

TECNOLÓGICO DE ESTUDIOS SUPERIORES DE COACALCO

Testeo de software automotriz para
área de infotainment

Estancias

Que para obtener el título de
Ingeniero en Mecatrónica

Con la especialidad de automatización

Presenta:

Joshua Armando Luna Miranda
201420031

Asesor:

Ing. Francisco Javier Sánchez Rico

Coacalco de Berriozábal, Septiembre 2024

RESUMEN

El presente informe, detalla las experiencias del aprendizaje adquiridos durante las estancias realizadas como ingeniero en el área de testeo de software automotriz como parte del proceso de obtención del título de ingeniero en mecatrónica, el tema central de este informe es presentar, evaluar y validar algunas de las técnicas de testeo de software utilizadas y como refuerzan los conocimientos teóricos adquiridos durante carrera de Ingeniería Mecatrónica, el área de testeo y validación de software es un componente critico en la industria automotriz moderna debido a la creciente complejidad y sofisticación de los sistemas electrónicos en los vehículos.

ÍNDICE

RESUMEN	2
ÍNDICE.....	3
ÍNDICE DE TABLAS Y FIGURAS	5
CAPÍTULO I GENERALIDADES DEL PROYECTO	6
Introducción	7
Problema a resolver	8
Objetivos	9
Justificación	10
Alcance y limitaciones	11
CAPÍTULO II MARCO TEÓRICO.....	13
2.1 Infotainment	14
2.2 Software	14
2.3 ISTQB	15
2.4 Protocolo de comunicación.....	15
2.5 CAN BUS	17
2.6 CAN FD.....	17
2.7 Bus LIN.....	18
2.8 AUTOSAR	19
2.9 Requerimiento	19
2.10 CANalyzer	20
CAPÍTULO III DESARROLLO	21
3.1 Introducción al proceso de validación de software.	22
3.2 Paso 1	22
3.3 Paso 2	23
3.4 Paso 3	23
3.5 Paso 4.	24
3.6 Paso 5	24
3.7 Paso 6	25
CAPÍTULO IV RESULTADO.....	26

4.1 Resultados	27
Conclusiones	28
Objetivos educacionales	29
Fuentes de información	30
Anexos.....	32

ÍNDICE DE TABLAS Y FIGURAS

Figura 1. Metodología AUTOSAR	19
Figura 2. Definición de fechas para las rondas productivas.....	22
Figura 1. Vista general de la página principal del programa	23
Figura 2. Ronda productiva de features a testear	23
Figura 5. Imagen de la ronda productiva	24
Figura 6. Comentarios en sistema para consulta	25
Figura 3. Documentación dentro de un request terminado	25

CAPÍTULO I

GENERALIDADES DEL PROYECTO

INTRODUCCIÓN

La elaboración de este informe se realiza con el objetivo de documentar y analizar algunas de las metodologías, herramientas y prácticas utilizadas en el testeado de software automotriz. Además, se pretende resaltar la importancia de la calidad y fiabilidad del software en el funcionamiento seguro y eficiente en los vehículos.

Este informe también busca proporcionar una visión integral de las competencias y conocimientos adquiridos durante la carrera de Ingeniería Mecatrónica y su relación con las nuevas competencias adquiridas durante las estancias, así como su aplicación práctica en el campo de la ingeniería mecatrónica en general.

Esta introducción sirve como marco para comprender el contenido y relevancia del informe, se establecen las bases para el análisis detallado de las actividades realizadas como ingeniero de testeado de software automotriz en el área de infotainment y los resultados obtenidos en dicha área.

PROBLEMA A RESOLVER

En la actualidad, la industria automotriz enfrenta nuevos desafíos debido a la rápida evolución de la tecnología y a la creciente integración de sistemas electrónicos y de software cada vez más complejos en los vehículos. Estos avances, aunque ofrecen numerosas ventajas en términos de funcionalidad y seguridad, también incrementan la complejidad de los sistemas y, por ende, la probabilidad de fallos y errores. Un área crítica en este contexto es el testeado de software que se encarga de garantizar que los sistemas de software funcionen correctamente y de manera segura bajo diversas condiciones.

El principal problema que se aborda en este informe es la necesidad de contar con metodologías y herramientas eficaces para la validación y verificación de software automotriz. Esta necesidad surge debido a varios factores:

1. Complejidad de los sistemas. Los vehículos modernos incorporan una gran cantidad de sistemas interconectados que deben funcionar de manera armoniosa. La complejidad de estos sistemas requiere métodos de testeado avanzados que puedan detectar errores y es asegurar la integridad de software.
2. Seguridad y Fiabilidad. La seguridad es una preocupación primordial en la industria automotriz. Un fallo en el software puede tener consecuencias graves, tanto para los ocupantes del vehículo, como para otros usuarios de la vía.
3. Regulaciones y Normativas. Las regulaciones en la industria automotriz son cada vez más estrictas, especialmente en lo que respecta a la seguridad (sistemas como ABS, cinturones de seguridad, cámaras, radares, sensores, etc.) y la eficiencia de los vehículos (decibeles emitidos por los sistemas de escape, emisiones al medio ambiente y/o vida útil de componentes eléctricos y baterías).
4. Costos y Tiempo. Los procesos de testeado y validación son costosos y pueden prolongar el tiempo de desarrollo de los vehículos, sin embargo, detectar los posibles errores en una edad temprana del desarrollo del vehículo puede garantizar un ahorro en recursos de producción y tiempo.

