

Trabajo Fin de Máster
Máster en Ingeniería Industrial

Desarrollo de la unidad central de un sistema de sellado electrónico.

Autor: Francisco Javier Rodríguez Atienza

Tutor: Daniel Gutiérrez Reina

**Dpto. de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2022



Trabajo Fin de Máster
Máster en Ingeniería Industrial

Desarrollo de la unidad central de un sistema de sellado electrónico.

Autor:

Francisco Javier Rodríguez Atienza

Tutor:

Daniel Gutiérrez Reina

Profesor Titular

Dpto. de Ingeniería Electrónica
Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2022

Proyecto Fin de Carrera: Desarrollo de la unidad central de un sistema de sellado electrónico.

Autor: Francisco Javier Rodríguez Atienza

Tutor: Daniel Gutiérrez Reina

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2022

El Secretario del Tribunal

A mis familiares

A mis amigos

Agradecimientos

Tras dos nuevos años de esfuerzo y sacrificio, toca poner el broche final a la formación que llevo realizando desde hace ya 7 años, agradeciendo de corazón a todos aquellos que han tenido la paciencia de acompañarme, apoyarme y motivarme para llegar hasta aquí.

Gracias a todas aquellas personas que han aportado su granito de arena para ayudarme con este trabajo dentro de las oficinas de Montrel S.A., haciendo mención especial a Pablo, por permitirme llevar a cabo este proyecto, dándome toda la información posible y prestándose siempre a resolver mis dudas, y gracias también a Manuel, por todas esas veces que he ido a su mesa frustrado porque no daba con la tecla de algún código y se ha parado a verlo conmigo.

Gracias a todos los profesores que han formado parte de esta etapa por enseñarme todos los conocimientos que me han permitido descubrir el inmenso e increíble mundo de la ingeniería.

Especial agradecimiento para mis padres, por haberme educado como lo han hecho, por darme unos valores fundamentales para crecer como persona y por seguir dándome todas las facilidades necesarias para formarme enseñándome a ser la persona que soy.

También quiero destacar la figura de mis amigos, esos que me han estado apoyando desde el primer día, ayudándome a desconectar cuando me hacía falta con todos esos momentos de alegrías y risas. Con ellos he aprendido que rodeado de los tuyos todo es más fácil y divertido.

Por último, me hubiese encantado que mis abuelas hubiesen podido estar conmigo para compartir este momento y que vieses en quien me he convertido. Por todo el amor que me han dado, mi agradecimiento también va para ellas.

A todas y cada una de estas personas os doy las gracias de corazón porque sin vosotros nada de esto hubiera sido posible.

Francisco Javier Rodríguez Atienza

Sevilla, 2022

Resumen

La empresa MONTREL S.A. consta de un Sistema de Sellado Electrónico de camiones cisterna implantado hace ya algo más de quince años. Los equipos que lo componen se solicitan a un proveedor de la empresa para su posterior conexión y montaje para el cliente. Tras el avance de los trabajos llevados a cabo en las instalaciones de la empresa, se contempla el desarrollo y fabricación de sus propios componentes para poder abaratar el coste del sistema así como poder darle un impulso a su uso en la actualidad, pero ¿es viable desde el punto de vista operativo y desde el punto de vista económico?

El presente TFM se centra en el desarrollo de toda la programación lógica que se encuentra detrás de la unidad central haciendo uso de un software web y de los componentes electrónicos necesarios para su puesta en marcha.

La estructura del TFM consta de tres partes. La primera consiste en una introducción a los camiones cisterna, que abarca desde el origen del transporte de los combustibles hasta sus partes principales. La segunda se centra en la explicación del funcionamiento junto con las partes del Sistema de Sellado. Y por último, se ha llevado a cabo la realización del programa, la comprobación de su funcionamiento a través de un cuadro de mandos y la elaboración de unas conclusiones basadas en el trabajo realizado en Node-Red junto con los elementos necesarios para la puesta en marcha.

Abstract

The company MONTREL S.A. has an Electronic Sealing System for tank trucks implemented a little over fifteen years ago. The equipment that composes it is requested from a company supplier for subsequent connection and assembly for the client. After the advances made in the company's facilities, the development and manufacture of its own components is contemplated to reduce the cost of the system as well as to give a boost to its use today, but is it viable from an economic and operational point of view?

This TFM focuses on the development of all the logical programming that is behind the central unit using web software and the necessary components for its implementation.

The structure of the TFM has three parts. The first one consists of an introduction to tank trucks, from the origin of fuel transport to its main parts. The second focuses on the explanation of the operation together with the parts of the Sealing System. And finally, the realization of the program has been carried out, the verification of its operation through a dashboard and the elaboration of some conclusions based on the work carried out in Node-Red together with the necessary elements for the launch.

Índice

Agradecimientos	ix
Resumen	xi
Abstract	xiii
Índice	xv
Índice de Figuras	xvii
Notación	xix
1 Introducción	1
1.1. <i>Objetivos</i>	2
2 El camión cisterna	3
2.1. <i>El transporte de combustible</i>	3
2.1.1. <i>Historia</i>	3
2.1.2. <i>Actualidad</i>	4
2.1.3. <i>Riesgos</i>	5
2.2. <i>Partes</i>	5
2.3. <i>Otros elementos importantes del proceso de distribución</i>	8
3 El Sistema de Sellado	10
3.1. <i>Legislación</i>	11
3.2. <i>Partes</i>	12
3.3. <i>Funcionamiento</i>	18
4 Desarrollo Main Unit	23
4.1. <i>Node-Red</i>	23
4.1.1. <i>Descripción</i>	23
4.1.2. <i>Instalación</i>	24
4.2. <i>Raspberry Pi 4 model B</i>	26
4.2.1. <i>Características</i>	26
4.2.2. <i>Acceso y uso</i>	27
4.3. <i>Programación</i>	29
4.4. <i>Pruebas</i>	41
4.5. <i>Hardware adicional para implementación real</i>	47
5 Conclusiones	49
6 Bibliografía	51
7 ANEXO	53
7.1 <i>Códigos funciones compartimento 1</i>	53
7.2 <i>Diagramas de flujo</i>	59

ÍNDICE DE FIGURAS

Ilustración 1-1. Refinería de petróleo	2
Ilustración 2-1. John Davison Rockefeller	3
Ilustración 2-2. El primer camión cisterna	4
Ilustración 2-3. Cabeza tractora	5
Ilustración 2-4. Acoplamiento API	6
Ilustración 2-5. Válvula de fondo	6
Ilustración 2-6. Boca de hombre	7
Ilustración 2-7. Acoplador recuperación de vapor	7
Ilustración 2-8. Esquema partes principales camión cisterna.	7
Ilustración 2-9. Estación de servicio.	8
Ilustración 2-10. Refinería y centro de almacenamiento y carga de Singapur.	9
Ilustración 3-1. Sistema de sellado instalado en un camión cisterna.	10
Ilustración 3-2. Unidad central con display.	13
Ilustración 3-3. Caja Interfase sensores SPD.	13
Ilustración 3-4. Caja interfase sensores de líquido.	14
Ilustración 3-5. Interfase de salidas digitales y caja de pilotos luminosos.	15
Ilustración 3-6. Caja de separación galvánica.	15
Ilustración 3-7. Sensores en acoplamiento API.	15
Ilustración 3-8. Sensor en boca de hombre.	16
Ilustración 3-9. Sensor de presión neumática y colocación en válvula de fondo.	16
Ilustración 3-10. Protección de sobrellenado Scully.	17
Ilustración 3-11. Conexión general de equipos.	17
Ilustración 3-12. Relación de transiciones de estado.	19
Ilustración 3-13. Diagrama de transición de estados del SECC.	20
Ilustración 3-14. Transiciones en caso de fallo de alimentación.	21
Ilustración 4-1. Ventana principal de Node-Red.	23
Ilustración 4-2. Tipos de nodos.	24
Ilustración 4-3. Página web NodeJS.	25
Ilustración 4-4. Instalador Setup Wizard.	25
Ilustración 4-5. Archivos instalados.	25

Ilustración 4-6. Ejecución de Node-Red.	26
Ilustración 4-7. Raspberry Pi 4 model B.	27
Ilustración 4-8. Ventana inicial del programa PuTTY.	27
Ilustración 4-9. Ventana de control de la Raspberry.	28
Ilustración 4-10. Ventana principal VNC Viewer.	28
Ilustración 4-11. Introducción IP para conexión remota con el escritorio de la Raspberry.	29
Ilustración 4-12. Código para lectura de variable <i>Scully</i> .	30
Ilustración 4-13. Nodos necesarios para la lectura de <i>Scully</i> .	30
Ilustración 4-14. Configuración del contador para la variable <i>Scully</i> .	30
Ilustración 4-15. Propiedades nodo <i>resend</i> .	31
Ilustración 4-16. Nodo de lectura y representación de <i>PN_CirPrin</i> .	31
Ilustración 4-17. Distribución de los nodos para mostrar la fecha y la hora.	31
Ilustración 4-18. Propiedades nodos <i>inject</i> y nodo <i>Date/Time Formatter</i> .	32
Ilustración 4-19. Flujo de la máquina de estados compartimento 1.	32
Ilustración 4-20. Código y propiedades de la función <i>ESTADO FÁBRICA</i> .	33
Ilustración 4-21. Propiedades nodo switch denominado <i>Maq Est 1</i> .	33
Ilustración 4-22. Flujo para registrar una anomalía.	34
Ilustración 4-23. Flujo Guardar ante Fallo.	36
Ilustración 4-24. Flujo para la iluminación de los leds a través del dashboard.	36
Ilustración 4-25. Propiedades del nodo <i>LED Circuito</i> .	37
Ilustración 4-26. Flujo para mostrar en el dashboard el estado de cada compartimento.	37
Ilustración 4-27. Nodos del módulo Raspberry Pi.	38
Ilustración 4-28. Propiedades nodo <i>rpi-gpio in</i> .	39
Ilustración 4-29. Flujo de lectura entradas generales y compartimento 1.	39
Ilustración 4-30. Flujo para la iluminación de los leds a través de la Raspberry.	40
Ilustración 4-31. Display y módulo PCF8574.	40
Ilustración 4-32. Flujo para mostrar en el display los estados.	41
Ilustración 4-33. Dashboard previo al conexionado de la Raspberry.	41
Ilustración 4-34. Dashboard recién conectado el sistema.	41
Ilustración 4-35. Dashboard cuando se tiene los estados COMPROBAD, COMPROBAD y CON RESTO.	42
Ilustración 4-36. Dashboard con compartimentos en estado ACOPLADO, COMPROBAD y PURGADO.	42
Ilustración 4-37. Dashboard mostrando los estados LLENANDO; COMPROBAD Y DISPONIBL.	43
Ilustración 4-38. Dashboard cuando se tienen los estados LLENO, ACOPLADO Y DISPONIBL.	43
Ilustración 4-39. Dashboard cuando los compartimentos están LLENO, DISPONIBL y ACOPLADO.	43
Ilustración 4-40. Dashboard con compartimentos en estado LLENO, DISPONIBL y LLENO.	44
Ilustración 4-41. Dashboard con dos compartimentos en SELLADO y uno DISPONIBL.	44
Ilustración 4-42. Dashboard teniendo los estados DESPRECIN, DISPONIBL y DESPRECIN.	44
Ilustración 4-43. Dashboard tras la finalización de la descarga de dos depósitos.	45
Ilustración 4-44. Dashboard mostrando los estados SIN PURGA, DISPONIBL Y SIN PURGA.	45

Ilustración 4-45. Dashboard tras la primera operación de purga.	45
Ilustración 4-46. Dashboard tras haber purgados los depósitos necesarios.	46
Ilustración 4-47. Conexionado de display a Raspberry con estados iniciados.	46
Ilustración 4-48. Estados mostrados por el display.	46
Ilustración 4-49. Convertidor DC 24V a 5V.	47
Ilustración 4-50. Relé sólido de 4 canales.	47
Ilustración 4-51. Caja antideflagrante con ventana externa	48
Ilustración 4-52. Barrera de aislamiento intrínseco.	48

Notación

l	Litros
km/h	Kilómetros por hora
V	Voltio
W	Wattios
€	Euros
API	American Petroleum Institute
UNE	Una Norma Española
ISO	International Organization for Standardization
CPU	Central Processing Unit
GB	Gigabyte
USB	Universal Serial Bus
HDMI	High-Definition Multimedia Interface

1 INTRODUCCIÓN

Hoy en día nos resulta inconcebible la vida sin gas natural ni petróleo, son dos pilares fundamentales para la vida tal y como la conocemos. Esto se debe a que ambos proporcionan una gran parte de la totalidad del combustible necesario para el transporte terrestre, aéreo y marítimo, además de ser dos de las principales fuentes de energía utilizadas en los hogares para la calefacción, lo que los lleva a ser una fuente vital de ingresos económicos para los países más desarrollados.

Por un lado, el petróleo tiene una infinidad de usos más allá de su uso como combustible, se utiliza en el sector de la industria no solo como fuente de energía sino como lubricante ya que la mayoría de estos se producen de derivados del petróleo, al igual que el amoníaco que se utiliza como fertilizante en la agricultura, el caucho de los neumáticos, el alquitrán de las carreteras o el plástico, que abarca desde la carcasa de una pantalla de ordenador hasta el nylon, siendo este último el plástico más comercializado en la actualidad.

Por otro lado, el gas natural también se encuentra muy asentado en la vida diaria, siendo tanto la principal fuente de energía para la calefacción, sustituyendo al butano, como una de las materias primas de elaboración de plásticos y químicos. Además, la tecnología que está en torno al gas natural está evolucionando y cada vez se está desarrollando más sus usos como por ejemplo para lavavajillas a gas natural, frigoríficos de gas, hornos domésticos de gas e incluso como combustible de vehículos debido a que reduce notablemente el gasto en combustible por su precio inferior al de la gasolina entre otros [1].

Todos los usos recientemente comentados son llevados a cabo con las materias en estado óptimo para su consumo, sea del tipo que sea, pero para llegar a dicho estado tanto el petróleo como el gas natural han de ser extraídos siguiendo su propio ciclo de tratado.

