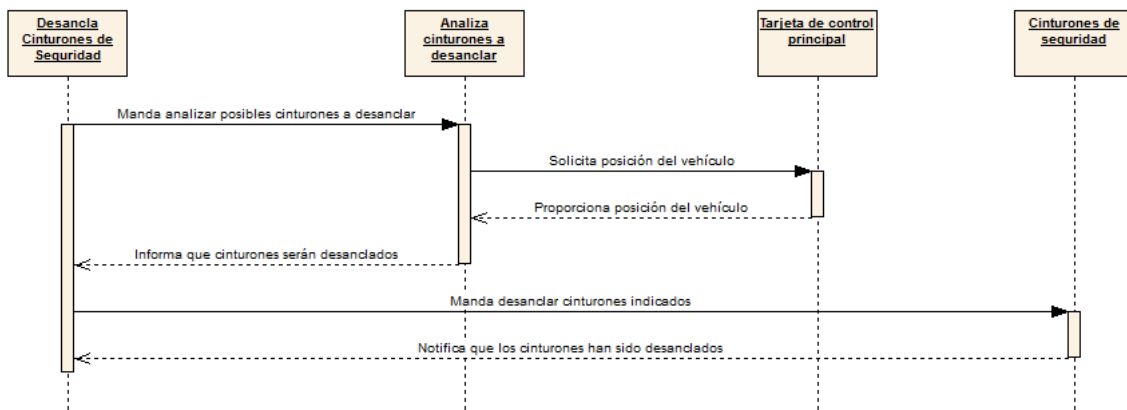
E.8. Diagrama de Secuencia de caso de uso 8: “Abre Seguro de Puertas”



E.9. Diagrama de secuencia de caso de uso 9: “Bajar Ventanas”



E.10. Diagrama de Secuencia de Caso de Uso 10: “Desancla cinturones de seguridad”



Anexo F. PSEUDOCÓDIGO

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	1
Número de Caso de Uso	1	Caso de Uso	Evalúa Colisión		
Inicio					
Procedimiento Evalúa_Colisión					
Comentario: Declara variables.					
Colision = verdadero tipo booleano					
Velocidad = falso tipo booleano					
Fuerza = falso tipo booleano					
SensorVelocidad = 0 tipo entero					
SensorFrontal = 0 tipo entero					
SensorLateralDerecho = 0 tipo entero					
SensorLateralIzquierdo = 0 tipo entero					
SensorFuerzaG = 0 tipo entero					
Mientras (Colision == falso) y (Velocidad == verdadero) y (Fuerza == verdadero) hacer					
Comentario: Verifica sensor de velocidad.					
SensorVelocidad = Lee sensor de velocidad, a qué velocidad va el vehículo					
Si SensorVelocidad >= 100 entonces					
Velocidad = Verdadero					
Si no					
Velocidad = Falso					
Fin Si					
Comentario: Verifica sensores de impacto					
Lee sensor de impacto frontal					
SensorFrontal = Lee sensor de impacto frontal, detecto colisión					
Si SensorFrontal == si entonces					
Colision = verdadero					
Si no					
SensorTrasero = Lee sensor de impacto trasero, detecto colisión					

Si SensorTrasero == si entonces

Colision = verdadero

Si no

SensorLateralDerecho = Lee sensor de impacto lateral derecho, detecto colisión

Si SensorLateralDerecho == si entonces

Colision = verdadero

Si no

Sensor LateralIzquierdo = Lee sensor de impacto lateral izquierdo, detecto

Colisión

Si SensorLateralIzquierdo == si entonces

Colision = verdadero

Fin Si

Fin Si

Fin Si

Fin Si

Comentario: Verifica sensor acelerómetro (Fuerza G)

SensorFuerzaG = Lee sensor acelerometro, que fuerza G se generó con el impacto

Si SensorFuerzaG >= 3 entonces

Fuerza = Verdadero

Si no

Fuerza = Falso

Fin Si

Fin Mientras

Comentario: Despliega resultados después de verificar sensores y cumplir con condiciones establecidas para un Impacto Severo (Detectar un impacto, velocidad >= 100 Km por hora y una Fuerza G > 3).

Escribir “Se detectó un impacto severo”

Escribir “Impacto Frontal - ” + SensorFrontal

Escribir “Impacto Trasero – “ + SensorTrasero

Escribir “Impacto Lateral Derecho – “ + SensorLateralDerecho
 Escribir “Impacto Lateral Izquierdo – “ + SensorLateralIzquierdo
 Escribir “Velocidad a la que viajaba el vehículo – “ + SensorVelocidad
 Escribir “Fuerza G – “ + SensorFuerzaG
 Fin Procedimiento
 Fin

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	2
Número de Caso de Uso		2	Caso de Uso	Detecta Movimiento	
Inicio Procedimiento Detecta_Movimiento Comentario: Declara Variables Velocidad = 1 tipo entero Mientras Velocidad <>0 hacer Velocidad == Lee sensor de velocidad, que velocidad tiene el vehículo Fin Mientras Comentario: Cuando el sensor de velocidad detecta inamovilidad (velocidad = 0) indica que el vehículo se detuvo. Se informa que el vehículo se ha detenido. Escribir “El vehículo se detuvo”. Fin Procedimiento Fin.					

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	3
Número de Caso de Uso		3	Caso de Uso	Tiempo razonable sin movimiento	
Inicio					
Procedimiento Tiempo_Razonable_Sin_Movimiento					
Comentario: Declara variables					
TiempoEspera = “yyyy/mm/dd 00:45” tipo date time					
Ejecuta Sleep(TiempoEspera segundos * 1000)					
Comentario: Informa que el tiempo razonable sin movimiento sin movimiento ha terminado					
Escribir “Tiempo de espera razonable de 45 segundos ha concluido”					
Fin Procedimiento					
Fin					

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	4
Número de Caso de Uso	4	Caso de Uso	Posición del vehículo		
Inicio Procedimiento Posición_Vehiculo Comentario: Declaración de variables PosicionXYZ = “00,00,00” tipo cadena ValorX = 0 tipo entero ValorY = 0 tipo entero ValorZ = 0 tipo entero Posicion = “ ” tipo cadena PosicionXYZ = Lee sensor giroscopio, que posición tiene el vehículo ValorX = Extrae de cadena antes de la primera coma (“,”) ValorY = Extrae de cadena después de la primera coma (“,”) y antes de la segunda coma (“;”)					

ValorZ = Extrae de cadena después de segunda coma (“,”)

Según PosicionXYZ hacer

Caso (ValorX>1 y ValorX<99) y (ValorY>1 y ValorY<99) y (ValorZ>1 y ValorZ<99)

Posicion = “Todas las llantas hacia arriba”

Caso (ValorX>2 y ValorX<-10) y (ValorY>2 y ValorY<-10) y (ValorZ>2 y ValorZ<-10)

Posicion = “Todas las llaves hacia abajo”

Caso (ValorX>3 y ValorX<160) y (ValorY>3 y ValorY<160) y (ValorZ>3 y ValorZ<160)

Posicion = “Llantas traseras hacia arriba”

Caso (ValorX>-4 y ValorX <-91) y (ValorY>-4 y ValorY<-91) y (ValorZ>-4 y ValorZ<-91)

Posicion = “Llantas delanteras hacia arriba”

Caso (ValorX>-7 y ValorX<88) y (ValorY>-7 y ValorY<88) y (ValorZ>-7 y ValorZ<88)

Posicion = “Llantas lado del piloto hacia arriba”

Caso (ValorX>56 y ValorX<6) y (ValorY>56 y ValorY<6) y (ValorZ>56 y ValorZ<6)

Posicion = “Llantas lado del copiloto hacia arriba”

Fin Según

Comentario: Informar la posición en la que se encuentra el vehículo.