OBJETIVOS

Objetivo General

Presentar, analizar y evaluar las metodologías y prácticas utilizadas en el testeado de software automotriz del área de infotainment, con el fin de contribuir al desarrollo software seguro y confiable en la industria automotriz.

Objetivos específicos.

1. Describir las metodologías utilizadas en el proceso de testeado de software, con el propósito de proporcionar un entendimiento detallado de las técnicas empleadas, dirigido a los estudiantes y profesionales en ingeniería mecatrónica.
2. Identificar las fortalezas y debilidades de las prácticas de testeado observadas durante las estancias, utilizando un análisis comparativo, para mejorar la eficiencia y efectividad del testeado de software, orientado a los equipos de desarrollo y calidad en la industria automotriz.
3. Proponer mejoras para optimizar el proceso del testeado del software, basándose en las experiencias y análisis realizados durante las estancias, con el objetivo de incrementar la calidad y reducir los costos de desarrollo, orientado a las empresas del sector automotriz.
4. Difundir mediante este informe los conocimientos de aprendizaje adquiridos durante las estancias, con el propósito de fomentar la formación y actualización de los futuros ingenieros mecatrónicos en técnicas avanzadas de testeado de software.

JUSTIFICACIÓN

La elección de trabajar en el área de testeo de software automotriz para la realización de este informe responde a varios motivos personales, académicos y sociales.

Motivo personal: Desde siempre, he tenido un gran interés por la industria automotriz. La complejidad y la innovación constante en este campo han sido una fuente de inspiración y motivación para mí. Trabajar en el área de testeo de software automotriz no solo me permite estar inmerso en una de las áreas más dinámicas y tecnológicamente avanzadas, sino que también me brinda la oportunidad de aplicar y expandir mis conocimientos en ingeniería mecatrónica.

Motivo académico: Este informe es una pieza clave para la obtención de mi título en ingeniería mecatrónica. A través de las estancias y el desarrollo de este proyecto, puedo demostrar mis competencias y habilidades adquiridas durante mi formación académica. Además, la elaboración de este informe me permite integrar y aplicar de manera práctica los conceptos teóricos aprendidos, consolidando mi preparación como profesional en el campo de la mecatrónica.

Motivo social: La industria automotriz es fundamental para el desarrollo tecnológico de la sociedad. Contribuir a esta industria mediante la mejora de los procesos en este caso de testeo de software es una manera de apoyar el avance tecnológico y la seguridad en el transporte.

En resumen, mi motivación para trabajar en el testeo del software automotriz surge del deseo de combinar mi pasión personal por la industria automotriz, cumplir con los requisitos académicos para obtener mi título, y contribuir al desarrollo y mejora de procesos en una industria tan importante para la sociedad.

ALCANCE Y LIMITACIONES

Alcance

El presente informe académico se centra en el presentar, analizar y evaluar algunas de las técnicas de testeo de software automotriz, abarcando tanto su funcionamiento como los resultados esperados. Durante el desarrollo del informe, se detallarán las metodologías empleadas en el testeo de software automotriz, proporcionando una visión general de las practicas observadas y aplicadas durante las estancias. Recordando que el principal objetivo del testeo de software es la detección oportuna de errores y en la mejora de la seguridad y funcionalidad del software automotriz.

El informe también abordara la importancia de un testeo riguroso y sistemático, destacando su papel crucial en la industria automotriz moderna. Se proporcionarán ejemplos y casos específicos para ilustrar como las diferentes técnicas de testeo contribuyen a la calidad y fiabilidad del software, asegurando que los vehículos operen de manera segura y eficiente.

Limitaciones

1. Confidencialidad de la información: debido a la naturaleza confidencial de muchos datos y procesos en la industria automotriz, algunas técnicas específicas y resultados detallados no podrán ser presentados en su totalidad. Esto limita la capacidad de proporcionar ejemplos completamente exhaustivos y detallados en algunos aspectos del testeo de software.
2. Alcance temporal y espacial: El informe se basa en las experiencias y observaciones realizadas durante un periodo especifico de estancias en determinada empresa del sector automotriz. Por lo tanto, los hallazgos y conclusiones reflejan las prácticas y condiciones de ese periodo particular, y pueden no ser representativos de todas las prácticas de testeo de software automotriz a nivel global.
3. Enfoque en Técnicas específicas: Si bien se describirán y analizarán diversas técnicas de testeo, el informe no pretende cubrir exhaustivamente todas las metodologías existentes en el campo del testeo de software automotriz.
4. Limitaciones técnicas y practicas: Algunas limitaciones técnicas y practicas innatas a las herramientas y métodos utilizados en el testeo del software también restringen

la capacidad de realizar un análisis completamente exhaustivo. Estas limitaciones incluyen la disponibilidad de recursos y las restricciones impuestas por las propias herramientas de testeo.