El ciclo del petróleo consiste en una serie de etapas que van desde su extracción a su distribución para consumo pasando por el proceso de procesado, refinado y transporte. La primera es la de extracción, fase en la que se obtiene el crudo de los pozos previamente perforados, a través de las tuberías de producción o “tubing” que lo conducen hasta la superficie. Posteriormente a la extracción se procede al refinado del crudo ya que en este estado natural no es de utilidad y debe ser convertido a productos deseables tales como gasolina o aceites combustibles entre otros. Esta fase suele realizarse relativamente cerca de los lugares de consumo ya que los pozos petrolíferos suelen encontrarse en zonas muy remotas de forma que, previo al refinado, está la fase de transporte de crudo que suele realizarse mediante buques petroleros u oleoductos. Una vez que el crudo ya se encuentra en aquellos lugares en los que se ha demandado, se procede a la fase de refinado anteriormente comentada. Tras tener el crudo tratado en la refinería, los distintos componentes y materias primas obtenidas tales como naftas, queroseno, gasóleos o lubricantes, se distribuyen y comercializan mediante lo que se considera como distribución secundaria hasta las industrias que lo necesiten, los consumidores finales o las estaciones de servicio que lo provean.

Todas estas etapas recién comentadas se engloban bajo el nombre de tres fases que se usan dentro del sector petrolífero y son las fases de:

- Upstream, que hace referencia a la exploración, extracción del crudo y producción.
- Midstream, que se refiere al proceso de transporte de crudo y su almacenamiento.
- Downstream, que engloba al refino y a la venta y distribución de las materias primas obtenidas [2].

A lo largo del desarrollo de este Trabajo de Fin de Máster, nos vamos a centrar única y exclusivamente en la distribución secundaria, es decir, en la distribución de las materias primas obtenidas de las refinerías que denominamos combustibles, concretamente en su transporte por carretera a través de camiones cisterna.

Es en dichos camiones cisterna donde se implementa el sistema de sellado que se va a tratar a lo largo de todo este trabajo. La motivación principal de este trabajo es la de desarrollar bajo las oficinas de MONTREL S.A. una unidad central que controle todo ese sistema abaratando los costes y permitiendo adaptarlos a sus funciones concretas, de manera que no se haga uso de una unidad con un alcance mayor al desempeñado en este sistema y pudiendo tener así una implementación más moderna. Además, con este avance se pretende impulsar de nuevo este sistema.

1.1 Objetivos

El Sistema de Sellado para precintado electrónico de camiones cisterna es un proyecto desarrollado y llevado a cabo por la empresa de ingeniería eléctrica MONTREL S.A. desde el año 2006, y es ahora cuando se pretende dar un paso más acerca de su desarrollo. Este proyecto se realiza para garantizar unas condiciones óptimas de seguridad de cara a los procesos de carga, descarga y transporte para asegurar la plena fiabilidad y calidad del producto en cuestión.

En este documento se va a llevar a cabo el desarrollo de un componente principal del Sistema de Sellado, la Main Unit, con la finalidad de poder fabricarse dentro de las propias oficinas de MONTREL S.A. bajo la supervisión y revisión de técnicos e ingenieros de la empresa. De esta forma, el principal objetivo va a ser el conseguir desarrollar la programación que hay bajo dicha unidad principal y adaptarla para hacerla funcionar en una Raspberry Pi 4, que será el hardware del nuevo sistema de sellado. Para realizar este desarrollo se van a diferenciar varias partes, por un lado, se va a realizar una introducción al mundo del transporte de los combustibles por carretera, entrando más en profundidad en los camiones cisterna. Posteriormente se van a explicar todos los componentes y elementos básicos para poder entender y comprender como funciona y en qué consiste dicho Sistema de Sellado. Por último, se realizará el desarrollo de la CPU principal de cara a su fabricación por parte de la empresa MONTREL S.A. evitando así tener que solicitarla a otra empresa pudiendo hacer el proyecto aún más económico.



Ilustración 1-1. Refinería de petróleo [3].

2 EL CAMIÓN CISTERNA

El camión cisterna se define como una variedad de camión que se utiliza para el transporte de multitud de tipos de líquidos, tales como agua, leche, vino, combustibles, algunos productos químicos, e incluso para el mantenimiento de estos durante un largo período de tiempo, en función de las características que presente. También se pueden utilizar para transportar gases licuados ya que ocupan menor volumen de esta manera, facilitando su transporte y aumentando la cantidad que puede ser transportada, siendo necesario aumentar la presión interior.

Siendo conscientes del gran abanico de posibilidades de transporte que se pueden realizar en un camión cisterna, nos centraremos en el transporte de combustible, mayormente de petróleo, haciendo un breve recorrido histórico por cómo se ha desarrollado a lo largo de los años, su gran trascendencia en el presente y por último cómo es el camión cisterna en la actualidad, las partes que lo componen y lo que engloba a la distribución secundaria.

2.1 El transporte de combustibles

2.1.1 Historia

Aunque era conocido desde la antigüedad, el petróleo no había despertado ningún interés como forma de obtención de energía durante siglos hasta que apareció una lámpara que conseguía iluminar quemando queroseno, un derivado del petróleo, aportando mayor luminosidad a un precio razonable. Sin embargo, había un gran pero en torno a este combustible, y era su complicada obtención. Fue en el año 1859 cuando se garantizó el primer suministro estable de crudo de la mano del aventurero Edwin L. Drake y aquello caló tanto en la sociedad que de la noche a la mañana brotaron cientos de torres petrolíferas en el área llamada Oil Creek. El crudo que se obtenía de dichos pozos se guardaba y distribuía en barriles reutilizados de whisky hasta que se empezó a canalizar directamente mediante tubería anexas a las vías férreas dando lugar a los oleoductos. El auténtico negocio que surgió de todo este avance no fue el de obtener el crudo, sino el de refinarlo y posteriormente vender las materias primas y combustibles obtenidos tales como el queroseno. Aprovechando esto surge la figura de John Davison Rockefeller, que se convirtió en el máximo organizador del sector petrolero al frente de Standard Oil, empresa que desde 1870 compró, procesó y vendió la mayoría del crudo producido en Estados Unidos.



Ilustración 2-1. John Davison Rockefeller [3].

Inicialmente la distribución de petróleo se realizó en latas o bidones transportados en camiones de carga, posteriormente se empezó a distribuir en tanques más grandes tirados por caballos y fue ya en 1910 cuando Standard Oil comenzó a utilizar camiones cisterna a motor en EE. UU., avance que llevaría a Londres, de la mano de Anglo American Oil introduciendo tanques subterráneos y camiones cisterna de entrega. Las áreas rurales continuaron haciendo uso de los caballos, pero alrededor de 1920 el transporte era plenamente a través de camiones cisterna. En la ilustración 2-2 se puede observar uno de los primeros camiones cisterna de los que se tiene registro con capacidad de 780 galones, unos 3000 litros aproximadamente. Estos tanques eran originalmente redondos y muy rara vez rectangulares, de hecho, no fue hasta la llegada de la primera guerra mundial cuando se empezaron a desarrollar los tanques de forma elíptica o cilíndrica que hoy en día conocemos. Al finalizar la guerra, todos los tanques tanto de camiones cisterna como de los vagones cisterna que transportaban combustible por ferrocarril, se reconfiguraron a una de las dos formas. Varios años después fue cuando la tecnología que rodeaba a los camiones cisterna se desarrolló dando lugar a los remolques completos, los semirremolques y los remolques de tractor.



Ilustración 2-2. El primer camión cisterna [4].

2.1.2 Actualidad

Según un estudio realizado por Epdata, desde el inicio del año 2019 hasta el mes de agosto de ese mismo año, en España se consumieron más de 20 000 kilotoneladas de combustible, incluyendo gasóleo y gasolina. Estos datos ayudan a hacernos una idea acerca de la cantidad de camiones cisterna que a lo largo del año circulan por las carreteras ya no solo españolas sino mundiales abasteciendo a las diferentes industrias y estaciones de servicio [5].

Existen varios tipos diferentes de camiones cisterna actualmente, tienden a ser grandes, aunque también los hay de tamaño más reducido, pueden estar presurizados o no, pueden llevar un único tanque o llevarlo dividido en varios compartimentos e incluso pueden ser semirremolques o poseer una estructura recta. El rango de capacidad actual para los camiones cisterna se encuentra comprendido entre unos 5 000 litros para los más pequeños y unos 45 000 litros de capacidad para los más grandes. Además, algunos de estos camiones tienen la capacidad de transportar múltiples productos a la vez al tener el tanque compartimentado en varias partes, normalmente desde 2 hasta 6 partes. La división del tanque ha ido adquiriendo cada vez más importancia debido a dos factores importantes:

- Estabilidad: Los diferentes compartimentos a lo largo del tanque aseguran que a la hora de conducir sobre tramos con pendiente no se sobrecarguen los ejes del camión.
- Eficiencia: Permiten la posibilidad de transportar más de un combustible a la vez.

2.1.3 Riesgos

El transporte de combustibles lleva una serie de riesgos asociados no solo debido a la peligrosidad que tiene el propio transporte de mercancías, sino que además si estas son consideradas peligrosas aumenta considerablemente tanto el índice de siniestralidad como los riesgos potenciales existentes.

Los factores más importantes son los siguientes:

- La carga es el factor que distingue a los camiones cisterna del resto de los vehículos de transporte. Para portar determinadas cargas las cisternas deben pasar controles muy exhaustivos y deben cumplir determinadas normativas en función de la inflamabilidad de la mercancía en cuestión. También hay que controlar los posibles contactos térmicos, mecánicos, eléctrico o cualquier interacción electrostática por débil que sea ya que una inflamación y posible explosión puede tener consecuencias desastrosas.
- La dimensión es otro factor por considerar ya que la envergadura de la cisterna puede influir de cara a extremar la precaución en la conducción, de hecho, cuando dicho tamaño ya es considerable hace falta un carné especial denominado carné ADR que conlleva superar unas pruebas específicas por parte de los conductores de estos vehículos. Las medidas usuales para España, obtenidas de la Organización Internacional para la estandarización, son de unos 3 metros de altura por en torno a 2 metros y medio de ancho y entre 10 y 12 metros de largo. El peso también influye ya que suele ir desde las 15 toneladas en vacío hasta las 30 toneladas cargado. Teniendo en cuenta estos aspectos, una colisión contra cualquier vehículo podría resultar desastroso y por ello aumenta el nivel de precaución a tener.
- La velocidad a la que pueden circular estos vehículos puede alcanzar más de 100 km/h y no suelen tener limitador de velocidad por lo que la velocidad a la que se circule queda bajo la responsabilidad del conductor. Teniendo en cuenta no solo la velocidad sino el peso que está transportando, cualquier impacto a esa velocidad puede ser muy desafortunado.
- Otros factores de riesgo a considerar pueden ser los golpes o las caídas de personas a diferente altura, los contactos con elementos móviles del camión, las vibraciones, ruidos e incluso una posible intoxicación debido a la inhalación de gases tóxicos producidos por la carga [6].

2.2 Partes

Para diferenciar las partes de un camión cisterna, se descompone este en sus dos componentes principales, por un lado, está la cabeza tractora y por otro, la propia cisterna.

La cabeza tractora, que se muestra en la ilustración 2-3, es la parte de un camión que se conduce, dónde se encuentra el motor y que se utiliza para remolcar la carga, en este caso la cisterna. También incluye la cabina donde está el conductor del camión, suele incluir un compartimento para dormir y en ella se encuentran tanto los depósitos de combustible como el deflector de viento que actúa como una visera para ayuda a reducir la resistencia del viento en carretera. El último elemento importante de una cabeza tractora es la quinta rueda, se denomina así al mecanismo de seguridad que permite acoplar el semirremolque a la cabeza tractora encargándose de dejar el espacio suficiente entre ambos elementos de manera que se permita realizar las maniobras necesarias de manera cómoda y segura.



Ilustración 2-3. Cabeza tractora [7].

Por otro lado, la cisterna es el elemento en el que se almacena la mercancía, en este caso el combustible que se va a transportar, normalmente en forma de semirremolque elíptico. A su vez, la cisterna está formada por una serie de elementos indispensables que son los siguientes:

- **Compartimentos:** se denomina compartimento a cada una de las subdivisiones que tiene el depósito de la cisterna en las cuales se deposita el combustible. Pueden cargarse con productos diferentes ya que están aislados unos de otros, suelen haber entre 2 y 6 compartimentos en una cisterna, siendo lo más frecuente que hayan 5, y suelen tener una capacidad que va desde los 3000 litros hasta los más de 12000 litros.
- **Bocas de Carga y descarga:** También se denominan acoplamientos seco API y se utilizan para proporcionar una carga y una descarga segura y limpia, sin goteo, de camiones cisterna. Están diseñados para la conexión de mangueras de manera fácil y rápida permitiendo que se pueda operar con ellas de manera sencilla, evitando pérdidas y siendo necesario que la unidad esté totalmente acoplada para que se abra la válvula de seguridad y pueda hacerse la carga o la descarga. Hay una por compartimento.



Ilustración 2-4. Acoplamiento API [8].

- **Válvula de fondo:** Las válvulas de fondo constituyen un sistema de seguridad que permiten que se realicen con total seguridad tanto el transporte como la carga y la descarga. Su objetivo es gestionar la apertura de la cisterna regulando el paso de combustible y se encuentra situada en la parte baja de cada uno de los compartimentos.



Ilustración 2-5. Válvula de fondo [9].

- **Colector y válvulas de recuperación:** Los colectores son los conductos que sirven de unión entre la válvula de fondo y los acoplamientos API facilitando el flujo de combustible y dónde se pueden quedar restos de este. Junto con las válvulas de recuperación su misión es evitar que salgan a la atmósfera los gases que emite el combustible durante las cargas y descargas.
- **Boca de hombre:** se denomina boca de hombre a la abertura que se encuentra en la parte superior de un depósito y que permite tanto acceder a una persona al interior de cada compartimento como realizar las inspecciones necesarias del material que este contiene. Anteriormente se utilizaban también para la carga de mercancía, pero por cuestiones de seguridad se dejó de realizar haciéndose por las bocas de carga y descarga antes descritas. Hay una sobre cada compartimento.



Ilustración 2-6. Boca de hombre [10].

- Sistema de presión neumático: es el mecanismo que garantiza que las válvulas se abran y cierren por completo en todo momento consiguiendo mayor seguridad y control minimizando errores de operarios.
- Acoplador de recuperación de vapor: Este sistema se encarga de permitir la evacuación de gases durante el momento de la carga. Antes de cargar debe conectarse al camión un brazo de gases a una boca especial denominada Racor, que se encarga de recogerlos y mediante un circuito cerrado se llevan al tanque dónde están saliendo, o bien se llevan a una Unidad de Recuperación de Vapores (URV) para tratarlos.