Escribir Posicion

Fin Procedimiento

Fin

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	5
Número de Caso de Uso	5	Caso de Uso	Deshabilita fuente primaria de		

			energía
Inicio Procedimiento Deshabilita_FuenteDeEnergia Switch = Lee Switch principal Switch. Cerrar_Switch Fin Procedimiento Fin			

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	6
Número de Caso de Uso	6	Caso de Uso	Habilita fuente secundaria de energía		
Inicio					
Procedimiento Habilita_FuenteSecundaria (FuncionalidadActual tipo cadena)					
Comentario: Declara variables					
CapacitorTarjPrinc = Condensador tipo objeto					
CapacitorLuzInterna = Condensador tipo objeto					
CapacitorSeguroPP = Condensador tipo objeto					
CapacitorSeguroPC = Condensador tipo objeto					
CapacitorSeguroPTP = Condensador tipo objeto					
CapacitorSeguroPTC = Condensador tipo objeto					
CapacitorReleSegPP = Condensador tipo objeto					
CapacitorReleSegPT = Condensador tipo objeto					
CapacitorReleSegPTP = Condensador tipo objeto					
CapacitorReleSegPTC = Condensador tipo objeto					
CapacitorSensorPresionPP = Condensador tipo objeto					
CapacitorSensorPresiónPC = Condensador tipo objeto					

CapacitorSensorPresiónPTP = Condensador tipo objeto
CapacitorSensorPresiónPTC = Condensador tipo objeto
CapacitorElevadorPP = Condensador tipo objeto
CapacitorElevadorPT = Condensador tipo objeto
CapacitorElevadorPTP = Condensador tipo objeto
CapacitorElevadorPTC = Condensador tipo objeto
CapacitorReleElevadorPP = Condensador tipo objeto
CapacitorReleElevadorPT = Condensador tipo objeto
CapacitorReleElevadorPTP = Condensador tipo objeto
CapacitorReleElevadorPTC = Condensador tipo objeto
CapacitorCinturonPP = Condensador tipo objeto
CapacitorCinturonPT = Condensador tipo objeto
CapacitorCinturonPTP = Condensador tipo objeto
CapacitorCinturonPTC = Condensador tipo objeto

CapacitorTarjPrin.EnciendeCapacitor

Escribir “Capacitores de Tarjeta Principal Encendidos”

Según FuncionalidadActual hacer

Caso “Enciende Luces Internas”

CapacitorLuzInterna.EnciendeCapacitor

Escribir “Capacitor de Luz Interna Encendido”

Caso “Baja Seguros”

CapacitorSeguroPP.EnciendeCapacitor

CapacitorSeguroPC.EnciendeCapacitor

CapacitorSeguroPTP.EnciendeCapacitor

CapacitorSeguroPTC.EnciendeCapacitor

CapacitorReleSegPP.EnciendeCapacitor

CapacitorReleSegPT.EnciendeCapacitor

CapacitorReleSegPTP.EnciendeCapacitor

CapacitorReleSegPTC.EnciendeCapacitor

Escribir “Capacitores de Funcionalidad de bajar Seguros Eléctricos Encendidos”

Caso “Bajar Ventanas”

CapacitorSensorPresionPP.EnciendeCapacitor

CapacitorSensorPresiónPC.EnciendeCapacitor

CapacitorSensorPresiónPTP.EnciendeCapacitor

CapacitorSensorPresiónPTC.EnciendeCapacitor

CapacitorElevadorPP.EnciendeCapacitor

CapacitorElevadorPT.EnciendeCapacitor

CapacitorElevadorPTP.EnciendeCapacitor

CapacitorElevadorPTC.EnciendeCapacitor

CapacitorReleElevadorPP.EnciendeCapacitor

CapacitorReleElevadorPT.EnciendeCapacitor

CapacitorReleElevadorPTP.EnciendeCapacitor

CapacitorReleElevadorPTC.EnciendeCapacitor

Escribir “Capacitores de Funcionalidad de Bajar Ventanas Encendidos”

Caso “Desancla Cinturones”

CapacitorCinturonPP.EnciendeCapacitor

CapacitorCinturonPT.EnciendeCapacitor

CapacitorCinturonPTP.EnciendeCapacitor

CapacitorCinturonPTC.EnciendeCapacitor

Escribir “Capacitores de Funcionalidad de Desanclar Cinturones de Seguridad Encendidos”

Fin Según

Fin de Procedimiento

Fin

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	7
Número de Caso de Uso	7	Caso de Uso	Luces internas		
Inicio Procedimiento Enciende_Luces Comentario: Declara variables LuzInterna = “ ” tipo Objeto Comentario: Ejecta procedimiento para encender la Fuente secundaria de energía antes de iniciar con la funcionalidad. Habilita_FuenteSecundaria(“Enciende Luces”) LuzInterna = Lee tarjeta que controla la luz interna del vehículo LuzInterna.EnciendeLuz Fin Procedimiento Fin					

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	8
Número de Caso de Uso		8	Caso de Uso	Abre seguros de puertas	
Inicio Procedimiento AbreSegurosPuertas Comentario: Declara variables Rele = “ ” tipo Objeto Seguro = “ ” tipo Objeto Comentario: Ejecta procedimiento para encender la Fuente secundaria de energía antes de iniciar con la funcionalidad. Habilita_FuenteSecundaria(“Abre seguros de Puertas”) Comentario: “SPP” = Seguro de Puerta Principal, “SPC” = Seguro de Puerta Copiloto, “SPTP” = Seguro de Puerta Trasera Piloto, “SPTC” = Seguro de Puerta Trasera Copiloto” Para cada x en {“SPP”, ”SPC”, ”SPTP”, ”SPTC”} hacer					

Rele = Buscar Rele de x Seguro = Buscar Seguro de puerta de x Rele.Activar Seguro.BajarSeguro Fin para cada Escribir “Seguros de Puertas están Abajo” Fin Procedimiento Fin

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	9
Número de Caso de Uso		9	Caso de Uso	Bajar Ventanas	
Inicio					
Procedimiento BajarVentanas					
Comentario: Declara variables					
PosicionVehiculo = “ ” tipo cadena					
PresionVP = falso tipo booleano					
PresionVC = falso tipo booleano					
PresionVTP = falso tipo booleano					
PresionVTC = falso tipo booleano					
BajarVP = falso tipo booleano, Comentario: Ventana Piloto					
BajarVC = falso tipo booleano, Comentario: Ventana Copiloto					
BajarVTP = falso tipo booleano, Comentario: Ventana Trasera Piloto					
BajarVTC = falso tipo booleano, Comentario: Ventana Trasera Copiloto					
Rele = “ ” tipo Objeto					
Elevador = “ ” tipo Objeto					
Comentario: Ejecuta procedimiento para encender la Fuente secundaria de energía antes de					

iniciar con la funcionalidad.

Habilita_FuenteSecundaria(“Bajar Ventanas”)

Comentario: Se lee la tarjeta de control principal para buscar la variable de posición del vehículo, previamente detectada en el caso de uso 8.

PosicionVehiculo = Lee tarjeta de control principal, posición de vehículo.

PresionVP = Lee sensor de presión ventana piloto, se detectó presión.

PresionVC = Lee sensor de presión ventana copiloto, se detectó presión.

PresionVTP = Lee sensor de presión ventana trasera piloto, se detectó presión.

PresionVTC = Lee sensor de presión ventana trasera copiloto, se detectó presión.

ListaEvaluaPresion = [4] tipo booleano

Según PosicionVehiculo hacer

Caso “Todas las llantas hacia arriba”

BajarVP = verdadero

BajarVC = verdadero

BajarVTP = verdadero

BajarVTC = verdadero

Caso “Todas las llantas hacia abajo”

ListaEvaluaPresion = RevisaPresion(PresionVP, PresionVC, PresionVTP, PresionVTC)

BajarVP = ListaEvaluaPresion[1]

BajarVC = ListaEvaluaPresion[2]

BajarVTP = ListaEvaluaPresion[3]

BajarVTC = ListaEvaluaPresion[4]

Caso “Llantas traseras hacia arriba”

ListaEvaluaPresion = RevisaPresion(PresionVP, PresionVC, PresionVTP, PresionVTC)

BajarVP = ListaEvaluaPresion[1]

BajarVC = ListaEvaluaPresion[2]

BajarVTP = ListaEvaluaPresion[3]

BajarVTC = ListaEvaluaPresion[4]

Caso “Llantas delanteras hacia arriba”

ListaEvaluaPresion = RevisaPresion(PresionVP, PresionVC, PresionVTP, PresionVTC)

BajarVP = ListaEvaluaPresion[1]

BajarVC = ListaEvaluaPresion[2]

BajarVTP = ListaEvaluaPresion[3]

BajarVTC = ListaEvaluaPresion[4]

Caso “Llantas lado del piloto hacia arriba”

ListaEvaluaPresion = RevisaPresion(PresionVP, PresionVC, PresionVTP, PresionVTC)

BajarVP = ListaEvaluaPresion[1]

BajarVC = ListaEvaluaPresion[2]

BajarVTP = ListaEvaluaPresion[3]

BajarVTC = ListaEvaluaPresion[4]

Caso “Llantas lado del copiloto hacia arriba”

ListaEvaluaPresion = RevisaPresion(PresionVP, PresionVC, PresionVTP, PresionVTC)

BajarVP = ListaEvaluaPresion[1]

BajarVC = ListaEvaluaPresion[2]

BajarVTP = ListaEvaluaPresion[3]

BajarVTC = ListaEvaluaPresion[4]

Fin Según

Si BajarVP = verdadero entonces

Rele = Buscar Rele de Ventana Piloto

Elevador = Buscar Elevador de Ventana Piloto

Rele.Activar

Elevador.BajarVentana

Si no

Si BajarVC = verdadero entonces

Rele = Buscar Rele de Ventana Copiloto

Elevador = Buscar Elevador de Ventana Copiloto

Rele.Activar

Elevador.BajarVentana

Si no

Si BajarVTP = verdadero entonces

Rele = Busca Rele de ventana tracera piloto

Elevador = Busca Elevador de ventana tracera piloto

Rele.Activar

Elevador.BajarVentana

Si no

Si BajarVTC = verdadero entonces

Rele = Buscar Rele de ventana tracera copiloto

Elevador = Busca Elevador de ventana tracera copiloto

Rele.Activar

Elevador.BajarVentana

Fin si

Fin si

Fin si

Fin si

Escribir “Las Ventanas han sido bajadas”