A pesar de estas limitaciones, se ha realizado un esfuerzo significativo para proporcionar una visión clara y precisa de las técnicas de testeo de software automotriz y su impacto en la confiabilidad y seguridad del producto final. Este informe busca contribuir al conocimiento académico y profesional en el campo de la ingeniería mecatrónica y la industria automotriz.

CAPÍTULO II

MARCO TEÓRICO

2.1 Infotainment

Desde la navegación vía satélite hasta los sistemas avanzados de ayuda a la conducción, las tecnologías modernas ya garantizan el registro de todos los datos importantes del vehículo, se lo ofrecen al conductor y lo vinculan con servicios de Internet. Para el conductor, esto significa más comodidad, eficacia y control sobre el entorno vehicular.

La gestión moderna del infoentretenimiento significa nada más y nada menos que la visualización simple de información importante del vehículo. El diálogo holístico entre el hombre y la máquina garantiza un intercambio simple de información entre la persona, el vehículo y el ambiente exterior. En un mundo cada vez más complejo, la gestión de la información en el automóvil también se simplifica y abre las puertas a nuevas funciones vehiculares, como la conducción automatizada y la integración simple de los dispositivos móviles.

Infoentretenimiento en el automóvil. (2022, abril 1). Continental AG. <https://www.continental.com/es-mx/products-innovation/innovation/infoentretenimiento-en-el-automovil/>

2.2 Software

Software es un término informático que hace referencia a un programa o conjunto de programas de cómputo, así como datos, procedimientos y pautas que permiten realizar distintas tareas en un sistema informático.

Comúnmente se utiliza este término para referirse de una forma muy genérica a los programas de un dispositivo informático, sin embargo, el software abarca todo aquello que es intangible en un sistema computacional.

De, E. (2014, mayo 6). *Software*. Enciclopedia Significados. <https://www.significados.com/software/>

2.3 ISTQB

ISTQB® es el esquema de certificación global líder en el campo de pruebas de software.

ISTQB® estableció y continúa desarrollando el programa Certified Tester reconocido internacionalmente, que es una cartera de certificaciones que desarrollan, amplían y validan las habilidades de evaluación. El programa se basa en programas de estudio creados por una red global de expertos, el glosario ISTQB®, exámenes de muestra y el Testing Body of Knowledge (TBOK).

La certificación Certified Tester Foundation Level proporciona a los profesionales de las pruebas una terminología esencial y una gran cantidad de conocimientos. Es un requisito previo para los demás módulos del programa que ofrecen profundidad y especialización.

What we do. (s/f). ISTQB Not-for-Profit Association. Recuperado el 5 de agosto de 2024, de <https://www.istqb.org/about-us/what-we-do>

2.4 Protocolo de comunicación

El protocolo, se define como las reglas para la transmisión de la información entre dos puntos. Un protocolo de red de comunicación de datos es un conjunto de reglas que gobierna el intercambio ordenado de datos dentro de la red.

Los elementos básicos de un protocolo de comunicaciones son: un conjunto de símbolos llamados conjunto de caracteres, un conjunto de reglas para la secuencia y sincronización de los mensajes construidos a partir del conjunto de caracteres y los procedimientos para determinar cuando ha ocurrido un error en la transmisión y como corregir el error. El conjunto de caracteres se formará de un subconjunto con significado para las personas (usualmente denominado como caracteres imprimibles) y otro subconjunto que transmite información de control (usualmente denominado caracteres de control). Hay una correspondencia entre cada carácter y los grupos de símbolos usados en el canal de transmisión, que es determinado por el código. Muchos códigos estándar con sus respectivas equivalencias de grupos de unos y ceros (bits) han sido definidos con el paso de los años. El conjunto de reglas a seguir por el emisor y el receptor propicia: que haya un

significado con secuencias permitidas y a tiempo, entre los caracteres de control y los mensajes formados a partir de los símbolos. La detección de error y los procedimientos de corrección permiten la detección y la recuperación ordenada de los errores causados por factores fuera del control de la terminal en cada extremo.

Para que exista comunicación en ambos puntos al extremo de un canal se deben emplear la misma configuración de protocolos.

Los protocolos gestionan dos niveles de comunicación distintos. Las reglas de alto nivel definen como se comunican las aplicaciones, mientras que las de bajo nivel definen como se transmiten las señales.

El protocolo de bajo nivel es básicamente la forma en que las señales se transmiten, transportando tanto datos como información y los procedimientos de control de uso del medio por los diferentes nodos. Los protocolos de bajo nivel más utilizados son: Ethernet, Token ring, Token bus, FDDI, CDDI, HDLC, Frame Relay y ATM.

El protocolo de red determina el modo y organización de la información (tanto los datos como los controles) para su transmisión por el medio físico con el protocolo de bajo nivel. Los protocolos de red más comunes son: IPX/SPX, DECnet, X.25, TCP/IP, AppleTalk y NetBEUI.