Ilustración 2-7. Acoplador recuperación de vapor.

A continuación, en la ilustración 2-7 se muestra un esquema general de los elementos principales que constituyen un semirremolque cisterna.

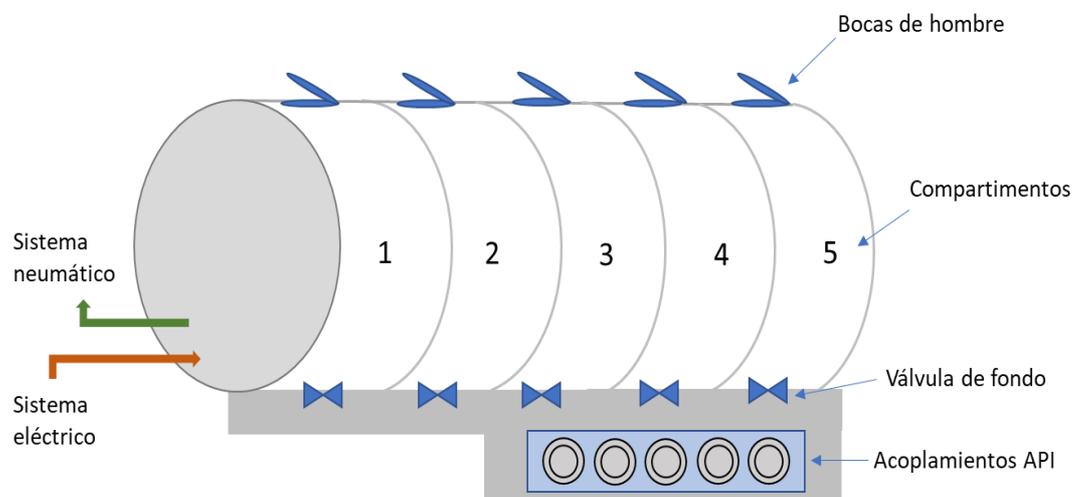


Ilustración 2-8. Esquema partes principales camión cisterna.

2.3 Otros elementos importantes del proceso de distribución

Tal y como se ha comentado anteriormente, el hecho de centrarnos en el camión cisterna es debido a que es el encargado mayoritario de distribuir los combustibles, una vez refinados, en el proceso de distribución secundaria. En todo este proceso no sólo encontramos a este tipo de vehículos, sino que se pueden diferenciar otras figuras relevantes en este ámbito, que son las mostradas a continuación:

- Las estaciones de servicio. De todos los elementos es el que más presente está en la vida cotidiana ya que para todo aquel que tenga coche, el acercarse a una de ellas a llenar el depósito le sonará corriente. Cada vez más están incorporando tanto servicios de compra para clientes como zonas de descanso, restauración o los túneles de lavado. De cara al almacenaje y distribución de combustible, que es lo que aquí interesa, en el subsuelo de las estaciones es dónde se encuentran los tanques de almacenado de dicho combustible que se recibe a través de los camiones cisterna. El suministro a los clientes que la estación recibe se realiza mediante los surtidores, que son los dispositivos que controlan tanto el tipo de combustible que se desea, como la cantidad requerida, mediante una bomba que lo aspira desde el tanque hasta el depósito del vehículo.

Hoy en día ya están muy automatizadas con un sistema de medición automática de tanques que está constantemente midiendo tanto la cantidad de combustible que hay en cada uno como su temperatura. Gracias a esto, cuando el sistema detecta que hay menos producto del mínimo para garantizar un consumo estándar, se emiten de manera automáticas las órdenes que solicitan el reabastecimiento de combustible a los centros encargados de su distribución.



Ilustración 2-9. Estación de servicio [11].

- Operadores. Son las empresas petroleras que compran el producto y lo comercializan a las industrias, estaciones de servicio o consumidores finales. Estos serían los mayores interesados en el sistema de sellado ya que tienen que garantizar que el producto que sale de los centros de carga y que el cliente solicita, es el mismo que el que recibe tanto en calidad como en cantidad cuando llegan a su destino los camiones cisterna.
- Empresas transportistas. Son las compañías contratadas por los operadores para hacer el transporte de combustible a todos los clientes que lo hayan adquirido. Suelen contar con una gran flota de camiones cisterna con los que realizar los diferentes aprovisionamientos. Estas empresas reciben la mayoría de la responsabilidad de todo el proceso de distribución ya que son los responsables del producto desde que sale del centro de almacenamiento y carga hasta que se realiza la descarga del combustible en la estación de servicio correspondiente. Se encargan también de planificar las rutas de distribución y de minimizar los riesgos de este transporte de cara a la población.

- Centros de almacenamiento y carga. Son los centros encargados de almacenar los combustibles ya refinados, recibidos vía marítima a través de buques petroleros o a través de oleoductos, en tanques de almacenaje de una capacidad en torno a los 50.000 metros cúbicos. En estos centros es en dónde se producen las operaciones de carga, concretamente en la zona de cargadero, y dónde se controla la calidad de los diferentes combustibles [6].



Ilustración 2-10. Refinería y centro de almacenamiento y carga de Singapur [12].

3 EL SISTEMA DE SELLADO

El Sistema de Sellado, también conocido como Sistema de Sellado Electrónico de Camiones Cisterna o SECC, es un proyecto de la empresa MONTREL S.A., que surge tras la necesidad de realizar un avance tecnológico del sistema de precintado electrónico que tiene incorporada la flota de camiones cisterna de CEPESA desde el año 1999, y que es debido a la revisión realizada al Acuerdo Europeo de Transporte Internacional de Mercancías Peligrosas por Carretera (ADR 2005).

La finalidad de realizar este nuevo sistema de sellado es la de permitir al destinatario o a la autoridad competente en la vía pública, comprobar que la estanqueidad de la cisterna no ha sido afectada por ninguna manipulación entre el remitente y el destinatario, además de poder controlar los procesos de carga y descarga de los depósitos, presentando en cada momento información del estado en el que se encuentra cada uno de los compartimentos, pudiendo guardar la información de los registros de cualquier situación que se haya producido en la cisterna. Igualmente, el nuevo sistema de sellado también incorpora la posibilidad de volcar información a los ordenadores para su posterior tratamiento, localización y envío en tiempo real de las incidencias que se van produciendo y la posibilidad de impedir la apertura de las válvulas de fondo fuera de los lugares geográficos previamente autorizados para las cargas y descargas.

Una vez realizadas todas estas mejoras y tras ver que el avance ha sido positivo, obteniendo muy buenos resultados de este sistema de sellado, es cuando Montrel pretende dar un paso más tratando de desarrollar en su propia planta uno de los componentes más importantes de todo el sistema, la unidad central, que es la interfase hombre-máquina integrando en una sola unidad todas las interfases que se comentarán a lo largo de este capítulo, de manera que se simplifique el hardware que hay entorno al sistema de sellado.



Ilustración 3-1. Sistema de sellado instalado en un camión cisterna [13].

3.1 Legislación

Para la redacción del proyecto del SECC se tomaron como referencia una serie de leyes, normas, reglamentos, decretos y códigos que hoy en día siguen vigentes tras haber sido actualizados y renovados a lo largo de estos años. Cada uno de ellos proviene de una organización, normativa o acuerdo diferentes los cuales se irán explicando brevemente a la vez que se enumeran dichas leyes y normas.

- ADR: es el acrónimo de “Acuerdo Europeo para el Transporte Internacional de Mercancías Peligrosas por Carretera”, entendiéndose por mercancías peligrosas a aquellas cuyo posible accidente durante su transporte pueda suponer riesgo tanto para la población como para el medioambiente y por ello las condiciones para su transporte están reguladas de manera específica. En España hay un decreto, el Real Decreto 551/2006, que obliga a cumplir el ADR, siendo actualmente el ADR 2021 el que se encuentra vigente y el cuál es de obligado cumplimiento para este proyecto [14].
- Asociación Electrotécnica Española: Es una institución de carácter técnico-científico, fundada en 1930 como una iniciativa de un grupo de profesionales relacionados con la ingeniería eléctrica. Es una asociación sin ánimo de lucro, y sus objetivos son promover el desarrollo de la ingeniería eléctrica en todos sus aspectos y desde su creación, ha sido una entidad que ha participado activamente en la normalización y modernización del sector electrotécnico. De aquí se cumplen las normas y recomendaciones establecidas por la asociación [15].
- Normas UNE-EN ISO: Las normas ISO provienen de la “Organización Internacional de Normalización” y son una serie de estándares con reconocimiento internacional para establecer homogeneidad entre los ámbitos de gestión, desarrollo de productos y prestación de servicios en la industria. Las normas UNE provienen de “Una Norma Española”, único organismo de normalización en España, que contribuye al desarrollo de la economía a través de la elaboración de normas técnicas y las actividades de cooperación internacional. De forma que, son un conjunto de normas experimentales e informes creados por los comités técnicos de normalización pertenecientes a la Asociación Española de Normalización y Certificación, AENOR. En caso de adopción de normas internacionales o europeas se añade el prefijo UNE a la codificación de esta, de manera que la Norma UNE-EN 123 es una adopción nacional de la norma EN 123. En estos tipos de normas, las que se cumplen son las siguientes:
 - ISO 12098:2020. Vehículos de carretera. Conectores para la conexión eléctrica de vehículos tractores y remolcados. Conector de 15 polos para vehículos con tensión de alimentación nominal de 24 V.
 - ISO 7638-1:2018. Vehículos de carretera. Conectores para la conexión eléctrica de vehículos tractores y remolcados. Conectores para sistemas de frenado y tren de rodaje de vehículos con tensión de alimentación nominal de 24 V.
 - UNE-EN 60079-10-1:2016. Clasificación de emplazamientos en atmósferas explosivas gaseosas.
 - UNE-EN 60079-14:2016. Instalaciones eléctricas en emplazamientos peligrosos.
 - UNE-EN 60079-26:2015. Material con nivel de protección de material eléctrico (EPL) 1 G.
 - UNE-EN 13094:2021. Cisternas para el transporte de mercancías peligrosas. Cisternas metálicas con descarga por gravedad. Diseño y construcción.
 - UNE-EN 14025:2019. Cisternas para el transporte de mercancías peligrosas. Cisternas metálicas sometidas a presión. Diseño y fabricación [16].
- Reales Decretos: son normas jurídicas con rango de reglamento que emanan del gobierno y en virtud de las competencias prescritas en la Constitución. Llevan el termino Real si el rey lo manda como Jefe de Estado, y se trata entonces de una norma que proviene de ministerio. También están las órdenes que son normas de rango reglamentario que emana de cualquiera de los ministros del Gobierno de España. Los decretos y órdenes tomadas como referencia son:
 - Real Decreto 97/2014, de 14 de febrero, por el que se regulan las operaciones de transporte de mercancías peligrosas por carretera en territorio español.

- Real Decreto 70/2019, de 15 de febrero, por el que se modifican el Reglamento de la Ley de Ordenación de los Transportes Terrestres y otras normas reglamentarias en materia de formación de los conductores de los vehículos de transporte por carretera, de documentos de control en relación con los transportes por carretera, de transporte sanitario por carretera, de transporte de mercancías peligrosas y del Comité Nacional del Transporte por Carretera.
- Reglamento Electrotécnico para baja tensión e ITC aprobado en el Real Decreto 842/2002, con fecha de 2 de agosto.
- Orden FOM/606/2018, de 25 de mayo, sobre el contenido del informe anual para el transporte de mercancías peligrosas por carretera.
- Real Decreto 1437/2002, de 27 de diciembre, por el que se adecuan las cisternas de gasolina al Real Decreto 2102/1996, de 20 de septiembre, sobre control de emisiones de compuestos orgánicos volátiles (C.O.V.).
- Real Decreto 1388/2011, de 14 de octubre, por el que se dictan las disposiciones de aplicación de la Directiva 2010/35/UE del Parlamento Europeo y del Consejo de 16 de junio de 2010 sobre equipos a presión transportables.
- Real Decreto 298/2021, de 27 de abril, por el que se modifican diversas normas reglamentarias en materia de seguridad industrial.
- Real Decreto 840/2015, de 21 de septiembre, por el que se aprueban medidas de control de los riesgos inherentes a los accidentes graves en los que intervengan sustancias peligrosas.
- Real Decreto 340/2010, de 19 de marzo, por el que se modifica el Real Decreto 948/2003, de 18 de julio, por el que se establecen las condiciones mínimas que deben reunir las instalaciones de lavado interior o desgasificación y despresurización, así como las de reparación o modificación de cisternas de mercancías peligrosas.
- Ley 31/1995, de 8 de noviembre, de prevención de Riesgos Laborales [17].
- DIN: estas siglas se corresponden al “Deutsches Institut für Normung”, que es el Instituto Alemán de Normalización y es una organización nacional que elabora los estándares técnicos o normas para la racionalización y el aseguramiento de la calidad. En lo que se refiere DIN, de cara a este proyecto son de mucha importancia los acoplamientos DIN 28450 que se encargan de realizar la desconexión en seco para evitar las pérdidas de combustible debido al goteo.
- Por último, también debe cumplir con las normas del IEEEE y con la directiva ATEX. Por un lado, el IEEEE, Instituto de Ingenieros Eléctrico y Electrónico, es una asociación mundial de ingenieros dedicada a la normalización y el desarrollo en áreas técnicas. En este proyecto su contenido abarca tanto el sistema de comunicaciones como la instrumentación y el cableado eléctrico. Por otro lado, la directiva ATEX, cuyas siglas provienen de una directiva elaborada por el gobierno francés “Appareils destinés à être utilisés en Atmosphères Explosives”, describe que tipo de equipamientos y ambientes están permitidos para trabajo en una atmósfera explosiva. Es de esta directiva de donde se obtienen las zonas EX que hacen referencia a las áreas peligrosas debidas a la presencia de sustancias peligrosas ya sea en forma de gas, vapor o niebla clasificándolas en zona 0, 1, 2 entre otras [18].

3.2 Partes

En este apartado se van a enumerar y explicar los diferentes elementos que componen el SECC y que son imprescindibles para su correcto funcionamiento.