Fin Procedimiento

Comentario: Procedimiento auxiliar que evalua la presión de las ventanas

Procedimiento RevisaPresion(PresionVP, PresionVC, PresionVTP, PresionVTC) tipo Lista

Comentario: Declara Variables

BVP = falso tipo booleano

BVC = falso tipo booleano

```
BVTP = falso tipo booleano
BVTC = falso tipo booleano
Si PresionVP = verdadero entonces
    BVP = falso
Si no
    BVP = verdadero
Fin si
Si PresionVC = verdadero entonces
    BVC = falso
Si no
    BVC = verdadero
Fin si
Si PresionVTP = verdadero entonces
    BVTP = falso
Si no
    BVTP = verdadero
Fin si
Si PresionVTC = verdadero entonces
    BVTC = falso
Si no
    BVTC = verdadero
Fin si
Lista[1] = BVP
Lista[2] = BVC
Lista[3] = BVTP
Lista[4] = BVTC
Regresa Lista
Fin Procedimiento
Fin
```

Nombre	GRG	Fecha	29/08/2017	Número de Pseudocódigo	10
Número de Caso de Uso			10	Caso de Uso	Desancla cinturones de seguridad
Inicio					
Procedimiento DesanclaCinturones					
Comentario: Declara variables					
PosicionVehiculo = “ ” tipo cadena					
DesanclaCP = falso tipo booleano, Comentario: Cinturon Piloto					
DesanclaCC = falso tipo booleano, Comentario: Cinturon Copiloto					
DesanclaCTP = falso tipo booleano, Comentario: Cinturon Trasero Piloto					
DesanclaCTC = falso tipo booleano, Comentario: Cinturon Trasero Copiloto					
Rele = “ ” tipo Objeto					
Cinturon = “ ” tipo Objeto					
Comentario: Ejecuta procedimiento para encender la Fuente secundaria de energía antes de iniciar con la funcionalidad.					
Habilita_FuenteSecundaria(“Desancla cinturones de seguridad”)					
Comentario: Se lee la tarjeta de control principal para buscar la variable de posición del vehículo, previamente detectada en el caso de uso 8.					
PosicionVehiculo = Lee tarjeta de control principal, posición de vehículo.					
Según PosicionVehiculo hacer					
Caso “Todas las llantas hacia arriba”					
Escribir “No se desancla ningún cinturón de seguridad”					
Caso “Todas las llantas hacia abajo”					
DesanclaCP = verdadero					
DesanclaCC = verdadero					
DesanclaCTP = verdadero					
DesanclaCTC = verdadero					
Caso “Llantas traseras hacia arriba”					
DesanclaCP = falso					
DesanclaCC = falso					

DesanclaCTP = falso

DesanclaCTC = falso

Caso “Llantas delanteras hacia arriba”

DesanclaCP = falso

DesanclaCC = falso

DesanclaCTP = verdadero

DesanclaCTC = verdadero

Caso “Llantas lado del piloto hacia arriba”

DesanclaCP = falso

DesanclaCC = verdadero

DesanclaCTP = falso

DesanclaCTC = verdadero

Caso “Llantas lado del copiloto hacia arriba”

DesanclaCP = verdadero

DesanclaCC = falso

DesanclaCTP = verdadero

DesanclaCTC = falso

Fin Según

Si DesanclaCP = verdadero entonces

Cinturon = Buscar actuador cinturón piloto

Cinturon.DesanclarCinturon

Si no

Si DesanclaCC = verdadero entonces

Cinturon = Buscar actuador de cinturón copiloto

Cinturon.DesanclarCinturon

Si no

Si DesanclaCTP = verdadero entonces

Cinturon = Busca ancutador de cinturón trasero piloto

Cinturon.DesanclarCinturon

Si no

Si DesanclaCTC = verdadero entonces

Cinturon = Busca actuador de cinturón trasero copiloto

Cinturon.DesanclarCinturon

Fin si

Fin si

Fin si

Fin si

Escribir “Los cinturones de seguridad han sido desanclados”

Fin Procedimiento

Fin

Anexo G. CÓDIGO

Nombre	GRG	Fecha	29/08/2017	Programa	PosicionVehiculo.ino
Número de Caso de Uso	4	Caso de Uso	Posición del Vehículo		

```
#include <EEPROM.h>

int ve;

int addr = 0;

void setup() {

.....

}

void loop() {

.....

# Código reusado

#ifndef OUTPUT_READABLE_YAWPITCHROLL

// display Euler angles in degrees

mpu.dmpGetQuaternion(&q, fifoBuffer);

mpu.dmpGetGravity(&gravity, &q);

mpu.dmpGetYawPitchRoll(ypr, &q, &gravity);

y = ypr[0] * 180 / M_PI;

p = ypr[1] * 180 / M_PI;

r = ypr[2] * 180 / M_PI;

if ((p >= 6 && p <= 20) && (r >= -7 && r <= 7)) {

ve = 1;

EEPROM.write(addr, ve);

Serial.print("(1) Todas las Llantas Arriba (Y: ");

// Serial.println(F(" ..... Valores: ..... y : "));

Serial.print(y);

Serial.print(" P: ");

Serial.print(p);
```

```

    Serial.print(" R: ");
    Serial.print(r);
    Serial.println("");
}
else if ((p >= -10 && p <= 4) && (r >= -8 && r <= 6)) {
    ve = 2;
    EEPROM.write(addr, ve);
    Serial.print("(2) Todas las Lantas Abajo (Y: ");
    Serial.print(y);
    Serial.print(" P: ");
    Serial.print(p);
    Serial.print(" R: ");
    Serial.print(r);
    Serial.println("");
}
else if ((p >= 61 && p <= 75) && (r >= -6 && r <= 8)) {
    ve = 3;
    EEPROM.write(addr, ve);
    Serial.print("(3) Lantas Traseras Arriba (Y: ");
    Serial.print(y);
    Serial.print(" P: ");
    Serial.print(p);
    Serial.print(" R: ");
    Serial.print(r);
    Serial.println("");
}
else if ((p >= -88 && p <= -74) && (r >= -7 && r <= 7)) {
    ve = 4;
    EEPROM.write(addr, ve);

```

```

Serial.print("(4) Llantas Delanteras Arriba (Y: ");
Serial.print(y);
Serial.print(" P: ");
Serial.print(p);
Serial.print(" R: ");
Serial.print(r);
Serial.println(")");
}
else if (( p >= -5 && p <= 9) && (r >= -82 && r <= -68)) {
    ve = 5;
    EEPROM.write(addr, ve);
    Serial.print("(5) Llantas L Piloto Arriba (Y: ");
    Serial.print(y);
    Serial.print(" P: ");
    Serial.print(p);
    Serial.print(" R: ");
    Serial.print(r);
    Serial.println(")");
}
else if (( p >= -4 && p <= 10) && (r >= 68 && r <= 82)) {
    ve = 6;
    EEPROM.write(addr, ve);
    Serial.print("(6) Llantas L Copiloto Arriba (Y: ");
    Serial.print(y);
    Serial.print(" P: ");
    Serial.print(p);
    Serial.print(" R: ");
    Serial.print(r);
    Serial.println(")");
}

```

```

}

else {
    Serial.println("");
};

#endif
# Continúa Código reusado

.....
}

```

Nombre	GRG	Fecha	29/08/2017	Programa	BajarVentanas.ino
Número de Caso de Uso		9	Caso de Uso	Bajar Ventanas	
<pre>//Acceso libreria para motor #include <Servo.h> #include <EEPROM.h> //Se declaran variables Servo servoMotor; int fsrPin = 0; int fsrReading; int addr = 0; int ve; void setup() { Serial.begin(9600); servoMotor.attach(9);</pre>					

```

ve = EEPROM.read(addr);
Serial.print("Valor de Variable EEPROM : ");
Serial.print(ve);
Serial.print(" ");

Serial.println("Inicia Prueba.....");
switch(ve){
case 1:
    //Escenario 1 (posición del vehiculo: TODAS LAS LLANTAS ARRIBA)
    #####
    Serial.println("--- Todas las Llantas Arriba --- ");
    Serial.println("--- La prueba inicia con ventanas arriba --- ");
    Serial.println(" --- VENTANA PILOTO --- ");
    AuxUno();
    break;

case 2:
    //Escenario 2 (posición del vehiculo: TODAS LAS LLANTAS ABAJO)
    #####
    Serial.println("--- Todas las Llantas Abajo --- ");
    Serial.println("--- La prueba inicia con ventanas arriba --- ");
    Serial.println(" --- VENTANA PILOTO --- ");
    AuxDos();
    break;

case 3:
    //Escenario 3 (posición del vehiculo: LLANTAS TRASERAS HACIA ARRIBA)
    #####