En un circuito de comunicación de datos, la estación que transmite en el momento se llama estación maestra, y la estación que recibe se llama esclava. En una red centralizada, la estación primaria controla cuándo puede transmitir cada estación secundaria. Cuando transmite una estación secundaria se convierte en estación maestra, y la estación primaria es ahora la esclava. El papel de estación maestra es temporal, y la estación primaria determina cuál estación es maestra. Al principio, la estación primaria es maestra. La estación primaria solicita por turno a cada estación secundaria interrogándola. Una interrogación es una invitación de la primaria a una secundaria para que transmita un mensaje. Las estaciones secundarias no pueden interrogar a una primaria. Cuando una primaria interroga a una secundaria, inicia un cambio de dirección de línea; la secundaria interrogada ha sido designada como maestra y debe responder. Si la primaria selecciona una secundaria, ésta se

identifica como receptora. Una selección es una interrogación, por parte de una primaria o una secundaria, para determinar el estado de la secundaria (es decir, lista para recibir o no lista para recibir un mensaje). Las estaciones secundarias no pueden seleccionar a la primaria. Las transmisiones de la primaria van a todas las secundarias, y depende de las estaciones secundarias la decodificación individual de cada transmisión, y la determinación de si es para ellas. Cuando una secundaria transmite, sólo manda a la primaria.

Los protocolos de enlace de datos se clasifican en general como: asíncronos o síncronos. Por regla, los protocolos asíncronos usan un formato de datos asíncronos y módems asíncronos, mientras que los protocolos síncronos usan un formato de datos síncronos y módems síncronos.

Descripción general de los protocolos. Unam.mx:8080. Recuperado el 5 de agosto de 2024, <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/734/A6.pdf?sequence=6>

2.5 CAN BUS

La red de área del controlador: el bus CAN es un protocolo basado en mensajes diseñado para permitir que las unidades de control electrónico (ECU) que se encuentran en los automóviles actuales, así como otros dispositivos, se comuniquen entre sí de manera confiable y basada en prioridades. Todos los dispositivos de la red reciben mensajes o "tramas", lo que no requiere una computadora host. CAN está respaldado por un amplio conjunto de estándares internacionales bajo ISO 11898.

¿Qué es Can Bus (red de área de controlador)? (s/f). Soluciones de Adquisición de Datos (DAQ). Recuperado el 5 de agosto de 2024, de <https://dewesoft.com/es/blog/que-es-el-bus-can>

2.6 CAN FD

CAN FD es una versión de "datos flexibles (velocidad)" del bus CAN. La longitud estándar de cada mensaje se ha incrementado en un 800% a 64 bytes, y la velocidad máxima de

datos se ha incrementado de forma similar de 1 Mbps a 8 Mbps. La parte "flexible" se refiere al hecho de que las ECU pueden cambiar dinámicamente sus velocidades de transmisión y seleccionar tamaños de mensajes más grandes o más pequeños, según los requisitos en tiempo real.

A pesar de todos estos avances, CAN FD sigue siendo completamente retrocompatible con CAN 2.0 estándar. Hoy en día, CAN FD se encuentra en vehículos de muy alto rendimiento, pero se espera que eventualmente migre a todos o la mayoría de los vehículos.

¿Qué es Can Bus (red de área de controlador)? (s/f). Soluciones de Adquisición de Datos (DAQ). Recuperado el 5 de agosto de 2024, de <https://dewesoft.com/es/blog/que-es-el-bus-can>

2.7 Bus LIN

En el Sistema de multiplexado de red de datos encontraremos el termino, así como la red «LIN-Bus que es la abreviatura de Local Interconnect Network. El LIN-Bus es una extensión del bus de datos CAN. El sistema de bus LIN es un bus de un solo cable. El cable tiene un color básico (violeta) y un código de color. La sección transversal del cable es de 0,35 mm². Posee una velocidad de transmisión de 20Kbit/s, y es unidireccional (Se transmite en una sola dirección).

El bus LIN se diseñó para hacer la conexión de actuadores o sensores esta conexión proviene de las unidades de control. Desde la unidad de control maestra al sensor o actuador conectados en sentido descendente, que se les nombra «esclavo». El maestro puede transmitir órdenes hasta a 16 elementos esclavos conectados en sentido descendente. La unidad Maestra proviene previamente conectada del CAN BUS Asume la función de traducción entre las unidades de control LIN abonadas al sistema del LIN-Bus local y el CAN-Bus de datos.

Mecanica Automotriz, I. Y. (2021, septiembre 22). *¿Qué es el BUS Lin y cómo funciona?*
 INGENIERÍA Y MECÁNICA AUTOMOTRIZ.
<https://www.ingenieriaymecanicaautomotriz.com/que-es-el-bus-lin-y-como-funciona/>

2.8 AUTOSAR

AUTOSAR (AUTomotive Open System ARchitecture) es una asociación global de empresas líderes en la industria automotriz y de software para desarrollar y establecer el marco de software estandarizado y la arquitectura de sistema E/E abierta para la movilidad inteligente.