Unidad Central

Este componente es el núcleo principal del sistema, donde se encuentra la CPU y es la interfaz entre el proceso y operario, es decir, la interfaz hombre-máquina. Realiza las funciones de control y gestiona las diferentes

entradas/salidas del sistema, comprobando su correcto funcionamiento. Sobre ella se programan ajustes específicos según los diferentes procedimientos de carga o descarga.

Las señales que recibe la CPU provienen de los sensores de apertura de los acopladores API, la apertura de las bocas de hombre, la apertura de las válvulas de fondo de cada compartimento y de la existencia de líquido en cada depósito. Además gestiona también las comunicaciones con el centro de control remoto y puede incorporar una impresora local de cara a una posible impresión de tickets de descarga con el estado de los compartimentos.



Ilustración 3-2. Unidad central con display [19].

Interfase de sensores SPD

Es el elemento que se encarga de monitorizar las aperturas por las que puede ser extraído el combustible y registrarlas. Esto se realiza mediante los sensores que se enumeran a continuación:

- Sensores de detección de la presión neumática en las válvulas de fondo.
- Sensores de proximidad inductivos. Reconocen la apertura de las diferentes bocas de hombre y de los acoplamientos API.

Envía todos estos estados de los sensores a la unidad principal y es capaz de detectar los cambios de estado de los sensores cuando falta alimentación.

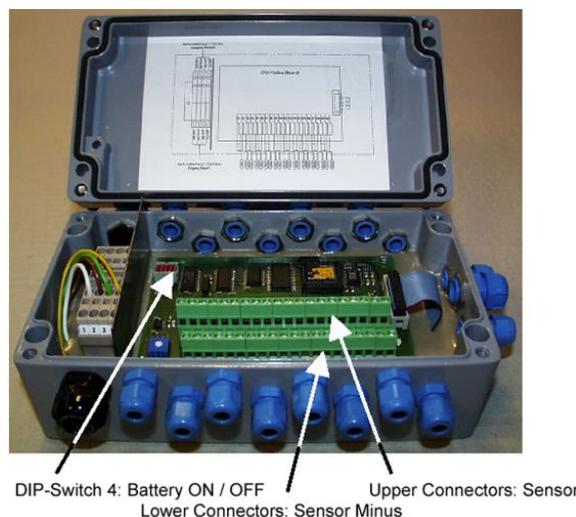


Ilustración 3-3. Caja Interfase sensores SPD [20].

Interfase de sensores de líquido

Esta interfase, ubicada en una caja similar a la de la interfase de sensores SPD, se encarga de detectar la existencia de líquido en todos y cada uno de los compartimentos que tiene la cisterna, mediante los sensores que se encuentran instalados en los acopladores API.

Junto con las respectivas entradas necesarias para estos sensores, se encuentran otras dos entradas de seguridad intrínseca siguientes:

- Una es para la conexión del sensor detector de presión neumática que hay en el circuito principal, de manera que gracias a este sensor se evitan ciertos errores que podrían producirse si no se tuviera la presión adecuada.
- La otra entrada es la encargada de detectar tanto la conexión del sistema de protección de sobrellenado, Scully, como del equipo de puesta a tierra, PAT. Ambas conexiones son imprescindibles ya que indican que el camión se encuentra en una isleta de carga y/o descarga.

Al igual que ocurre con la interfase de sensores SPD, los estados de todas las señales de la interfase de sensores de líquido son enviados a la unidad central para su procesamiento.



Ilustración 3-4. Caja interfase sensores de líquido [20].

Interfase de salidas digitales

Este componente se encarga de activar los pilotos luminosos verdes que indican el estado del sistema según el siguiente código:

- Pilotos apagados: VACÍO.
- Pilotos encendidos: SELLADO.
- Pilotos intermitentes: resto de estados.

También activa el piloto luminoso amarillo para indicar la existencia de presión general del sistema neumático de la cisterna. Estas señales las recibe desde la unidad central con órdenes de activación o desactivación.

Respecto a la cantidad de pilotos, habrá un indicador luminoso amarillo que indica la presión general del sistema, y tantos pilotos luminosos verdes como compartimentos haya en la cisterna. Estos solo se activan cuando el portalón de la caja de válvulas esté abierto.

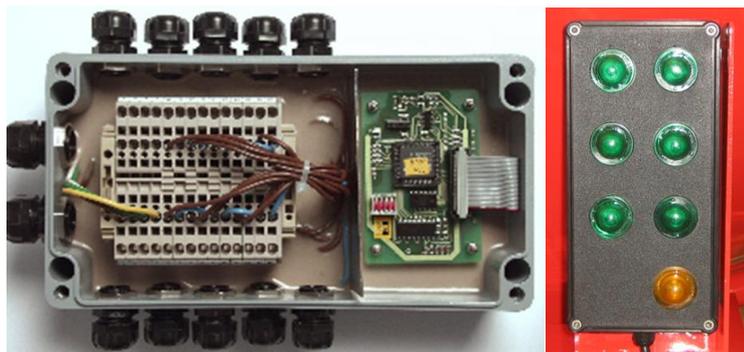


Ilustración 3-5. Interfase de salidas digitales y caja de pilotos luminosos [20].

Todos estos elementos comentados anteriormente son los que se pretenden sustituir por un único sistema que se desarrollará en el capítulo siguiente, el capítulo 4.

Caja de separación galvánica

Mediante esta caja se interconecta la base del sistema de control de sobrellenado que tiene la cisterna con el propio hardware del sistema de sellado para detectar que el camión se encuentra posicionado en un cargadero. Aunque la señal es generada a través de la conexión Scully, la separación galvánica tiene la función de garantizar el aislamiento eléctrico entre los dispositivos del SECC y el equipo del cargadero, evitando así posibles fallos o averías.



Ilustración 3-6. Caja de separación galvánica [13].

Sensores de detección de líquido y apertura acoplamiento API

Ambos sensores forman parte del acoplador API a instalar en cada compartimiento de la cisterna. Estos se roscan en la parte inferior del acoplador, consiguiendo la estanqueidad por medio de juntas tóricas resistentes a los hidrocarburos. Hay un sensor de cada tipo por cada compartimento que haya en la cisterna. En las siguientes ilustraciones se muestra la disposición de ambos sensores en el acoplador:



Ilustración 3-7. Sensores en acoplamiento API [20].

Sensores de boca de hombre

Estos sensores se ubican en la boca de hombre de manera que cuando esta se abre, el pestillo de la ilustración actúa y es de esta forma como la unidad central detecta su apertura. Hay un sensor de boca de hombre por cada compartimento, ya que cada uno tiene su propia boca de acceso.

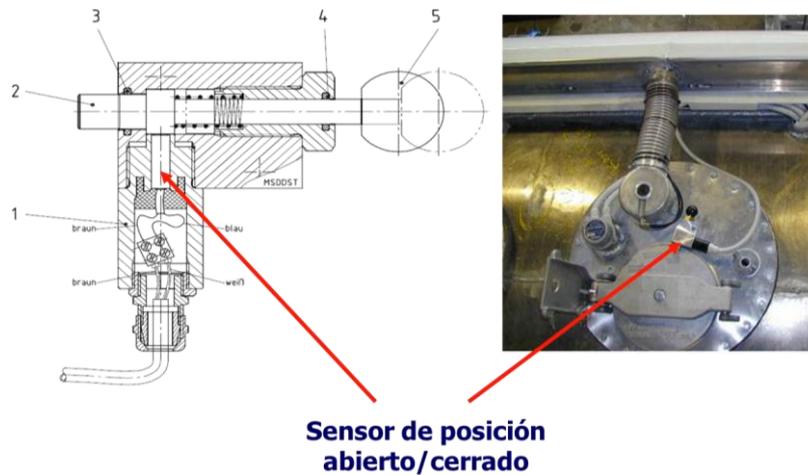


Ilustración 3-8. Sensor en boca de hombre [13].

Sensores de presión neumática

Estos sensores son los encargados de detectar la presencia de aire comprimido de accionamiento tanto en las válvulas de fondo como en el sistema neumático principal. Si no se tiene señal de presión neumática en cualquiera de las válvulas, significa que esa válvula se encuentra abierta, de manera que así es como la unidad principal es capaz de detectar su apertura. Hay un sensor de presión neumática en la válvula de fondo de cada uno de los compartimentos y luego otro para el circuito principal.

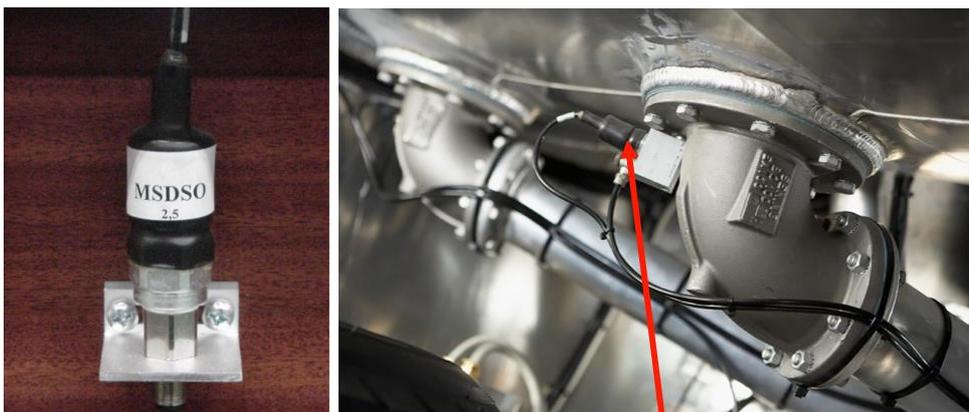


Ilustración 3-9. Sensor de presión neumática y colocación en válvula de fondo [20].

Protección de sobrellenado

Los sistemas de protección contra sobrellenado detectan una señal de nivel de líquido desde un sensor de sobrellenado. El sistema proporciona una señal automática al equipo de automatización del estante para cortar el flujo de líquido y advertir de una condición inminente de desbordamiento de un camión cisterna. Además

debe conectarse al llegar a un cargadero para que la unidad central pueda reconocer que se va a realizar una operación de carga o descarga. Para cada compartimento hace falta un sensor que va conectado a la boca principal de conexión que se muestra en la ilustración 3-10, que es dónde se realiza la conexión al llegar a la isleta de carga. Este sistema también se le denomina Scully debido a su fabricante.



Ilustración 3-10. Protección de sobrellenado Scully [21].

Conexiones

Tal y como se muestra en la ilustración 3-11, existe un cable que recorre la unidad central, la caja interfase de los sensores SPD y la caja de la interfase de los sensores de presencia de líquido por este orden. Por este cable se alimentan las diferentes unidades y se conectan entre sí mediante un bus de comunicaciones, por el que la Unidad Central supervisa el estado de los diferentes sensores. Este cable recibe el nombre de Alimentación / CAN-Bus.

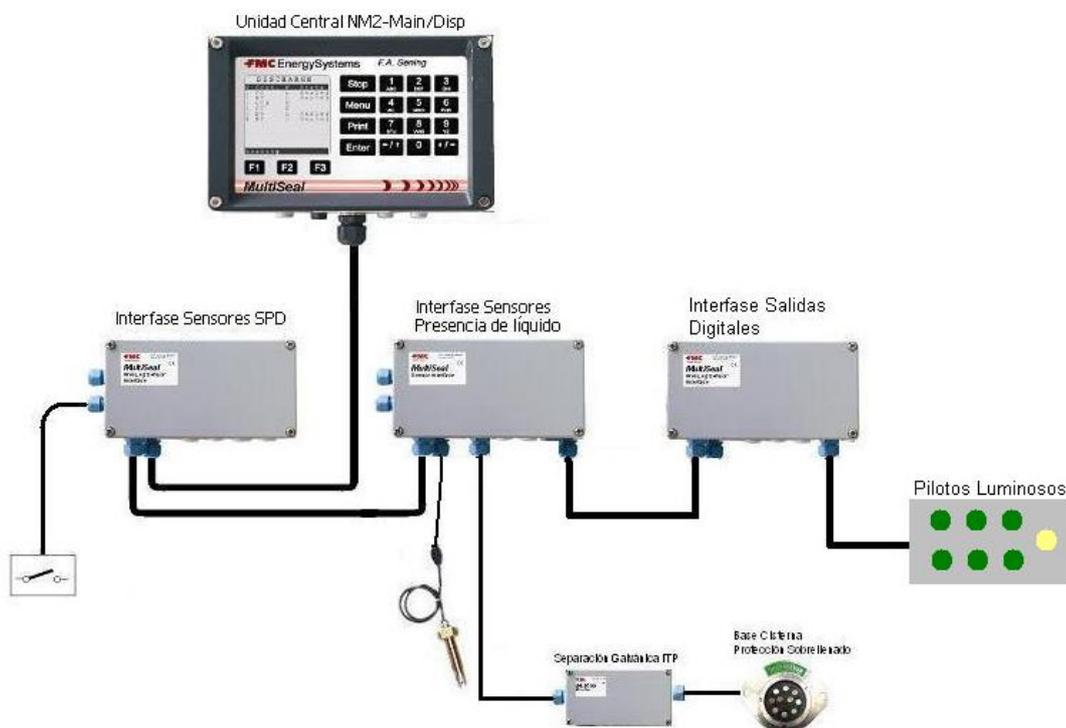


Ilustración 3-11. Conexión general de equipos [20].

Todo el sistema se encuentra alimentado con un cable que cumple con la norma CEI 60079, de Seguridad Aumentada, protegido por un fusible de 8 A o por un desconector automático colocado cerca de la fuente de tensión, siendo la potencia total consumida por el sistema de 130 W. Esta cantidad se desglosa en función de

los diferentes componentes de manera que la unidad central consume 30 W, la interfase de sensores detectores de líquido y la interfase del resto de sensores consumen otros 30 W cada una y la interfase de salidas digitales consume 40 W, siendo la suma total de estas potencias la potencia total anteriormente comentada de 130 W. Además, el sistema se comunica con el exterior a través de una unidad GPS-GPRS que permite el envío en tiempo real de las incidencias y cambios de estado que se vayan produciendo en la cisterna, añadiendo también coordenadas geográficas de la localización del vehículo y la hora real del mismo. Esto permite también, tener la posibilidad de impedir la apertura de las válvulas de fondo fuera de las localizaciones geográficas previamente autorizadas para carga y descarga.