```

```

Serial.println("--- Llantas Traseras hacia arriba --- ");
Serial.println("--- La prueba inicia con ventanas arriba --- ");
Serial.println(" --- VENTANA PILOTO --- ");
AuxUno();
break;

case 4:
    //Escenario 4 (posición del vehiculo: LLANTAS DELANTERAS HACIA ARRIBA)
    #####
    Serial.println("--- Llantas Delanteras hacia arriba --- ");
    Serial.println("--- La prueba inicia con ventanas arriba --- ");
    Serial.println(" --- VENTANA PILOTO --- ");
    AuxDos();
    break;

case 5:
    //Escenario 5 (posición del vehiculo: LLANTAS LADO DEL PILOTO HACIA ARRIBA)
    #####
    Serial.println("--- Llantas Lado del Piloto hacia arriba --- ");
    Serial.println("--- La prueba inicia con ventanas arriba --- ");
    Serial.println(" --- VENTANA PILOTO --- ");
    //Motor Ventana Arriba
    servoMotor.write(180);
    delay(3000);
    Serial.println(" Ventana Piloto Abajo ...");
    servoMotor.write(0);
    delay(3000);
    //Se inicializa ventana arriba
    servoMotor.write(180);

```

```
Serial.println(" --- VENTANA TRASERA PILOTO --- ");
```

```
delay(3000);
```

```
Serial.println(" Ventana Trasera Piloto Abajo ...");
```

```
servoMotor.write(0);
```

```
delay(3000);
```

```
//Se inicializa ventana arriba
```

```
servoMotor.write(180);
```

```
Serial.println(" --- VENTANA COPILOTO --- ");
```

```
delay(3000);
```

```
Serial.println(" Ventana Copiloto Arriba ...");
```

```
servoMotor.write(180);
```

```
delay(3000);
```

```
//Se inicializa ventana arriba
```

```
servoMotor.write(180);
```

```
Serial.println(" --- VENTANA TRASERA COPILOTO --- ");
```

```
delay(3000);
```

```
Serial.println(" Ventana Trasera copiloto Arriba ...");
```

```
servoMotor.write(180);
```

```
delay(3000);
```

```
//Se inicializa ventana arriba
```

```
servoMotor.write(180);
```

```
Serial.println(" ");
```

```
break;
```

```
case 6:
```

```
//Escenario 6 (posición del vehiculo: LLANTAS LADO DEL COPILOTO HACIA
```

ARRIBA)

#####

```
Serial.println("--- Llantas Lado del Copiloto hacia arriba --- ");
```

```
Serial.println("--- La prueba inicia con ventanas arriba --- ");
```

```
Serial.println(" --- VENTANA PILOTO --- ");
```

```
//Motor Ventana Arriba
```

```
servoMotor.write(180);
```

```
delay(3000);
```

```
Serial.println(" Ventana Piloto Arriba ...");
```

```
servoMotor.write(180);
```

```
delay(3000);
```

```
//Se inicializa ventana arriba
```

```
servoMotor.write(180);
```

```
Serial.println(" --- VENTANA TRASERA PILOTO --- ");
```

```
delay(3000);
```

```
Serial.println(" Ventana Trasera Piloto Arriba ...");
```

```
servoMotor.write(180);
```

```
delay(3000);
```

```
//Se inicializa ventana arriba
```

```
servoMotor.write(180);
```

```
Serial.println(" --- VENTANA COPILOTO --- ");
```

```
delay(3000);
```

```
Serial.println(" Ventana Copiloto Abajo ...");
```

```
servoMotor.write(0);
```

```
delay(3000);
```

```
//Se inicializa ventana arriba
```

```
servoMotor.write(180);
```

```

Serial.println(" --- VENTANA TRASERA COPILOTO --- ");
delay(3000);
Serial.println(" Ventana Trasera copiloto Abajo ...");
servoMotor.write(0);
delay(3000);
//Se inicializa ventana arriba
servoMotor.write(180);
Serial.println(" ");
break;
}

Serial.println(" ");
Serial.println(" Fin de Prueba .... ");
}

void AuxUno(){
//Rutina para Escenario 1 (posición del vehiculo: TODAS LAS LLANTAS ARRIBA) y
Escenario 3 (posición del vehiculo: LLANTAS TRASERAS HACIA ARRIBA)
#####
//Motor Ventana Arriba
servoMotor.write(180);
delay(3000);
Serial.println(" Ventana Piloto Abajo ...");
servoMotor.write(0);
delay(3000);
//Se inicializa ventana arriba
servoMotor.write(180);

```

```

Serial.println(" --- VENTANA TRASERA PILOTO --- ");
delay(3000);
Serial.println(" Ventana Trasera Piloto Abajo ...");
servoMotor.write(0);
delay(3000);
//Se inicializa ventana arriba
servoMotor.write(180);

Serial.println(" --- VENTANA COPILOTO --- ");
delay(3000);
Serial.println(" Ventana Copiloto Abajo ...");
servoMotor.write(0);
delay(3000);
//Se inicializa ventana arriba
servoMotor.write(180);

Serial.println(" --- VENTANA TRASERA COPILOTO --- ");
delay(3000);
Serial.println(" Ventana Trasera copiloto Abajo ...");
servoMotor.write(0);
delay(3000);
//Se inicializa ventana arriba
servoMotor.write(180);
Serial.println(" ");
}

void AuxDos(){
//Rutina para Escenario 2 (posición del vehiculo: TODAS LAS LLANTAS ABAJO) y
4(posición del vehiculo: LLANTAS DELANTERAS HACIA ARRIBA)

```

```
#####

//Motor Ventana Arriba
servoMotor.write(180);
delay(3000);
//Lee sensor de presión
fsrReading = analogRead(fsrPin);
Serial.print("Lectura Sensor Presion : ");
Serial.print(fsrReading);
// && fsrReading != 1023
if (fsrReading > 0 ) {
  Serial.println(" Se detectó presión, No bajara Ventana");
}
else{
  Serial.println(" Ventana Piloto Abajo ...");
  servoMotor.write(0);
};
delay(3000);
//Se inicializa ventana arriba
servoMotor.write(180);

Serial.println(" --- VENTANA TRASERA PILOTO --- ");
delay(3000);
//Lee sensor de presión
fsrReading = analogRead(fsrPin);
Serial.print("Lectura Sensor Presion : ");
Serial.print(fsrReading);
//&& fsrReading != 1023
if (fsrReading > 0 ) {
  Serial.println(" Se detectó presión, No bajara Ventana");
}
```

```

}
else{
  Serial.println(" Ventana trasera Piloto Abajo ...");
  servoMotor.write(0);
};
delay(3000);
//Se inicializa ventana
servoMotor.write(180);

Serial.println(" --- VENTANA COPILOTO --- ");
delay(3000);
//Lee sensor de presión
fsrReading = analogRead(fsrPin);
Serial.print("Lectura Sensor Presion : ");
Serial.print(fsrReading);
//&& fsrReading != 1023
if (fsrReading > 0 ) {
  Serial.println(" Se detectó presión, No bajara Ventana");
}
else{
  Serial.println(" Ventana Copiloto Abajo ...");
  servoMotor.write(0);
};
delay(3000);
//Se inicializa ventana
servoMotor.write(180);

Serial.println(" --- VENTANA TRASERA COPILOTO --- ");
delay(3000);

```

```

//Lee sensor de presión
fsrReading = analogRead(fsrPin);
Serial.print("Lectura Sensor Presion : ");
Serial.print(fsrReading);
//&& fsrReading != 1023
if (fsrReading > 0 ) {
    Serial.println(" Se detectó presión, No bajara Ventana");
}
else{
    Serial.println(" Ventana trasera copiloto Abajo ...");
    servoMotor.write(0);
};
delay(3000);
//Se inicializa ventana
servoMotor.write(180);
Serial.println(" ");
}

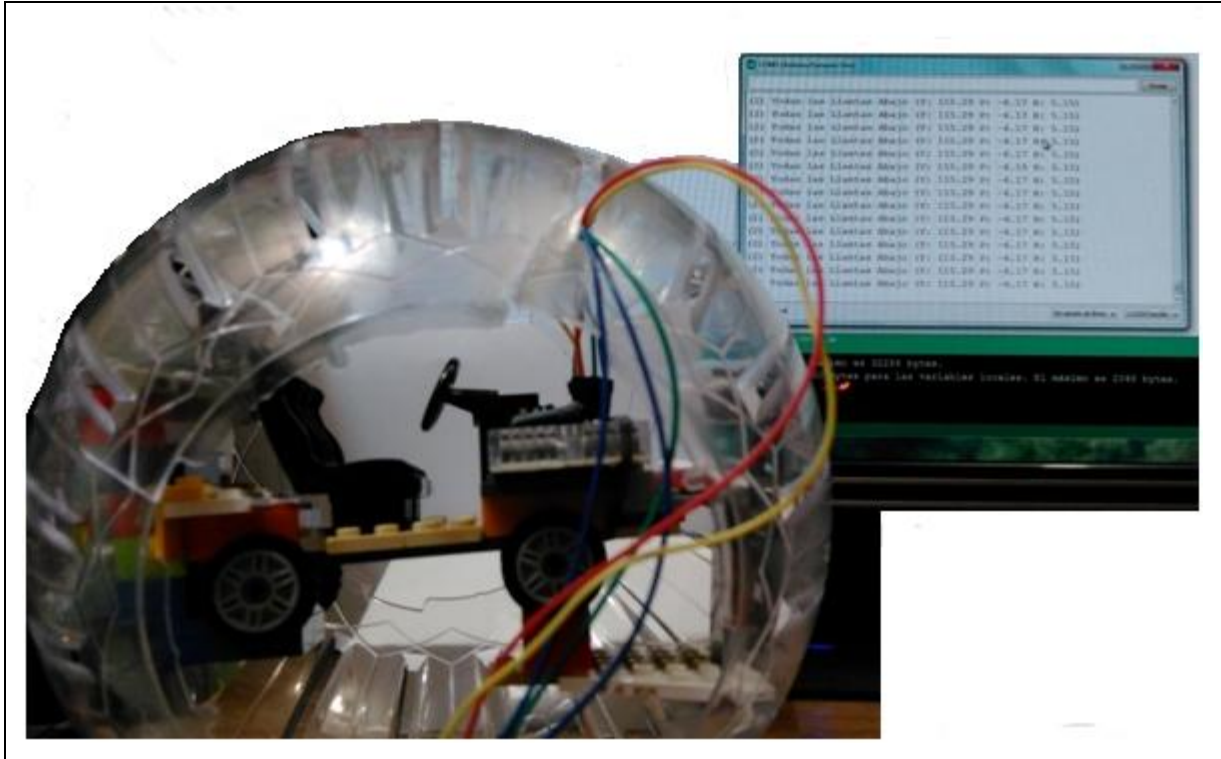
void loop() {

}