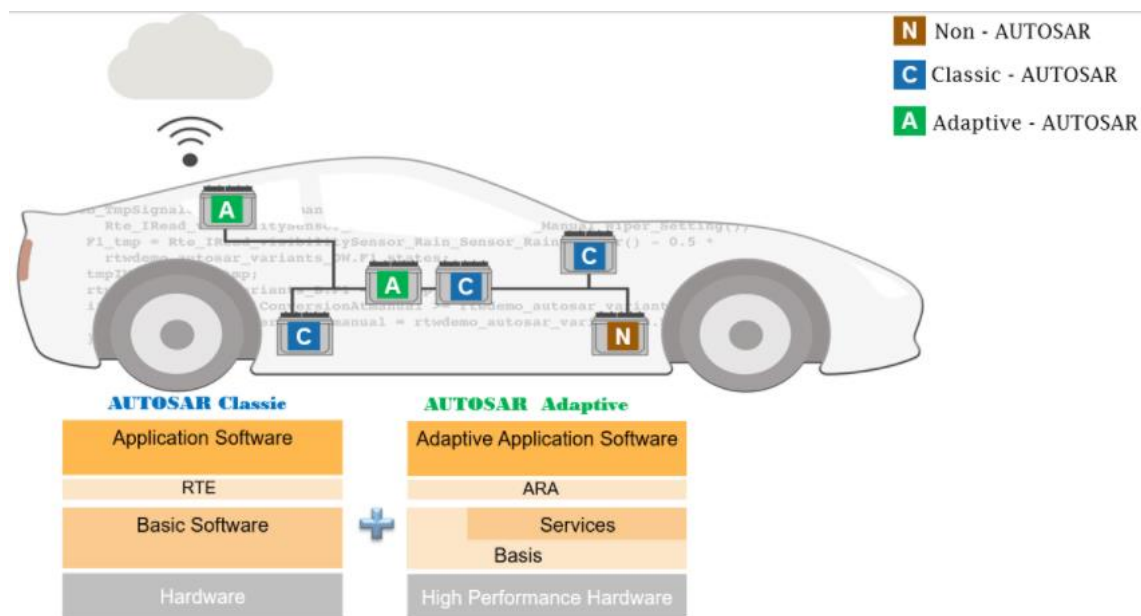


Figura 4. Metodología AUTOSAR.

Reichart, G. (2024, julio 24). *AUTOSAR enabling continuous improvement*. Autosar.org. <https://www.autosar.org/>

2.9 Requerimiento

En el mundo del desarrollo de software y aplicaciones, los requerimientos son críticos. Son los que definen la funcionalidad y el propósito de una pieza particular de software o aplicación. Sin estos bien definidos, es difícil crear algo que satisfaga las necesidades y expectativas del usuario.

Son simplemente una descripción de lo que un programa de software en particular debe hacer. Actúan como pautas para que los desarrolladores creen un producto funcional que satisfaga las necesidades de los usuarios. Siga leyendo para descubrir por qué los requerimientos son importantes para usted como ingeniero de software.

Northware. (2022, mayo 26). *Requerimientos en el desarrollo de software y aplicaciones*. Northware. <https://www.northware.mx/blog/requerimientos-en-el-desarrollo-de-software-y-aplicaciones/>

2.10 CANalyzer

CANalyzer es una herramienta de software integral con operación intuitiva para el análisis y estimulación de la comunicación en red en la industria automotriz e industrias relacionadas.

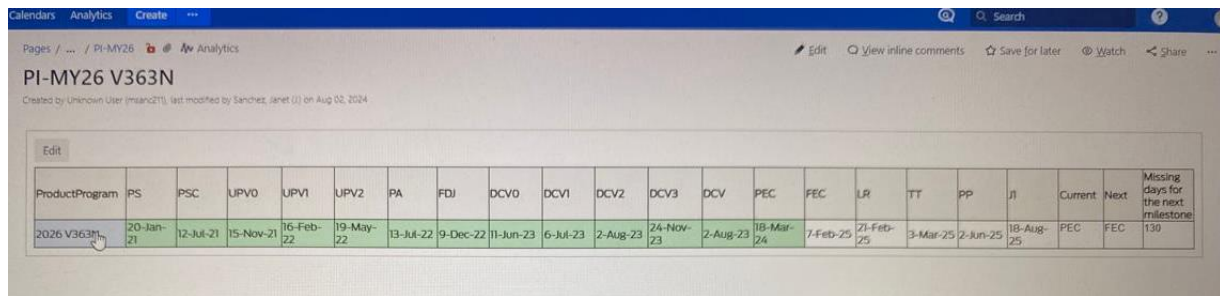
CANalyzer. (s/f). Vector Informatik GmbH. Recuperado el 5 de agosto de 2024, de <https://www.vector.com/us/en/products/products-a-z/software/canalyzer/>

CAPÍTULO III

DESARROLLO

3.1 Introducción al proceso de validación de software.

Durante el periodo de testeo de software se definen 3 rondas de validación y se establecen los tiempos en que dichas rondas deben de ser completadas. El analista del programa (vehículo a testear), realiza un análisis completo y detallado del vehículo a testear que incluye para que mercado va dirigido, variante, equipamiento, los módulos (pantalla, radio, cluster, ECU's, etc.), información de equipamiento, contactos de los módulos, los features (funcionalidades) a testear, así como el capital humano y económico, necesario para poder terminar las rondas de validación en los tiempos establecidos y completar los features al 100%.



Product/Program	PS	PSC	UPV0	UPV1	UPV2	PA	FDI	DCV0	DCV1	DCV2	DCV3	DCV	PEC	FEC	LR	TT	PP	JI	Current	Next	Missing days for the next milestone
2026 V363N	20-Jan-21	12-Jul-21	15-Nov-21	16-Feb-22	19-May-22	13-Jul-22	9-Dec-22	11-Jun-23	6-Jul-23	2-Aug-23	24-Nov-23	2-Aug-23	18-Mar-24	7-Feb-25	21-Feb-25	3-Mar-25	2-Jun-25	18-Aug-25	PEC	FEC	130

Figura 5. Definición de fechas para las rondas productivas.