3.3 Funcionamiento

El sistema de sellado tiene como función principal la de controlar los procesos de carga y descarga de combustible además del transporte de este monitorizando el estado de los compartimentos a través del análisis de los múltiples sensores comentados en el apartado anterior. Además, todos los datos se van almacenando en tiempos reales históricos con la finalidad de poder consultar cualquier suceso que se haya producido en el camión independientemente de cuando haya ocurrido. Como el SECC funciona a través de los diferentes estados que se pueden dar en cada uno de los compartimentos, hace la función de una máquina de estado. Una máquina de estado hace referencia a un modelo de comportamiento de un sistema con entradas y salidas en donde las salidas no sólo dependen de las entradas actuales sino también de las que se han dado con anterioridad, de manera que el historial de señales de entrada determina en cada instante un estado para la máquina de forma que la salida depende únicamente del estado y las entradas actuales. Son un modo habitual de representar un conjunto de estados interrelacionados de un proceso.

Los 15 estados posibles que se pueden dar en cada uno de los compartimentos pueden ser:

- Vacío: en la unidad central aparece como VACIO.
- Pulsado: en la unidad central aparece como PULSADO.
- Comprobado: en la unidad central aparece como COMPROBAD.
- Acoplado: en la unidad central aparece como ACOPLADO.
- Disponible: en la unidad central aparece como DISPONIBL.
- Con restos: en la unidad central aparece como CON RESTO.
- Purgado: en la unidad central aparece como PURGADO.
- Llenándose: en la unidad central aparece como LLENANDO.
- Lleno: en la unidad central aparece como LLENO.
- Sellado: en la unidad central aparece como SELLADO.
- Desprecintado: en la unidad central aparece como DESPRECIN.
- Descargado: en la unidad central aparece como DESCARGAD.
- Sin purgar: en la unidad central aparece como SIN PURGA.
- Purgar: en la unidad central aparece como PURGAR.
- Fallo: en la unidad central aparece como FALLO.

Las relaciones de transición que hay entre los estados anteriormente enumerados se muestran en la siguiente ilustración, en la cual se exponen las condiciones necesarias para que se den dichas transiciones y se pase de un estado u otro.

RELACIÓN DE TRANSICIONES DE ESTADOS

- A**-Scully, API cerrado y Válvula de fondo abierta durante 3 segundos.
- B**-Scully, API cerrado, Válvula de fondo cerrada y no hay líquido.
- C**-Scully, API abierto y hay líquido.
- D**-No Scully durante 60 segundos, API abierto o Válvula de fondo abierta.
- E**-Scully, API cerrado, Válvula de fondo cerrada, boca de hombre cerrada y hay líquido.
- F**-No hay Scully durante 5 segundos, API cerrado, Válvula de fondo cerrada, boca de hombre cerrada, hay líquido y no hay ningún otro depósito Con Resto.
- G**- API abierto o Válvula de fondo abierta o boca de hombre abierta, esta condiciones sin Scully.
- H**-No Scully, API abierto, Válvula de fondo abierta y hay no líquido durante 20 segundos.
- I**- Scully y se detecta líquido.
- J**-Scully, API abierto, Válvula de fondo abierta y 5 segundos sin líquido.
- K**-Scully, API cerrado y Válvula de fondo cerrada.
- L**-No hay Scully, API cerrado, Válvula de fondo cerrada y no se detecta líquido.
- M**-Scully, API cerrado y no líquido durante 8 segundos.
- N**- Detección de líquido.
- O**- Scully y API abierto.
- Q**-No líquido y API abierto.
- R**-No líquido durante 30 segundos.
- S**-Algún sensor del compartimento en fallo, en cortocircuito o desconectado.
- T**-Sensores del compartimento en estado correcto.
- U**-No Scully, Válvula de fondo abierta y no hay líquido durante 5 segundos.
- V**-Válvula de fondo abierta y hay líquido durante 5 segundos.
- W**-API abierto, Válvula de fondo abierta y no hay líquido durante 5 segundos.
- X**-Scully, API abierto, hay líquido y existe algún otro depósito Con Resto.

Ilustración 3-12. Relación de transiciones de estado [17].

Tras haber enumerado tanto los estados como las transiciones que se dan entre estos, se puede desarrollar el diagrama de flujos de transición que sigue la lógica del sistema de sellado con la finalidad de entender cómo se dan los diferentes saltos de estado y que transiciones siguen. Dicho diagrama es el que se muestra en la siguiente ilustración 3-13.

Los diferentes colores que aparecen en dicha ilustración se deben a que en verde se consideran las transiciones que entran dentro del normal funcionamiento del sistema y, por otro lado, en morado se encuentran aquellas transiciones que se producen debido a hechos no deseados, es por ello por lo que reciben el nombre de transiciones anómalas.

En el caso de que se produzca el fallo de alimentación, las transiciones posibles vienen representadas en un diagrama aparte ya que es un hecho muy concreto. En la ilustración 3-14 se muestra tanto la representación de la transición de estados en este caso, como la relación de estos y por qué se produce uno u otro.

DIAGRAMA DE TRANSICION DE ESTADOS SELLADO DE CISTERNAS

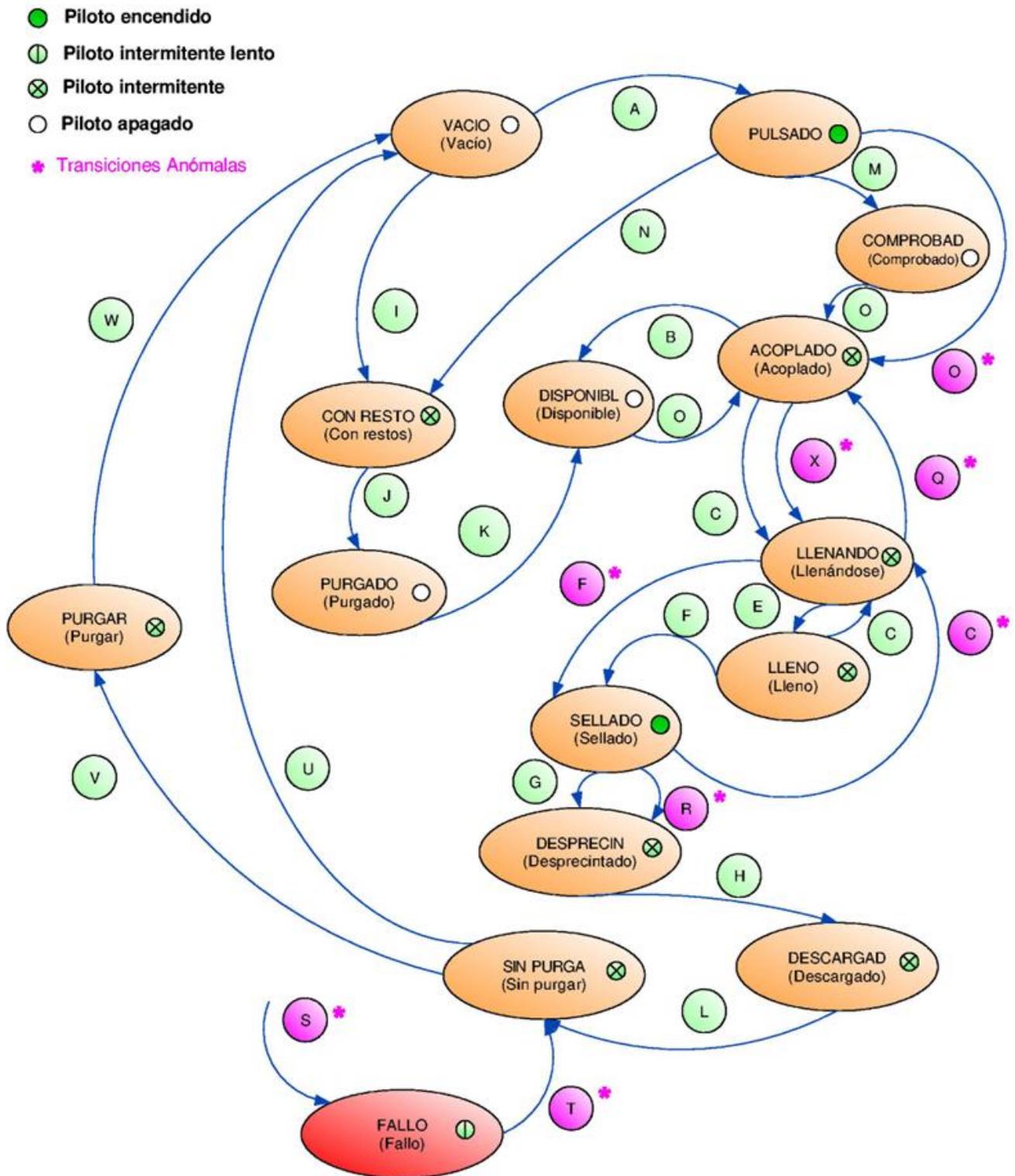
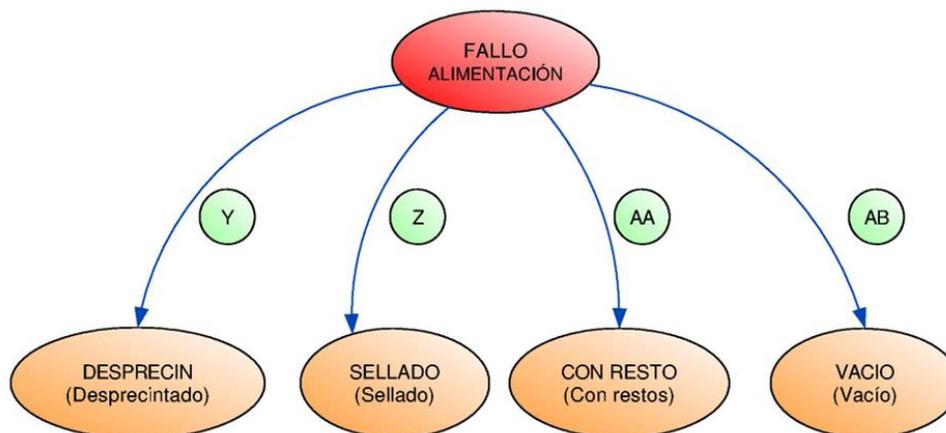


Ilustración 3-13. Diagrama de transición de estados del SECC [17].

DIAGRAMA DE TRANSICION DE ESTADOS TRAS FALLO ALIMENTACIÓN**RELACIÓN DE TRANSICIONES DE ESTADOS TRAS FALLO ALIMENTACIÓN**

Y-Partiendo del estado de Sellado, durante el período de tiempo sin alimentación, se ha manipulado el acoplador API, o la válvula de fondo, o la boca de hombre; o el depósito ya estaba desprecintado.

Z-Partiendo del estado de Sellado, durante el período de tiempo sin alimentación, **no** se ha manipulado el acoplador API, ni la válvula de fondo, ni la boca de hombre.

AA-Inicio del depósito tras fallo de alimentación, si se detecta líquido, y el depósito no estaba Sellado ni Desprecintado.

AB-Inicio del depósito tras fallo de alimentación, si **no** se detecta líquido, y el depósito no estaba Sellado ni Desprecintado.

Ilustración 3-14. Transiciones en caso de fallo de alimentación [17].

Una vez se han descrito todos los estados y sus transiciones se va a explicar cómo es el funcionamiento concreto que se da en los dos procesos principales en los que se ve involucrado el sistema de sellado, el proceso de carga por un lado y el proceso de descarga por otro.

Proceso de carga

Si el desarrollo de este proceso sigue un trascurso natural y sin anomalías, la secuencia de estados que se da es:

VACIO-PULSADO-COMPROBAD-ACOPLADO-LLENANDO-LLENO

El estado VACIO se da una vez se ha conectado el Scully a la cisterna la unidad central ha comprobado el estado de los depósitos indicándolo con los pilotos apagados. Tras ver este estado hay que abrir la válvula de fondo mostrando en el display PULSADO y tras 8 segundos, si está verdaderamente vacío, aparece COMPROBAD. Luego, se conecta el brazo de carga, se acciona la palanca de la válvula de brazo de carga abriéndose el API y así el estado cambia a ACOPLADO, haciendo que el piloto de dicho compartimento pase a estar iluminado de manera intermitente. Cuando inicia la carga en el display aparece ahora LLENANDO, ya que se está llenando el compartimento. Una vez finaliza el llenado se desconecta el brazo de carga dando lugar a LLENO.

El siguiente avance entre estados tiene diferentes alternativas que pueden producirse en función de un hecho u otro. Si al conectar el Scully se detecta que el compartimento no está vacío, aparece CON RESTO haciendo que el piloto se ilumine de manera intermitente. Para eliminar esos restos hay que purgar el recipiente, conectando el acoplador API, abriendo la válvula de fondo y accionando la palanca API, manteniéndola abierta hasta que en el display aparezca PURGADO, que hará que el piloto luminoso se apague.

Posteriormente, hay que cerrar el acoplador API, desconectarlo y comprobar que en display aparece COMPROBAD.

Una vez se ha completado el proceso de carga de todos los depósitos, desconectar el Scully y comprobar que en el display todos los compartimentos cargados aparecen con el estado SELLADO haciendo que los pilotos luminosos permanezcan encendidos. Este estado es el que garantiza que cuando el producto llega a la estación de servicio establecida, el combustible se encuentra en las mismas condiciones en las que se cargó en la cisterna. De manera que en ese estado es como deben permanecer durante el transporte todos los compartimentos que hayan sido cargados. Si se detectara la apertura de cualquiera de ellos de manera anómala quedaría registrado en el sistema y el estado que aparecería sería el de DESPRECIN. Los compartimentos que no hayan sido cargados pasan a estar con VACIO.

Por último, en el caso de que el acoplamiento entre boca de carga y brazo de carga no se haga de forma correcta, el sistema entra en bucle mostrando alternativamente los estados ACOPLADO y DISPONIBLE hasta que se produce el acoplamiento de la manera adecuada.