```

Anexo H. PRUEBAS

Nombre	GRG	Fecha	28/09/2017	Programa	PosicionVehiculo.ino
Nombre /Numero de Prueba	GRG/3				
Objetivo de la Prueba	Detectar posición del vehículo				
Descripción de la Prueba	Identificar la posición del vehículo y almacenar esta información en un tipo de memoria que no se borre cuando se desconecte la tarjeta Arduino de la energía. Porque esta información será utilizada más adelante para otras funcionalidades del sistema				
Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO • Tener las conexiones correctas entre el sensor acelerómetro giroscopio MPU6050 a la tarjeta Arduino UNO y esta a su vez conectada a la computadora • Abrir el monitor serial del IDE de Arduino para mostrar datos de la ejecución. • Configurar el monitor serial a 115200 baudios				
Resultados Esperados	(2) Todas las Llantas Abajo (valores de Y, P y R)				
Resultados Actuales	(2) Todas las Llantas Abajo (Y: 115.29, P: -6.17, R: 5.15)				
Evidencia Fotográfica de la ejecución					

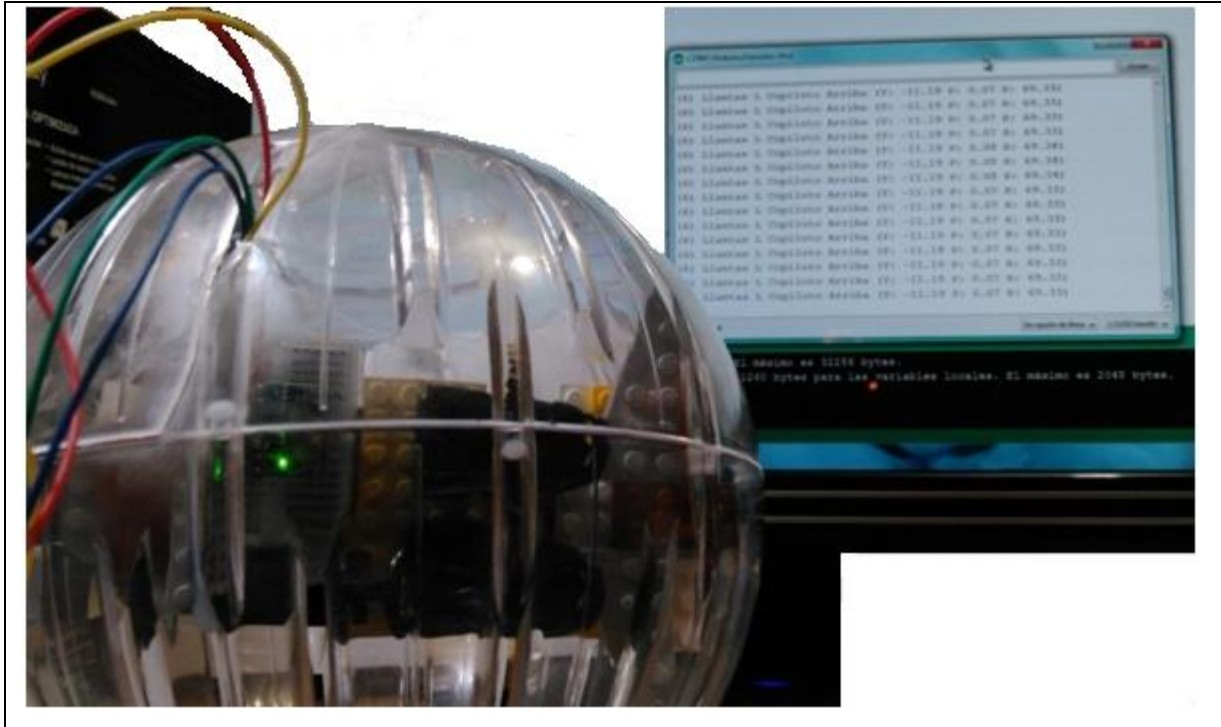


Nombre	GRG	Fecha	28/09/2017	Programa	PosicionVehiculo.ino
Nombre de Prueba	/Numero de GRG/4				
Objetivo de la Prueba	Detectar posición del vehículo				
Descripción de la Prueba	Identificar la posición del vehículo y almacenar esta información en un tipo de memoria que no se borre cuando se desconecte la tarjeta Arduino de la energía. Porque esta información será utilizada más adelante para otras funcionalidades del sistema				

Condiciones de la Prueba	- Tener instalado el programa en la tarjeta Arduino UNO - Tener las conexiones correctas entre el sensor acelerómetro giroscopio MPU6050 a la tarjeta Arduino UNO y esta a su vez conectada a la computadora - Abrir el monitor serial del IDE de Arduino para mostrar datos de la ejecución. - Configurar el monitor serial a 115200 baudios
Resultados Esperados	(4) Llantas Delanteras Arriba (valores de Y, P y R)
Resultados Actuales	(4) Llantas Delanteras Arriba (Y: 91.28, P: -77:59 , R: 2.49)
Evidencia Fotográfica de la ejecución	
	

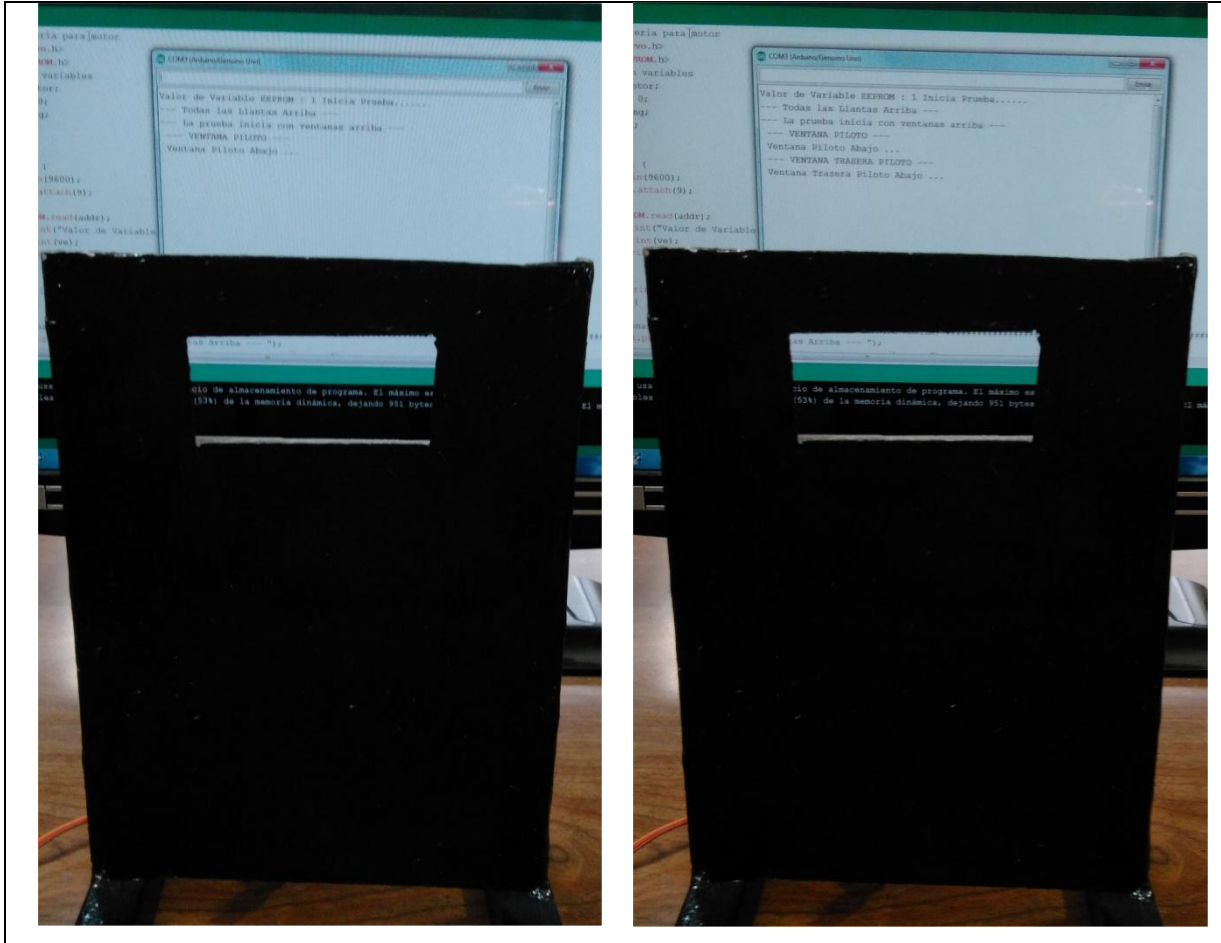
Nombre	GRG	Fecha	28/09/2017	Programa	PosicionVehiculo.ino
Nombre de Prueba	/Numero de GRG/5				