3.2 Paso 1

Como primera actividad del tester una vez que se tienen los módulos del vehículo para testear, deberá de verificar en que versión de software fue entregado por el proveedor y si es necesario contactar al D&R del módulo para obtener el ultimo software y poder realizar la actualización. En paralelo de acuerdo a la información del mercado, variante, equipamiento los módulos se deben de configurar ya que de esto depende las funcionalidades activas del área de infotainment, como pueden ser tipo pantalla de inicio, imágenes del vehículo a mostrar, tipo de radio (frecuencias activas AM, FM, HD o satelitales), asistencias visuales como cámara 360, cámara de reversa, numero de sensores frontales, traseros o laterales, asistencias de estacionamiento, funcionalidades de conectividad Wi-Fi, bluetooth, aplicaciones Wireless, etc.

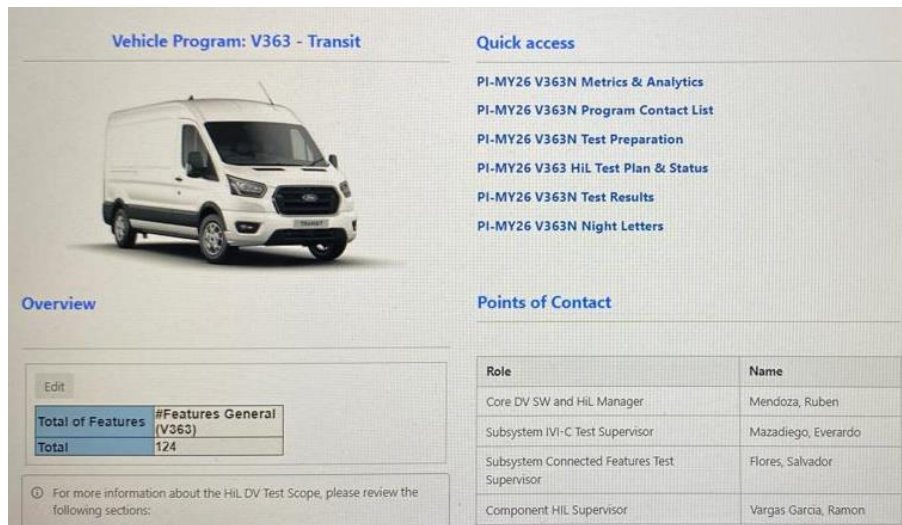


Figura 6. Vista general de la página principal del programa.

3.3 Paso 2

Una vez que se comienza las rondas de validación el tester comienza con los diferentes features, realizando los test cases que se enfocan en probar las diferentes funcionalidades del sistema de infotainment, para ello el material de apoyo son la documentación de variante y mercados ya que en ellos se encuentran descritas que funcionalidades están activas.

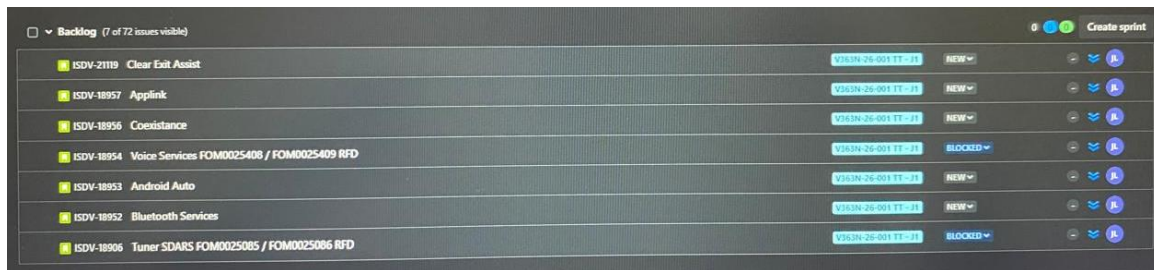


Figura 7. Ronda productiva de features a testear.

3.4 Paso 3

Durante la ronda de validación el tester se enfoca en realizar los test cases los cuales prueban diferentes escenarios para el sistema de infotainment como si fuera el usuario final, para ello es necesario revisar los requerimientos del feature, ya que en este documento se encuentra detallado el comportamiento esperado del software, cuales son los casos de excepción, así como información detallada del feature.

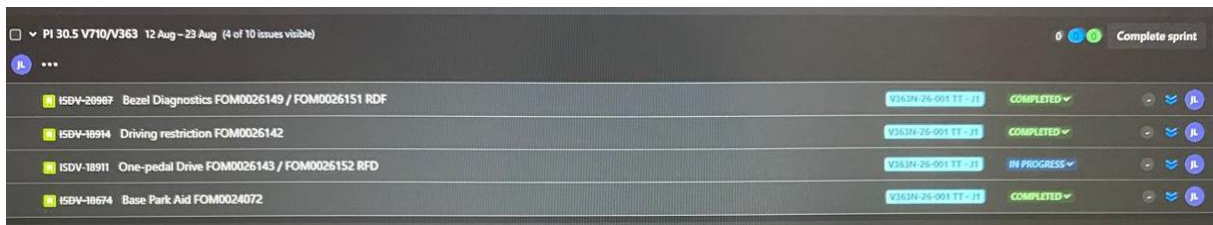


Figura 8. Imagen de la ronda productiva.