Proceso de descarga

Para este proceso la secuencia de estados natural de un correcto funcionamiento es:

SELLADO-DESPRECIN-DESCARGAD-SIN PURGA- PURGAR- VACIO

Llegando el camión cisterna a la estación de servicio para el proceso de descarga se encuentran los compartimentos en estado SELLADO y tanto el piloto amarillo como los verdes correspondientes se encuentran encendidos fijos. Conectar la manguera de descarga, abriéndose la válvula de fondo de manera que el depósito cambia a DESPRECIN y el piloto pasa a estar iluminado de manera intermitente. Este estado de DESPRECIN no se considera anomalía ya que el camión se encuentra en una zona de descarga y hay que romper el sello para proceder a extraer el producto. Una vez accionada la palanca API, se mantiene abierta hasta que se realiza la descarga por completo apareciendo DESCARGAD. Ahora se cierra el API, se extrae la manguera y se comprueba como el display cambia a SIN PURGA. Tras haber hecho esto con los respectivos compartimentos a vaciar, se coloca la cisterna en una zona nivelada y si el compartimento queda vacío queda con el estado VACIO, pero si aún hay líquido pasan a PURGAR, continuando los pilotos intermitentes. Para pasar entonces a VACIO, se procede a purgar el compartimento en cuestión de la forma comentada anteriormente con el estado CON RESTO.

En el caso de que no se opere siguiendo los procedimientos aquí comentados, las variaciones quedarán registradas en el fichero histórico activándose en la pantalla un mensaje indicando que ha habido una incidencia de este tipo. Sin embargo, si se produce el fallo de algún sensor, se indica en el display como FALLO en el depósito afectado, parpadeando el piloto luminoso muy lentamente, advirtiendo al operario de que debe comunicarse con la central para recibir más instrucciones.

Si un camión llega a una estación de servicio con algún compartimento en estado DESPRECIN, el contenido será rechazado por el personal de esta, con las consecuencias económicas y de desabastecimiento de combustible que originaría y dando lugar a penalizaciones para la empresa encargada del transporte en el caso de que no aporte alguna justificación acerca del desprecintado de dicho compartimento.

4 DESARROLLO MAIN UNIT

En este capítulo se describe todo el desarrollo llevado a cabo para la creación y puesta en funcionamiento de una unidad central diseñada única y exclusivamente para el sistema de sellado de camiones cisterna, adaptado a tres compartimentos a modo de guía para su adaptación a cisternas mayores.

4.1 Node-Red

4.1.1 Descripción

El software utilizado en este trabajo es Node-Red, una herramienta de programación visual que se implementa en dispositivos controladores de hardware. Trabaja mostrando de manera visual las relaciones y funciones de manera que se pueda programar sin escribir. Es un panel de flujo al que se pueden incorporar nodos que se comuniquen entre ellos y puede instalarse en equipos como ordenadores Windows, Linux, o en la nube.

Fue creado en el 2013 por el grupo de Servicios de Tecnologías Emergentes de IBM por Nick O' Leary y Dave Conway, con el objetivo de ofrecer una solución para integrar un hardware con otros servicios. Permite hacer uso de tecnologías muy complejas centrándose únicamente en lo importante y obviando las tareas excesivamente complejas para el usuario. Se distingue porque permite conectar varios dispositivos de manera muy rápida, fácil y no requiere de conocimientos profundos de programación. Hoy en día, se puede instalar en casi la totalidad de los dispositivos IOT y ordenadores.

Esta herramienta de programación conecta varios dispositivos a la vez, tanto de hardware como de servicios de Internet, lo que lo hace perfecto para los equipos dedicados a trabajo industrial o como prueba de soluciones para equipos de planta que necesiten comunicarse entre sí. Se trata de un motor de flujos que cuenta con un enfoque IoT (Internet of Things), y esta condición permite definir flujos de servicios a través de protocolos como el MQTT, que es un protocolo de mensajería ligero para usar en casos de clientes que necesitan una huella de código pequeño conectados a redes con recursos limitados de ancho de banda.

El entorno de ejecución se fundamenta en node.js aprovechando al máximo su modelo sin bloqueo basado en eventos. Además, cuenta con más de 225000 módulos en el repositorio de paquetes que permite ampliar el rango de nodos de paleta para agregar nuevas capacidades. Esto hace a Node-Red ideal para ejecutarse en un hardware de bajo coste tal y como puede ser la Raspberry Pi, que será el hardware utilizado en este proyecto.

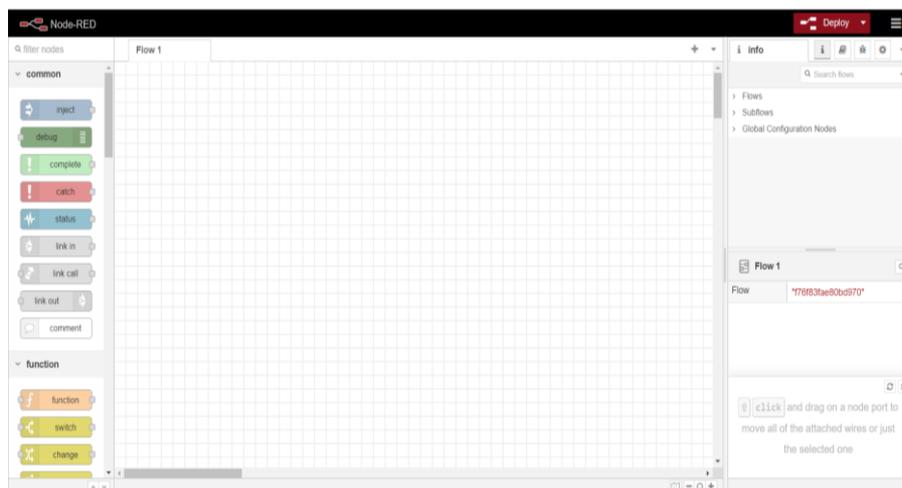


Ilustración 4-1. Ventana principal de Node-Red [22].

En esta ventana se pueden diferenciar las tres secciones más importantes de Node-Red. De izquierda a derecha se tiene primero la zona de los nodos, en medio los flujos y al final los distintos paneles del programa.

Los nodos son los encargados de realizar las tareas y transmitir la información entre unos y otros. Se dividen en tres tipos principales:

- Nodos de entrada: Cumplen la función de inyectar información en el flujo, de manera que permiten aportar toda la información necesaria para que el flujo pueda llevar a cabo su función y obtener así una salida. Se caracterizan por tener un único punto de conexión que se encuentra a la derecha del recuadro de dichos nodos.
- Nodos de función: Se encargan del tratamiento de los datos y de realizar las operaciones oportunas para obtener la salida deseada. Se pueden configurar para tener varias salidas y visualmente se caracterizan por tener una conexión a la izquierda y a la derecha de sus respectivos recuadros siendo la izquierda de entrada y la derecha de salida. De todos estos el más utilizado es el nodo *function*.
- Nodos de salida: Estos nodos tienen el único propósito de mostrar los datos que se han ido obteniendo a lo largo de todo el desarrollo del flujo, se caracterizan por sólo tener una conexión a su derecha, siendo el nodo *debug* el más corriente de todos.



Ilustración 4-2. Tipos de nodos [22].

Los nodos básicos se encuentran instalados en la versión inicial de Node-Red, pero el programa permite la instalación de los más de 1500 nodos adicionales que hay a través de la pestaña *Manage palettes* que se despliega de los paneles pudiendo así instalar todos aquellos que vayan a utilizarse en un flujo si necesidad de instalarlos todos, sino que se van haciendo por individual.

Los flujos contienen toda la parte lógica a donde se desplazan los nodos para poder configurarse como convenga y conectarse entre ellos. La información que se mueve entre ellos son los *.msg*, que tiene propiedades como el número de mensaje, payload o topic. Para realizar la configuración hay que hacer doble clic sobre el nodo a editar y proceder a rellenar los diversos campos que sean necesarios. Una vez realizado esto, es fundamental el hacer pulsar sobre el nodo *Deploy* que se encuentra en rojo arriba a la derecha en la ilustración 4-1, que significa desplegar y cuya función es la de compilar el flujo realizado, guardar el archivo y empezar a ejecutarlo. Si hay algún error se desplegará un aviso en rojo que lo notificará y si por el contrario todo está bien, aparecerá una ventana emergente en verde indicando que se ha desplegado con éxito.

Por último, la zona de los paneles se encarga de mostrar la información detallada de cada nodo, de cómo funciona, de que elementos necesita y de cómo se estructura, también muestra los mensajes de salida que se hayan configurado a lo largo del flujo, así como descripciones más detalladas de los errores que hayan podido haber en el programa. Además, hay una ventana de ayuda y otra de configuración general a cerca de los propios flujos [22].

4.1.2 Instalación

La descarga e instalación de este software se realiza a través de la propia página de Node js <https://nodejs.org/es/>, ya que Node-Red está creado a partir de NodeJS y JavaScript. En esta página se mostrará una ventana como la que se muestra en la ilustración 4-3, dónde únicamente hay que hacer clic sobre la versión deseada para que se realice la descarga.



Ilustración 4-3. Página web NodeJS [23].

Tras la realización de la descarga, se da paso a la instalación dónde lo primero que aparece tras ejecutar el archivo descargado es el Setup Wizard tal y como se muestra en la ilustración 4-4, aquí hacer clic en *Next > Instalar > Finalizar* teniendo así Node.js instalado.

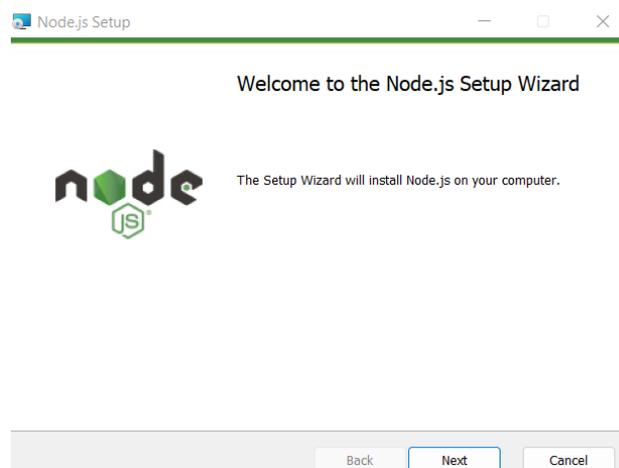


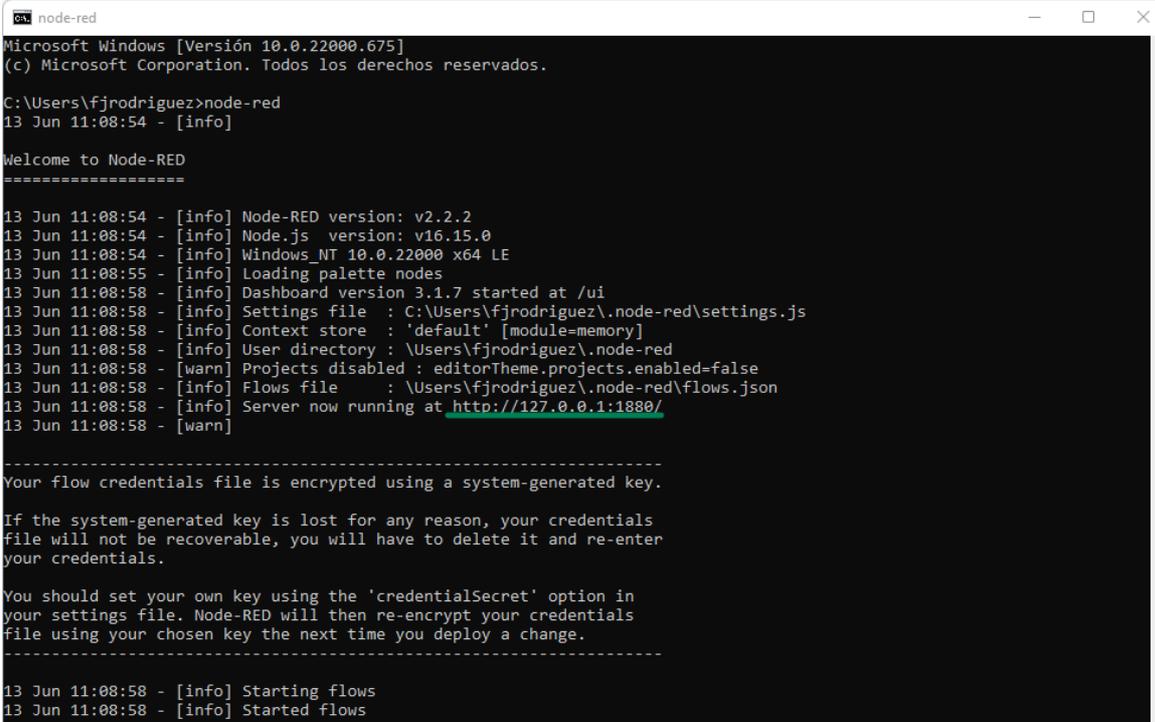
Ilustración 4-4. Instalador Setup Wizard.

Tras tenerlo instalado, los archivos instalados serán los que se ven en la ilustración 4-5, hay que hacer clic sobre *Node.js command prompt* de manera que se abrirá una ventana de símbolo del sistema dónde hay que escribir `npm install -g --unsafe-perm node-red` y ejecutarlo.



Ilustración 4-5. Archivos instalados.

Tras haber ejecutado ese código, ya se tendrá instalado Node-Red de forma que lo único que hay que hacer para ejecutarlo es abrir la aplicación Símbolo del Sistema del ordenador, escribir *node-red* y darle a Enter, tras varios segundos aparecerá la dirección del sitio web de Node-Red a la cuál acceder para poder hacer uso del programa.



```

node-red
Microsoft Windows [Versión 10.0.22000.675]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\fjrodriguez>node-red
13 Jun 11:08:54 - [info]

Welcome to Node-RED
=====

13 Jun 11:08:54 - [info] Node-RED version: v2.2.2
13 Jun 11:08:54 - [info] Node.js version: v16.15.0
13 Jun 11:08:54 - [info] Windows_NT 10.0.22000 x64 LE
13 Jun 11:08:55 - [info] Loading palette nodes
13 Jun 11:08:58 - [info] Dashboard version 3.1.7 started at /ui
13 Jun 11:08:58 - [info] Settings file : C:\Users\fjrodriguez\.node-red\settings.js
13 Jun 11:08:58 - [info] Context store : 'default' [module=memory]
13 Jun 11:08:58 - [info] User directory : \Users\fjrodriguez\.node-red
13 Jun 11:08:58 - [warn] Projects disabled : editorTheme.projects.enabled=false
13 Jun 11:08:58 - [info] Flows file : \Users\fjrodriguez\.node-red\flows.json
13 Jun 11:08:58 - [info] Server now running at http://127.0.0.1:1880/
13 Jun 11:08:58 - [warn]

-----
Your flow credentials file is encrypted using a system-generated key.

If the system-generated key is lost for any reason, your credentials
file will not be recoverable, you will have to delete it and re-enter
your credentials.

You should set your own key using the 'credentialSecret' option in
your settings file. Node-RED will then re-encrypt your credentials
file using your chosen key the next time you deploy a change.
-----

13 Jun 11:08:58 - [info] Starting flows
13 Jun 11:08:58 - [info] Started flows

```

Ilustración 4-6. Ejecución de Node-Red.