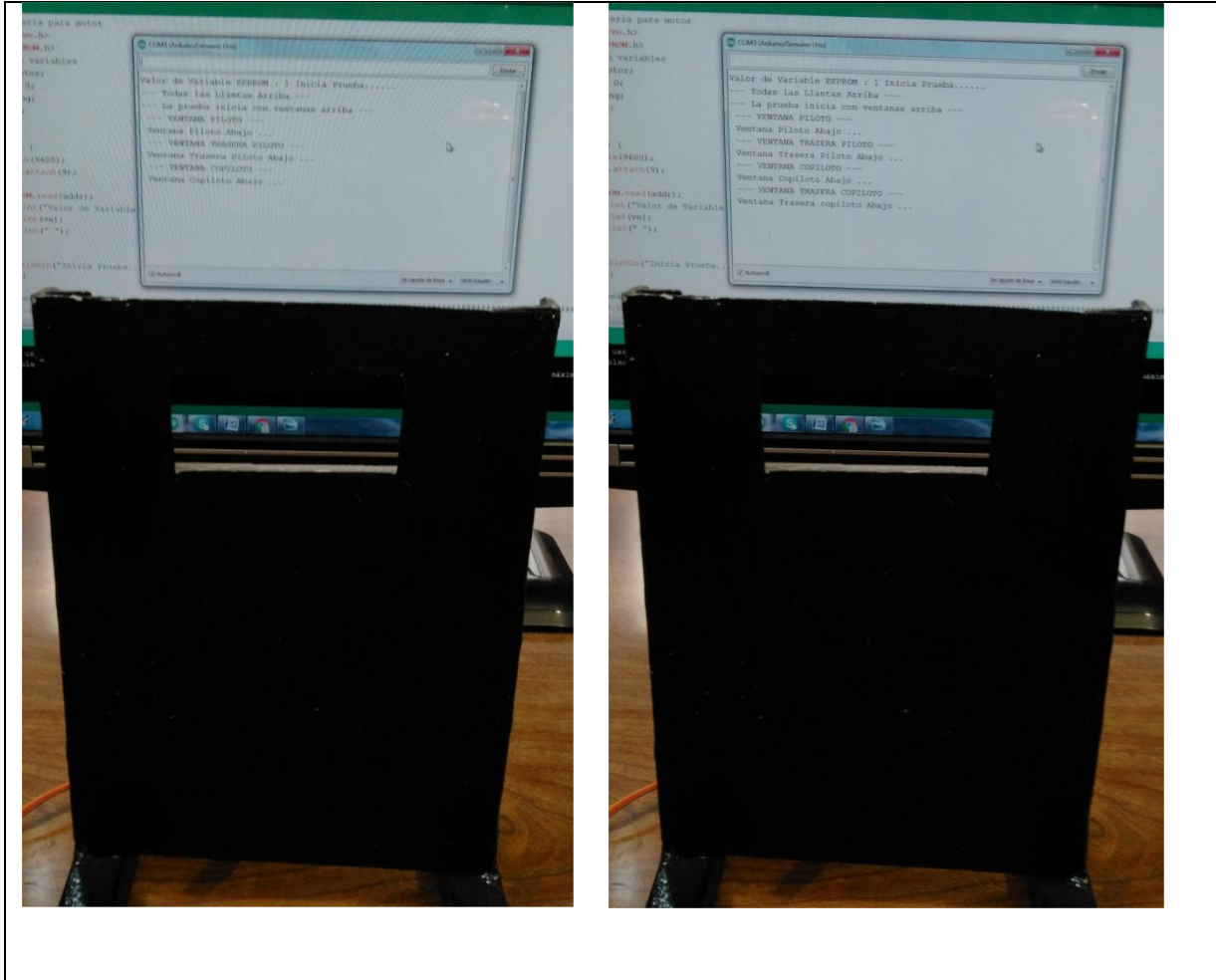
Nombre	GRG	Fecha	28/09/2017	Programa	PosicionVehiculo.ino
Nombre /Numero de Prueba	GRG/6				
Objetivo de la Prueba	Detectar posición del vehículo				
Descripción de la Prueba	Identificar la posición del vehículo y almacenar esta información en un tipo de memoria que no se borre cuando se desconecte la tarjeta Arduino de la energía. Porque esta información será utilizada más adelante para otras funcionalidades del sistema				
Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO • Tener las conexiones correctas entre el sensor acelerómetro giroscopio MPU6050 a la tarjeta Arduino UNO y esta a su vez conectada a la computadora • Abrir el monitor serial del IDE de Arduino para mostrar datos de la ejecución. • Configurar el monitor serial a 115200 baudios				
Resultados Esperados	(6) Llantas Lado Copiloto Arriba (valores de Y, P y R)				
Resultados Actuales	(6) Llantas L Copiloto Arriba (Y: -11.19, P: 0.07, R: 69.33)				
Evidencia Fotográfica de la ejecución					



Nombre	GRG	Fecha	01/10/2017	Programa	BajarVentanas.ino
Nombre de Prueba	GRG/9				
Objetivo de la Prueba	Simulación del bajado de ventanas del vehículo.				
Descripción de la Prueba	Bajar las ventanas del piloto, trasera piloto, copiloto y trasera copiloto dependiendo de la posición del vehículo previamente detectado y de la presión detectada en las ventanas.				
Condiciones de la Prueba	- Tener instalado el programa en la tarjeta Arduino UNO - Tener las conexiones correctas entre el sensor de fuerza FSR400 y el Servo Motor Micro Servo 9g SG90 Tower Pro a la tarjeta Arduino UNO y esta a su vez conectada a la computadora - Abrir el monitor serial del IDE de Arduino para mostrar				