3.5 Paso 4

En case encontrar un issue durante la ronda de validación, el tester levanta un ticket donde menciona información general que es, versión del software donde se encontró, modulo donde se propone está el issue, pasos para reproducirlo, información del test case, logs de los módulos involucrados, archivos de configuración de los módulos involucrados y una imagen o video como referencia visual del issue.

Una vez que se levantó el ticket se le da un seguimiento hasta que se resuelve dicho issue, ya sea en una nueva versión de software o porque es una funcionalidad que ya no está activa, para ambos casos se debe de comentar cual fue la resolución del equipo al cual se le asignó el ticket y se sube nueva evidencia para poder cerrar el ticket.

3.6 Paso 5

Una vez que se terminó un feature el tester sube evidencia del sistema que incluye, archivos de configuración de los módulos, archivo de versión de software de los módulos, archivo del número de test cases del feature donde indica la cantidad de pruebas pasadas, las que no aplicaron al vehículo (con su comentario puede ser por configuración, variante o mercado), las pruebas no completadas (por falta de material, módulos etc.), y las pruebas falladas debido a que se encontró un bug en el software.

Bezel Diagnostics FOM0026149		
Result	Qty	Comments
Passed	38	
Failed	0	
Not Tested	0	
Blocked	0	
Not Applicable	12	(139669 139671 139692 139696 139697 166789 181501 181506 181515 181519 139703 139706) According to the VSCS-2026-V363N-AJ-TT-001-WIP_08-05-2024_18-09-16: DSP= \$0 Disable Module: APIM

Development [Create branch](#) [Create commit](#)

Labels [RFD](#)

Target Program Increment None

Sprint [PI 30.5 V710/V363](#)

Vehicle Program [2026 V363 ICA3 NA](#)

GPDS Milestone None

Supplier None

SW Delivery Date None

Figura 9. Comentarios en sistema para una futura consulta.

Bezel Diagnostics FOM0026149 / FOM0026151 RDF

+

Apps

Description

Add a description...

Validation

None

Attachments 2

Name	Size	Date added		
FOM0026151 RDF.zip	123 KB	13 Aug 2024 12:54pm		
FOM0026149.zip	173 KB	13 Aug 2024 12:54pm		

Completed

✓ Completed

⚡ Actions

Details

Assignee

Joshua Luna

Reporter

Gonzalo Avendano

Labels

Target Program Increment

None

Sprint

[PI 30.5 V710/V363](#)

Vehicle Program

[2026 V363 ICA3 NA](#)

GPDS Milestone

None

Figura 10. Documentación dentro de un request terminado.

3.7 Paso 6

Todos los features asignados se deben correr durante las tres rondas de validación, ya que durante este tiempo se tienen cambios en el software de los módulos, para la última ronda el software que se utiliza es el más acercado al software que tendrá el vehículo en producción, por lo que es de suma importancia depurarlo en cuanto a bugs que pueda contener, después de completadas las tres rondas de validación se da por terminada la vida útil del programa de testeo y toda la información queda registrada en el sistema para una futura referencia.

CAPÍTULO IV

RESULTADOS

4.1 Resultados

Como resultado de las rondas de validación se puede ver que se cumplen con los objetivos planteados al inicio, en cada ronda se trata de encontrar todos aquellos issues que el software pueda contener a fin de poder llegar al producto con la mínima cantidad de errores del sistema de infotainment.

Toda la información generada a lo largo de las tres rondas de validación queda registrada en el sistema, lo cual la vuelve disponible a futuro en caso de ser necesaria una posterior revisión. En el sistema quedan registrados los módulos que se utilizaron con sus números de parte, versión de software que se utilizaron durante las rondas de validación, issues encontrados y cuál fue el estatus final (corrección del bug, falta de implementación, cancelación de la funcionalidad, etc), así como información relevante para el programa si es que durante las rondas hubo algún cambio en módulos o se implementó una nueva funcionalidad que no estaba contemplada, si se incluyó un nuevo feature etc.

Con las rondas de validación se puede ver cómo va evolucionando el programa ya que, al inicio de la vida del programa, en la primera ronda el tester no cuenta con todos los módulos disponibles y no se alcanzan a probar todas las funcionalidades, o en algunos casos cuando se encuentran issues como que una pantalla no se muestra correctamente o aparece de forma incorrecta más que un issue puede ser un tema de implementación ya que al ser una etapa temprana el software no está completamente terminado.

El área de infotainment es una parte crítica y fundamental del vehículo ya que le proporciona al usuario toda la información relacionada con el vehículo, ya sea a través de mensajes transitorios, imágenes, sonidos etc, lo cual hace del testing una parte fundamental para la detección de errores en el producto final.