A pesar de que la dirección web subrayada en la ilustración 4-6 va a ser la dirección permanente de Node-Red en el ordenador, no podemos acudir a ella a no ser que hayamos ejecutado el comando *node-red* en la aplicación de Símbolos del sistema de forma que lo que aparece al intentar acceder es que no se puede cargar el sitio web, siendo indispensable haber realizado lo antes descrito.

4.2 Raspberry Pi 4 model B

4.2.1 Características

Como se ha comentado anteriormente, el hardware principal del proyecto es la Raspberry Pi, concretamente la Raspberry Pi 4 model B. Este modelo viene de una serie de ordenadores de placa reducida o placa única de bajo coste desarrollados por Raspberry Pi Foundation para permitir el acceso a la informática y creación digital a todo el mundo sin la necesidad de requerir de un avanzado conocimiento en informática y programación. Su sistema operativo es el Raspberry Pi OS, anteriormente denominado como Raspbian, aunque también permite la instalación de Windows 10.

La Raspberry Pi 4 model B consta de un procesador Broadcom, una unidad de procesamiento gráfico (GPU), memoria RAM desde 1 hasta 8 GB, 4 puertos USB, dos de ellos de tipo 3.0, con 10 veces más velocidad que los otros dos que son USB tipo 2.0, una conexión de cable Ethernet, 40 pines GPIO (General Purpose Input/Output), una entrada de alimentación tipo C, dos salidas micro HDMI, un conector de audio analógico Jack, un conector para una cámara y otro conector para un display. Además, aunque ninguna de sus versiones incluye memoria, consta de una entrada para tarjeta MicroSD y, aparte de la conexión Ethernet, permite la conexión por Wi-Fi y cuenta con Bluetooth 5.0 [24].

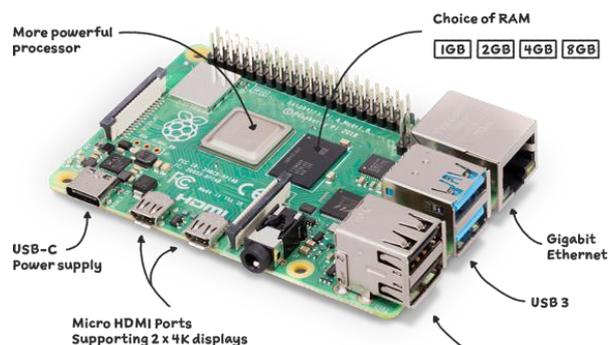


Ilustración 4-7. Raspberry Pi 4 model B [24].

La elección de este hardware se debe a su gran uso en el mundo de la industria tanto como ordenadores como sistemas de revisión y ejecución de tareas, además es un elemento muy presente en diversos campos de las oficinas de MONTREL S.A. por lo que está bastante asentado en la empresa y es conocido por gran parte de la totalidad de sus trabajadores. Su uso principal en el taller, entre otros, es el de ser el sistema mediante el cual fichan los trabajadores y además realiza la imputación de horas.

4.2.2 Acceso y uso

La manera más sencilla para poder uso de la Raspberry es alimentarla con una fuente de 5 V, conectarla a un monitor a través de un cable HDMI y conectarle un teclado y un ratón para hacer uso de ella de la misma tal y como si fuera un ordenador al uso. A pesar de que esta sea la forma más corriente, la particularidad del micro HDMI o el hecho de no tener un monitor a mano puede complicar la utilización de la Raspberry, pero para ello existe una forma alternativa de poder acceder a ella de manera remota, con la única necesidad de tener un cable Ethernet conectado a la Raspberry junto con la alimentación. Esto se realiza mediante un acceso SSH, Secure Shell, que es el nombre de un protocolo que permite el acceso remoto a un servidor por medio de un canal seguro en el que toda la información está cifrada.

El programa que se ha utilizado para dicha conexión es el programa PuTTY, que es un emulador gratuito que admite varios protocolos de red tales como el SSH comentado anteriormente. Para descargarlo hay que acceder a <https://www.putty.org/> y hacer clic sobre la versión para Windows, en este caso. Tras realizar la descarga, seguir los pasos del instalador Wizard de la misma forma que para instalar Node-Red. Una vez ya está instalado, abrir el programa y se mostrará una ventana como la que se aprecia en la ilustración 4-8.

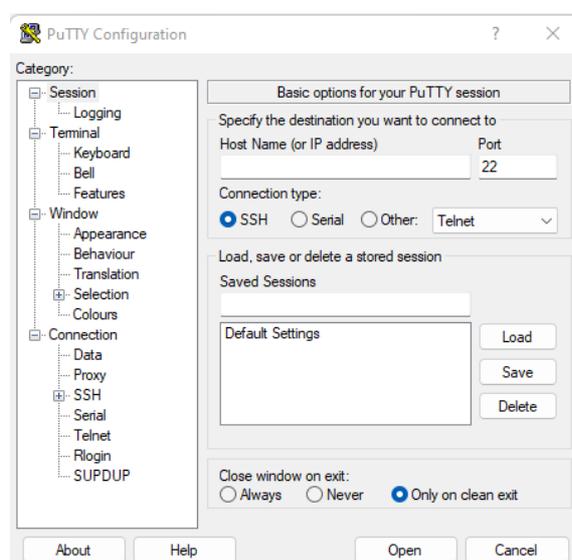
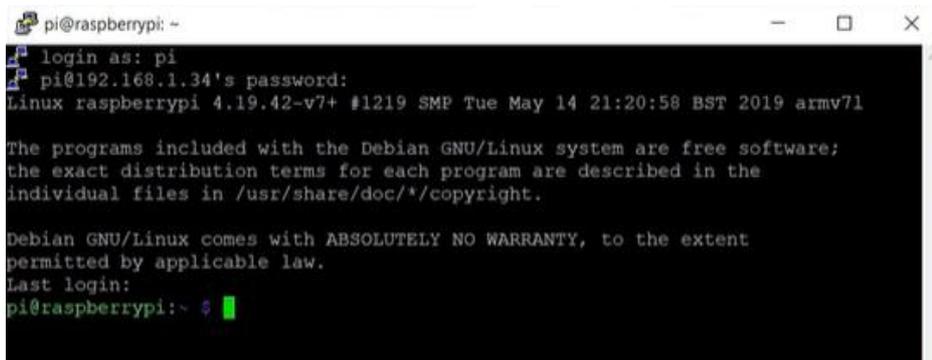


Ilustración 4-8. Ventana inicial del programa PuTTY.

Lo único que se necesita de todo lo que ahí aparece es la dirección IP de la Raspberry, una vez se escribe en el programa, se despliega una ventana similar a la de la aplicación Símbolos del sistema dónde pedirá el usuario y la clave de sistema operativo de la Raspberry, tras introducirlos ya se tiene acceso a ella a través de líneas de comando quedando la ventana tal y como se muestra a continuación:



```

pi@raspberrypi: ~
login as: pi
pi@192.168.1.34's password:
Linux raspberrypi 4.19.42-v7+ #1219 SMP Tue May 14 21:20:58 BST 2019 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login:
pi@raspberrypi:~

```

Ilustración 4-9. Ventana de control de la Raspberry.

También, existe la posibilidad de poder visualizar la interfaz gráfica de la Raspberry sin necesidad de conectarla a un monitor, esto se realiza a través de un visualizador. Para ello, el programa utilizado ha sido VNC Viewer. El primer paso para su instalación es ejecutar el comando `sudo apt-get install -y realvnc-vnc-server realvnc-vnc-viewer` en la ventana mostrada en la ilustración 4-9, una vez ejecutado y completada su carga, ejecutar el comando `vncserver`, que una vez se ejecute en la última línea que despliega mostrará una dirección IP donde está el escritorio de la Raspberry. El acceso a esta dirección se debe hacer a través de la aplicación mencionada antes VNC Viewer, su descarga se realiza a través su página web <https://www.realvnc.com/es/connect/download/viewer/>. El procedimiento es similar al de los demás programas de cara a la instalación, ejecutar el Wizard que emerge en la pantalla hasta finalizarlo. Una vez concluya, abrir la aplicación y aparecerá la ventana principal que se muestra en la siguiente ilustración:



Ilustración 4-10. Ventana principal VNC Viewer.

Una vez aquí, hacer clic en *Archivo > Nueva conexión* e introducir la dirección IP que obtuvimos al ejecutar el comando `vncserver` en el recuadro con el nombre VNC Server. Tras esto ya tendremos la conexión con el escritorio de la Raspberry realizada.

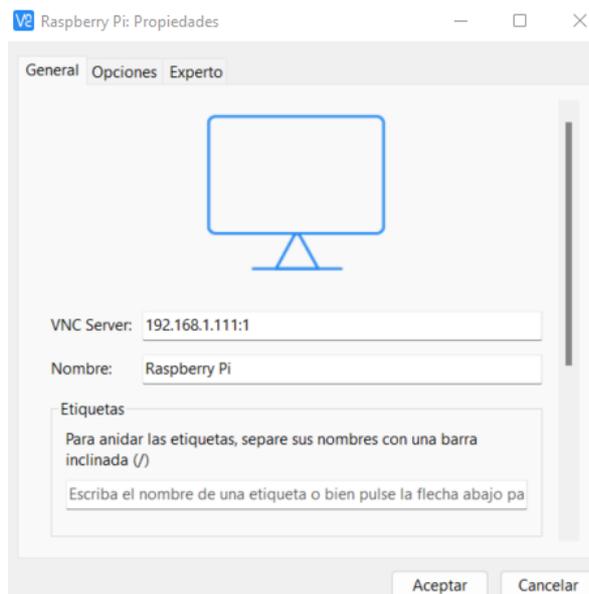


Ilustración 4-11. Introducción IP para conexión remota con el escritorio de la Raspberry.

Tras darle a *Aceptar* y abrir la ventana que aparece en la pantalla principal de la aplicación, ya aparece el escritorio de la Raspberry en pantalla pudiendo controlarse exactamente igual a cómo se realiza si se hiciera la conexión mediante la conexión directa de HDMI.

Al igual que ocurre al ejecutar Node-Red en el ordenador, no podemos acudir al escritorio de la Raspberry de forma remota si no se ha iniciado PuTTY y, una vez se ha accedido al sistema operativo tras introducir usuario y contraseña, se ha ejecuta el comando *vncserver*, de forma que si se intenta acceder sin hacer este proceso lo que aparece es que no se puede acceder porque no hay conexión.

4.3 Programación

El desarrollo de los programas para el control de los compartimentos y monitorización de los estados se ha realizado a través del Node-Red. Con la finalidad de poder avanzar en la programación y la facilidad a la hora de acceder al programa, primero se desarrolló en un ordenador y tras haber comprobado que funcionaba correctamente a través de un dashboard se adaptó a la propia Raspberry Pi 4. Por un lado, se va a explicar la programación realizada para el control mediante el ordenador y, por otro, se explicarán las modificaciones realizadas para la adaptación a la Raspberry.

Lo más importante de cara al inicio del desarrollo de la programación es conocer las entradas que se tienen ya que esa será la información con la que opere el programa y en torno a las cuales desarrolla sus funciones. Estas son:

- La lectura del conector Scully para identificar si el camión cisterna se encuentra en una isleta de carga y descarga o no, que será la variable *Scully*.
- La presión que tiene el circuito general, almacenada en *PN_CirPrin*.
- La lectura del sensor de la boca de hombre de cada compartimento para identificar si está abierta o no, para lo que se utilizarán las variables *BocaHom_i*.
- La lectura del sensor del acoplador API de cada compartimento para ver si se encuentra cerrado o no, que se almacenará en las variables *AcoplAPI_i*.
- La lectura del sensor de presencia de líquido de cada compartimento para ver si se encuentra vacío o contiene líquido, que serán las variables *PresenLiq_i*.
- La lectura del sensor presión de la válvula de fondo de cada compartimento para ver si se encuentra abierta o no, asignado a las variables *ValvF_i*.

Estas entradas son las que se van a controlar a través del dashboard para simular el montaje real y para ello se ha instalado el módulo de la dashboard para Node-Red ya que inicialmente no se encuentra incluido. Por tanto, se leen a través de la interfaz y en el programa esa lectura es proporcionada a cada variable en función de cuál sea su valor leído. Eso se realiza de la siguiente manera en un nodo *function*:

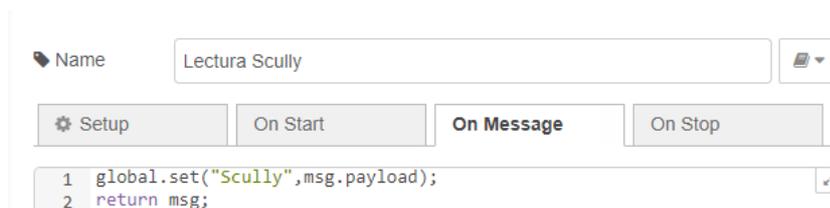


Ilustración 4-12. Código para lectura de variable *Scully*.

Aquí se declara una variable global, a la cual se puede acceder a lo largo de todos los flujos, denominada “Scully” y se le da el valor que se ha leído del dashboard almacenado en el payload del mensaje. Quedando conectado tal que así:

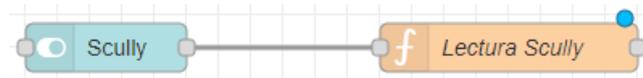


Ilustración 4-13. Nodos necesarios para la lectura de *Scully*.

Siendo el nodo celeste el de lectura del dashboard y el naranja el nodo *function*. Siguiendo este mismo procedimiento se hacen las lecturas de todas las entradas anteriormente enumeradas, es decir, de la boca de hombre, del acoplador API, del sensor de presencia de líquido y el sensor de la válvula de fondo.