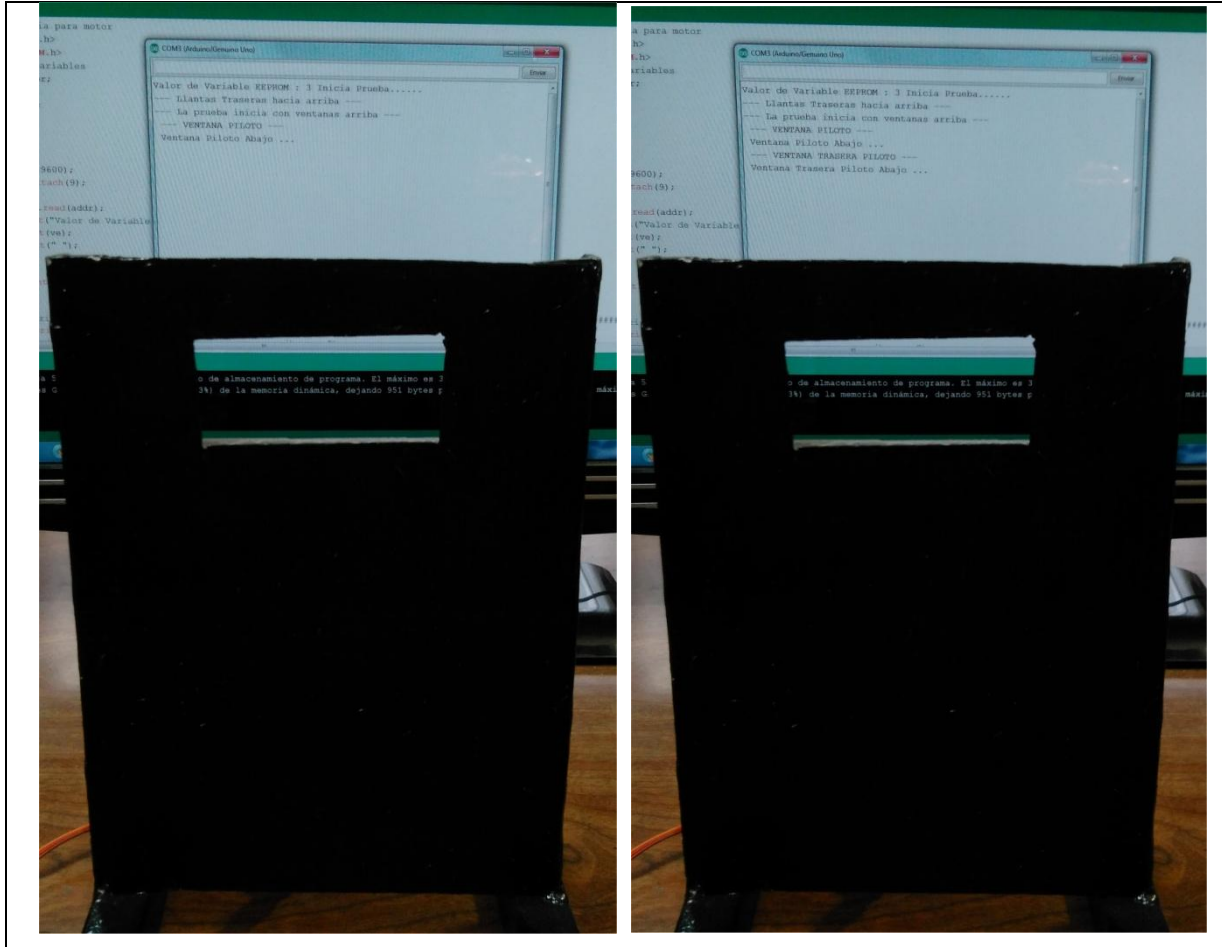
	la ejecución. • Configurar el monitor serial a 9600 baudios
Resultados Esperados	Todas las llantas arriba Ventana piloto Abajo Ventana trasera piloto Abajo Ventana copiloto Abajo Ventana trasera copiloto Abajo
Resultados Actuales	--- Todas las Llantas Arriba --- --- Ventana Piloto --- Ventana Piloto Abajo ... --- Ventana Trasera Piloto --- Ventana Trasera Piloto Abajo ... --- Ventana Copiloto --- Ventana Copiloto Abajo ... --- Ventana Trasera Copiloto --- Ventana Trasera Copiloto Abajo ...
Evidencia Fotográfica de la ejecución	

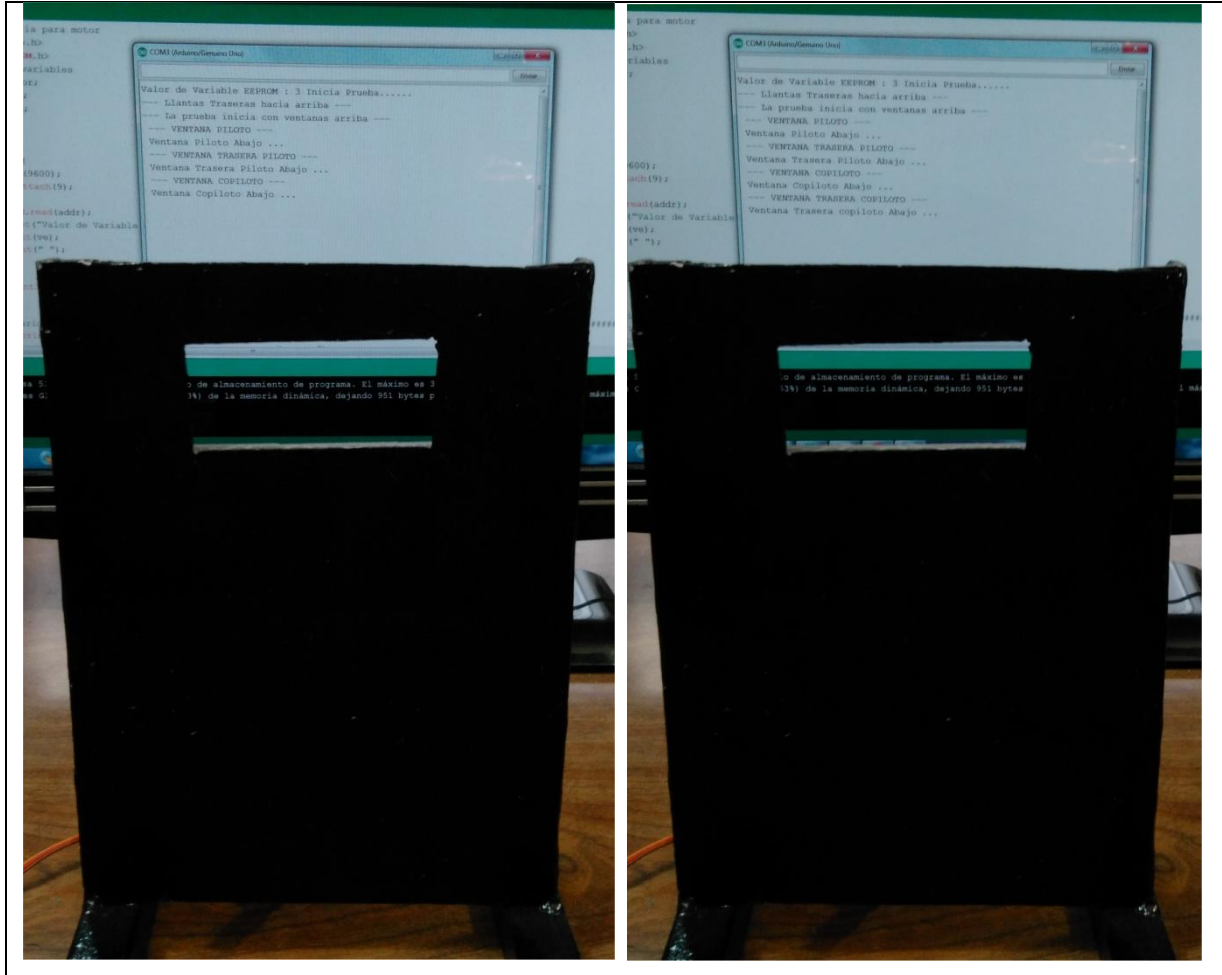




Nombre	GRG	Fecha	01/10/2017	Programa	BajarVentanas.ino
Nombre /Numero de Prueba	GRG/10				
Objetivo de la Prueba	Simulación del bajado de ventanas del vehículo.				
Descripción de la Prueba	Bajar las ventanas del piloto, trasera piloto, copiloto y trasera copiloto dependiendo de la posición del vehículo previamente detectado y de la presión detectada en las ventanas.				

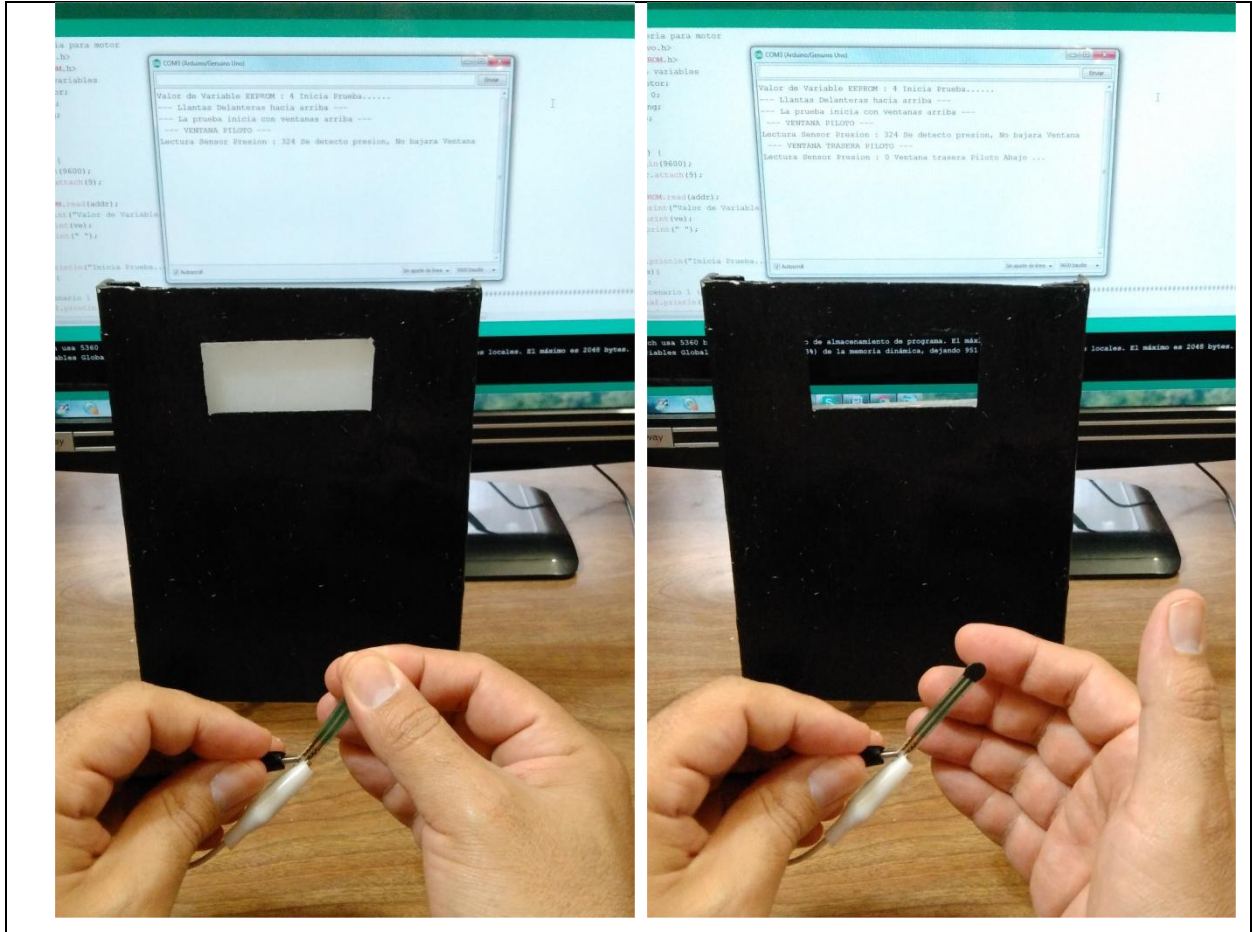
Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO • Tener las conexiones correctas entre el sensor de fuerza FSR400 y el Servo Motor Micro Servo 9g SG90 Tower Pro a la tarjeta Arduino UNO y esta a su vez conectada a la computadora • Abrir el monitor serial del IDE de Arduino para mostrar la ejecución. • Configurar el monitor serial a 9600 baudios
Resultados Esperados	Llantas traseras hacia arriba Ventana piloto Abajo Ventana trasera piloto Abajo Ventana copiloto Abajo Ventana trasera copiloto Abajo
Resultados Actuales	--- Llantas Traseras hacia arriba --- --- Ventana Piloto --- Ventana Piloto Abajo ... --- Ventana Trasera Piloto --- Ventana Trasera Piloto Abajo ... --- Ventana Copiloto --- Ventana Copiloto Abajo ... --- Ventana Trasera Copiloto --- Ventana Trasera Copiloto Abajo ...
Evidencia Fotográfica de la ejecución	