CONCLUSIONES

Los conocimientos adquiridos a lo largo de la carrera se ven implementados en la industria automotriz, podemos ver los diferentes protocolos de información implementados en un vehículo, podemos ver que dependiendo de los módulos y sus especificaciones son las funcionalidades que se pueden esperar, además de que el conocimiento técnico nos ayuda a poder implementar estrategias para la creación de los harnesses de necesarios de comunicación, con su cables de alimentación, de comunicación CAN BUS o LIN.

Como se puede ver al final de la vida de testing de un programa, el software está listo para los vehículos de producción, como se mencionó en los objetivos del programa con las metodologías de testing aplicadas se contribuye a desarrollar un software seguro.

El área de infotainment es importante y critica para la comunicación del usuario final con el vehículo, el área de testing es fundamental ya que ayuda a desarrollar el software, con cada feature que se valida se garantiza que en el vehículo final se va a tener la mínima cantidad de errores. Según las metodologías del ISTQB no es posible eliminar el 100% de defectos de un software, pero si contribuir a que sea la mínima cantidad. Además, establecen que la detección temprana de issues en los softwares representa un ahorro en tiempo y dinero, ya que mientras más tarde se detectan más recursos se invierten para su corrección además de que pueden retrasar el lanzamiento de un vehículo o que salga al mercado con dichos errores.

OBJETIVOS EDUCACIONALES

- AE1 Identificar, resolver y formular problemas de Ingeniería Mecatrónica aplicando conocimientos de ciencias básicas e ingeniería; Los conocimientos obtenidos durante la carrera de Mecatrónica sirven como base para poder interpretar los protocolos de conocimientos, interpretar los documentos de información y poder ejecutar las actividades de una forma adecuada.
- AE3 Desarrollar y dirigir experimentación adecuada; analizar e interpretar los datos obtenidos y utilizar el juicio ingenieril para establecer conclusiones; En ocasiones el análisis de los test cases ayudan a identificar si se trata de un issue de software o de un test case mal implementado, para ello es necesario utilizar el juicio en base a la ingeniería y el análisis crítico.
- AE4 Comunicarse de manera efectiva con diferentes audiencias, de forma oral y escrita; El área de testing implica estar en constante contacto con diferentes áreas, para validar que se tiene el ultimo nivel de software, que la información está actualizada, etc. Siempre manteniendo el profesionalismo.
- AE5 Reconocer sus responsabilidades éticas y profesionales en situaciones relevantes para la Ingeniería y realizar juicios informados que deben considerar el impacto de las soluciones de Ingeniería Mecatrónica en los contextos global, económico, ambiental y social; Se reconoce que las actividades realizadas tendrán un impacto futuro en el vehículo final, por lo que se cumplen con los más altos estándares de calidad y de profesionalismo.
- AE7 Trabajar de forma eficaz y eficiente en equipos que establecen metas, planean tareas, cumplen fechas límite y analizan riesgos e incertidumbre; Mediante el cumplimiento de las fechas establecidas en las rondas de validación se garantiza que el lanzamiento final del vehículo se cumpla.
- AE8 Innovar productos, procesos, equipos y sistemas mecatrónicas integrando tecnologías emergentes o de vanguardia; Se innova mejorando los test cases y registrando los conocimientos obtenidos de manera empírica para facilitar la ejecución de los mismos.

FUENTES DE INFORMACIÓN

CANalyzer. (s/f). Vector Informatik GmbH. Recuperado el 5 de agosto de 2024, de <https://www.vector.com/us/en/products/products-a-z/software/canalyzer/>

De, E. (2014, mayo 6). *Software*. Enciclopedia Significados. <https://www.significados.com/software/>

Infoentretenimiento en el automóvil. (2022, abril 1). Continental AG. <https://www.continental.com/es-mx/products-innovation/innovation/infoentretenimiento-en-el-automovil/>

Descripción general de los protocolos. Unam.mx:8080. Recuperado el 5 de agosto de 2024, de <http://www.ptolomeo.unam.mx:8080/xmlui/bitstream/handle/132.248.52.100/734/A6.pdf?sequence=6>

Mecanica Automotriz, I. Y. (2021, septiembre 22). *¿Qué es el BUS Lin y cómo funciona?* INGENIERÍA Y MECÁNICA AUTOMOTRIZ. <https://www.ingenieriaymecanicaautomotriz.com/que-es-el-bus-lin-y-como-funciona/>

Northware. (2022, mayo 26). *Requerimientos en el desarrollo de software y aplicaciones*. Northware. <https://www.northware.mx/blog/requerimientos-en-el-desarrollo-de-software-y-aplicaciones/>

¿Qué es Can Bus (red de área de controlador)? (s/f). Soluciones de Adquisición de Datos (DAQ). Recuperado el 5 de agosto de 2024, de <https://dewesoft.com/es/blog/que-es-el-bus-can>

Reichart, G. (2024, julio 24). *AUTOSAR enabling continuous improvement*. Autosar.org. <https://www.autosar.org/>

What we do. (s/f). ISTQB Not-for-Profit Association. Recuperado el 5 de agosto de 2024, de <https://www.istqb.org/about-us/what-we-do>

What is AUTOSAR? - MATLAB & Simulink - MathWorks España. (n.d.).
Mathworks.com. Retrieved August 14, 2024, from
<https://es.mathworks.com/help/autosar/ug/what-is-autosar.html>

ANEXOS