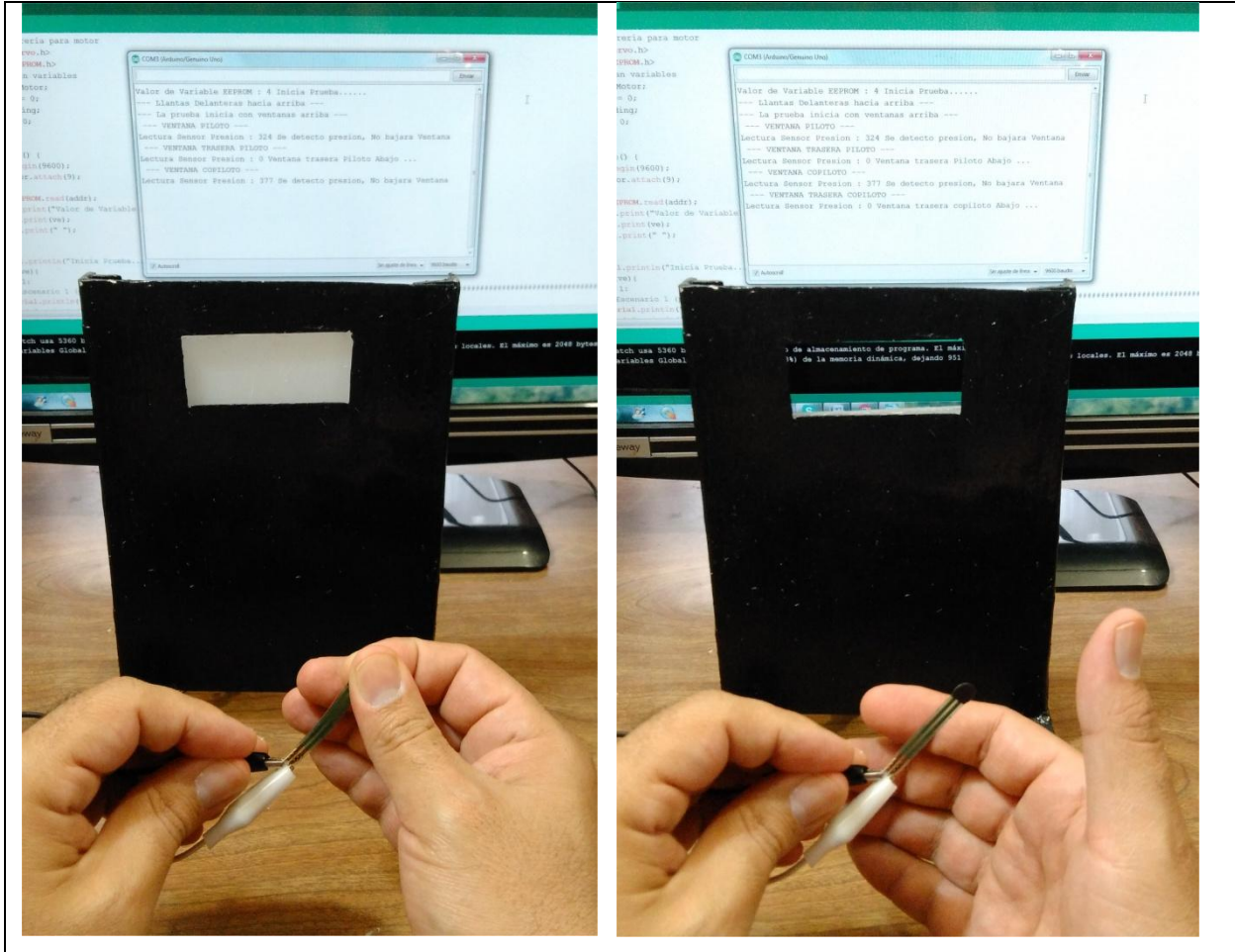




Nombre	GRG	Fecha	01/10/2017	Programa	BajarVentanas.ino
Nombre de Prueba	GRG/11				
Objetivo de la Prueba	Simulación del bajado de ventanas del vehículo.				
Descripción de la Prueba	Bajar las ventanas del piloto, trasera piloto, copiloto y trasera copiloto dependiendo de la posición del vehículo previamente detectado y de la presión detectada en las ventanas.				

Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO • Tener las conexiones correctas entre el sensor de fuerza FSR400 y el Servo Motor Micro Servo 9g SG90 Tower Pro a la tarjeta Arduino UNO y esta a su vez conectada a la computadora • Abrir el monitor serial del IDE de Arduino para mostrar la ejecución. • Configurar el monitor serial a 9600 baudios
Resultados Esperados	Llantas delanteras hacia arriba Ventana piloto Detectó presión, arriba Ventana trasera piloto No se detectó presión, abajo Ventana copiloto Detectó presión, arriba Ventana Trasera copiloto No se detectó presión, abajo
Resultados Actuales	--- Llantas Delanteras hacia arriba --- --- Ventana Piloto --- Lectura Sensor Presión : 324 Se detectó presión, No bajará ventana --- Ventana Trasera Piloto --- Lectura Sensor Presión : 0 Ventana Trasera Piloto Abajo ... --- Ventana Copiloto --- Lectura Sensor Presión : 377 Se detectó presión, No bajará ventana --- Ventana Trasera Copiloto --- Lectura Sensor Presión : 0 Ventana Trasera Copiloto Abajo ...
Evidencia Fotográfica de la ejecución	





Nombre	GRG	Fecha	01/10/2017	Programa	BajarVentanas.ino
Nombre de Prueba	GRG/12				
Objetivo de la Prueba	Simulación del bajado de ventanas del vehículo.				
Descripción de la Prueba	Bajar las ventanas del piloto, trasera piloto, copiloto y trasera copiloto dependiendo de la posición del vehículo previamente detectado y de la presión detectada en las ventanas.				

Condiciones de la Prueba	• Tener instalado el programa en la tarjeta Arduino UNO • Tener las conexiones correctas entre el sensor de fuerza FSR400 y el Servo Motor Micro Servo 9g SG90 Tower Pro a la tarjeta Arduino UNO y esta a su vez conectada a la computadora • Abrir el monitor serial del IDE de Arduino para mostrar la ejecución. • Configurar el monitor serial a 9600 baudios
Resultados Esperados	Llantas lado copiloto hacia arriba Ventana piloto Arriba Ventana trasera piloto Arriba Ventana copiloto Abajo Ventana trasera copiloto Abajo
Resultados Actuales	--- Llantas lado del Copiloto hacia arriba --- --- Ventana Piloto --- Ventana Piloto Arriba ... --- Ventana Trasera Piloto --- Ventana Trasera Piloto Arriba ... --- Ventana Copiloto --- Ventana Copiloto Abajo ... --- Ventana Trasera Copiloto --- Ventana Trasera Copiloto Abajo ...
Evidencia Fotográfica de la ejecución	