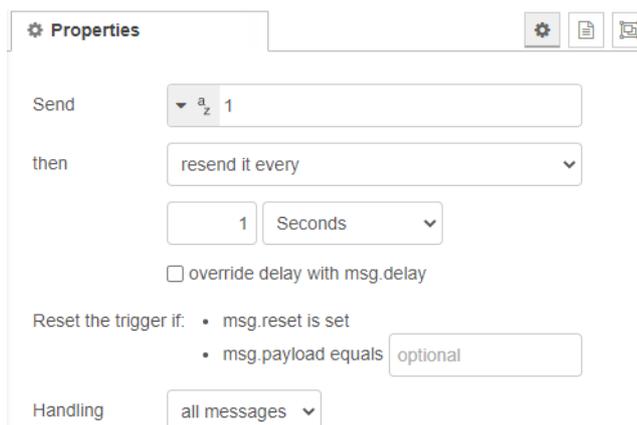
Hay que destacar que en este apartado se van a desglosar los programas necesarios para un compartimento solamente ya que son iguales entre ellos con la única diferencia de la nomenclatura entre variables. Para cada compartimento *i*, llevan al final *_i* a modo de identificación, por lo que la variable de la boca de hombre del compartimento 1 será *BocaHom_1*, mientras que la del compartimento 2 será *BocaHom_2*, y así con todas las demás variables empleadas. Además, los programas de dicho compartimento, los flujos al completo y la unión entre los distintos nodos se encuentran en el anexo adjunto al final del trabajo.

Una vez se tiene la lectura por parte del programa de todas las variables a considerar, y teniendo en cuenta los requisitos que harán falta tener para la transición entre estados, se ha dispuesto una configuración para contabilizar tiempos de conexión o de desconexión de una determinada variable todos de la misma forma mostrada a continuación:



Ilustración 4-14. Configuración del contador para la variable *Scully*.

Esto está estructurado de manera que el primer nodo, el nodo *start-up-trigger*, hace que este flujo empiece a funcionar a la vez que se conecta a Raspberry o se ejecuta el programa, posteriormente está la función *Inicializa contador* donde se define la variable global *contador* que va a contabilizar el tiempo y se inicializa a 0. Por la otra rama de la salida del primer nodo, se encuentra un *resend*, nodo que se encarga de reenviar la señal que tiene definida de manera que se activa cada cierto tiempo, en concreto, este nodo envía una señal 1 cada segundo tal y como se aprecia en la configuración del propio nodo mostrada a continuación:

Ilustración 4-15. Propiedades nodo *resend*.

Tras esto hay un nodo *function* cuya función es la de contabilizar los 5 segundos que se encuentra el mecanismo Scully desactivado bajo un estado en concreto especificado en el código programado dentro, que en este caso son los estados de LLENO, LLENANDO y SELLADO de cada compartimento. Siempre que esto ocurra se activará una variable que indique que el Scully lleva ya 5 segundos desactivado, que es la variable global *NoScully*. Los 5 segundos se contabilizan con una variable *contador*, que se va sumando de 1 en 1 cada vez que se ejecuta la función, que debido al *resend* previo, lo realizará segundo a segundo pudiendo realizar así la contabilización del tiempo. En caso contrario de tener el compartimento cualquier otro estado el contador será cero y la variable *NoScully* será false.

Esta configuración es la que van a tener todos los contadores utilizados a lo largo de los diferentes flujos para contabilizar los tiempos de funcionamiento o desconexión de las variables requeridas. Habrá uno para cada variable *ValvF_i* y otro para cada variable *PresenLiq_i*, ya que por un lado se necesita contabilizar cuando la válvula de fondo del compartimento en cuestión lleva 3 y 5 segundos abierta y por otro, se necesita contabilizar cuando el sensor de presencia de líquido se encuentra sin detectar líquido durante 5, 8 y 20 segundos.

Continuando con todo lo necesario de cara a la declaración de variables y su identificación, destacar que la lectura de la presión neumática del circuito mediante el dashboard no se realiza mediante un switch tal y como se ha hecho con el Scully o con todas las demás variables, sino que se hace mediante un nodo de entrada numérica cuyo nodo y cuya representación en el dashboard se muestra a continuación:

Ilustración 4-16. Nodo de lectura y representación de *PN_CirPrin*.

Por último, en lo referente a los flujos de entradas y el dashboard, resaltar que en este se muestra tanto la fecha como la hora actual. Esto se realiza utilizando el nodo *inject*, que mediante la asignación de *timestamp* al payload del mensaje, da la cantidad de milisegundos que hay desde el 1 de enero de 1970 hasta la actualidad. Esa cantidad, se transforma a formato HH:mm:ss para obtener la hora, minutos y segundos actuales y, también se transforma a formato DD/MM/YYYY para obtener así la fecha en la que estamos. Para realizar esos cambios en el formato del *timestamp*, se ha descargado el nodo *Date/Time Formatter* y se ha configurado como se ha comentado. Posteriormente para mostrarlo se ha hecho uso del nodo *text* que proporciona una salida de texto en el dashboard, quedando tal que así:

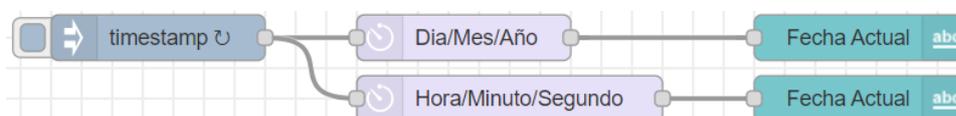


Ilustración 4-17. Distribución de los nodos para mostrar la fecha y la hora.

Y siendo la programación del *timestamp* y del *Date/Time Formatter* la siguiente:

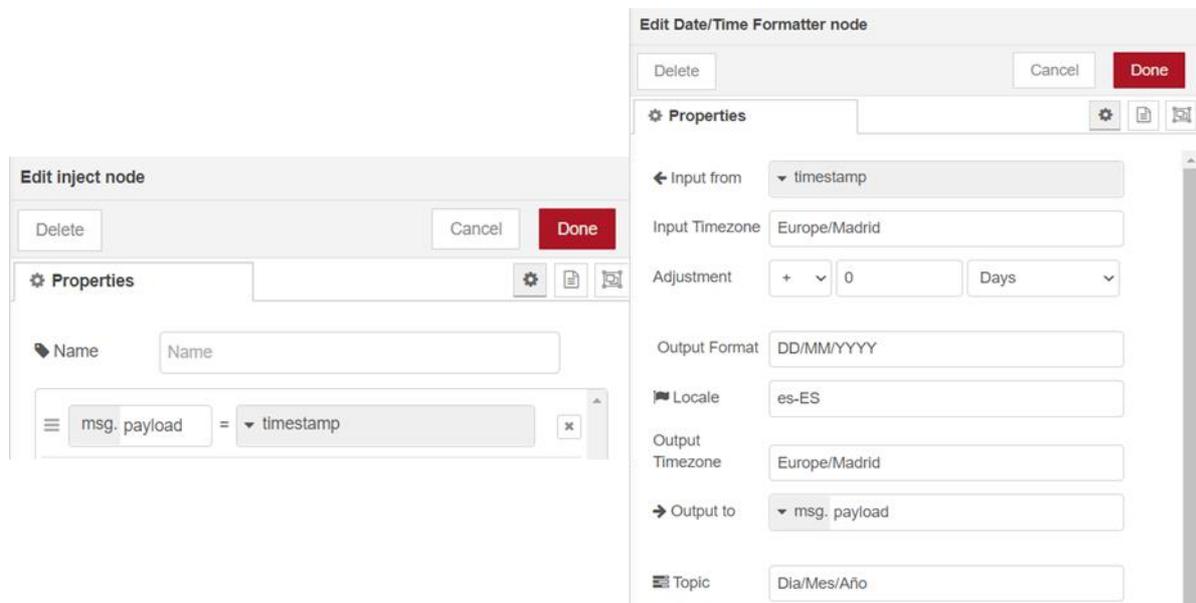


Ilustración 4-18. Propiedades nodos *inject* y nodo *Date/Time Formatter*.

Habiendo concluido ya el desarrollo de los nodos y códigos necesarios para la lectura y funcionamiento de todas las variables de entrada que se requieren en el sistema de sellado, ahora se va a proceder a explicar la programación de la máquina de estados que es el núcleo en torno al que giran todos los flujos, ya que este se encarga de realizar las transiciones entre estados. Recaltar también que únicamente se va a desarrollar la máquina de estados de un compartimento debido a que la programación es la misma para todos ellos con la única salvedad de la nomenclatura de las variables.

A continuación, se muestra la configuración general de la máquina de estados referente al compartimento 1:

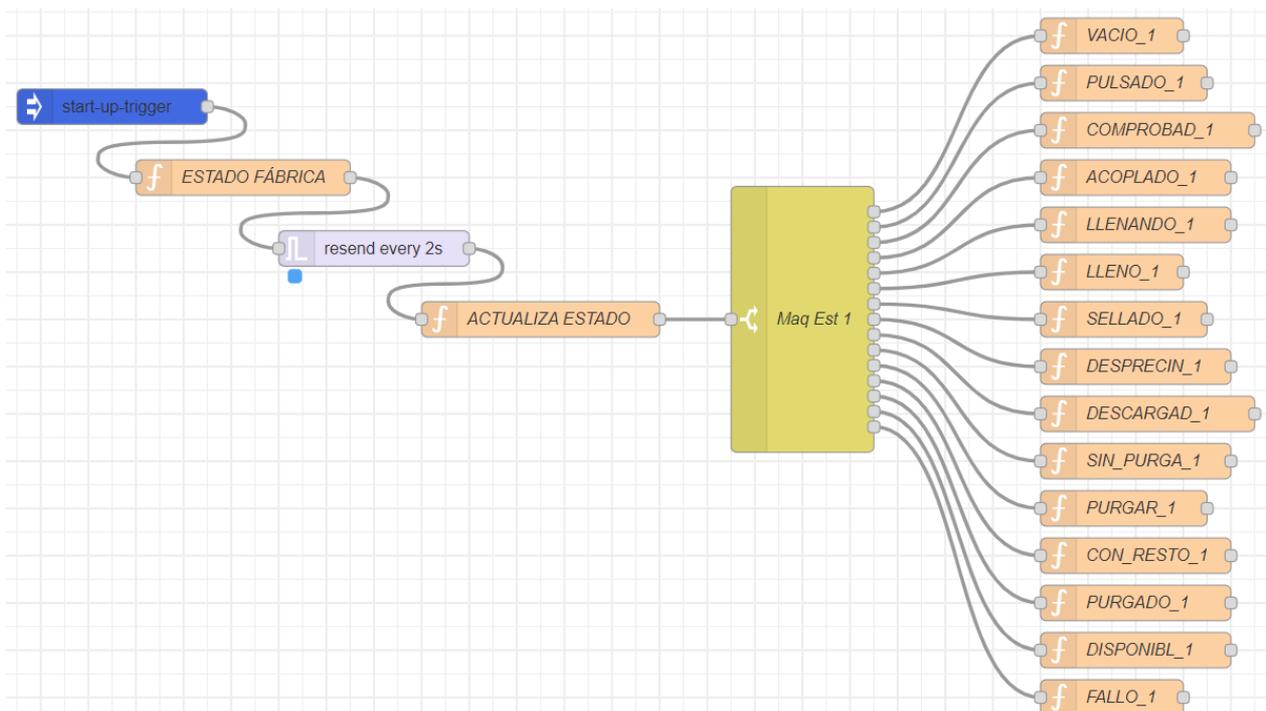


Ilustración 4-19. Flujo de la máquina de estados compartimento 1.

Ahora, se va a ir desglosando paso a paso los nodos utilizados y su configuración, añadiendo también los requisitos para que se dé un estado u otro facilitando así la comprensión de la programación realizada.

Primero se tiene un nodo *start-up-trigger*, que hace que el flujo se empiece a ejecutar al instante de pulsar el *Deploy* o al instante de encender la Raspberry. Luego se tiene un nodo *function* llamado *Estado Fábrica* que inicializa el estado del compartimento a VACIO, estado que viene predeterminado de fábrica y del que se parte inicialmente ya que se tiene en cuenta que el flujo arranca por primera vez al conectarse al camión cisterna deseado y permanece en constante funcionamiento de manera cíclica haciendo la lectura entre los distintos estados en los que se va encontrando, es decir, una vez que el flujo se ejecuta por primera vez, no vuelve a pasar por la función *Estado Fábrica*, por eso se encuentra antes del *resend*. Este *resend* que se encuentra a continuación, tiene un tiempo de ejecución de 2 segundos para poder visualizar en el ordenador cómo van modificándose los estados sin que haya un excesivo número de salidas al conectar un *debuj* al final del flujo. Al implementar este flujo en la Raspberry, este valor se sustituirá por el valor de 250 milisegundos.

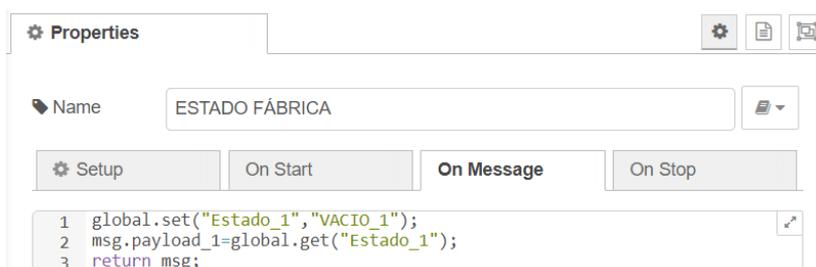


Ilustración 4-20. Código y propiedades de la función *ESTADO FÁBRICA*.

La función *Actualiza Estado* se encarga de ir escribiendo en el payload el estado en el que se encuentra la variable global *Estado_i*, permitiendo así que cualquier cambio de estado se almacene en esa variable y pueda entrar en el switch que hay después.

El nodo *Maq Est 1* tiene la función de dirigir el flujo hacia un estado u otro en función del valor del payload, es decir, en el caso de que en el `msg.payload` se tenga `VACIO_1`, el switch dirigirá la salida únicamente por aquella designada para dicho estado. Esto se realiza designando, dentro de las propiedades del nodo *switch*, tantas salidas como estados haya, concretamente 15. Hay que tener cuidado aquí ya que la lectura del `msg.payload` tiene que ser exactamente igual a cada uno de los comandos designados para cada salida, por lo que un simple espacio de más impide que el flujo continúe por dónde debe.

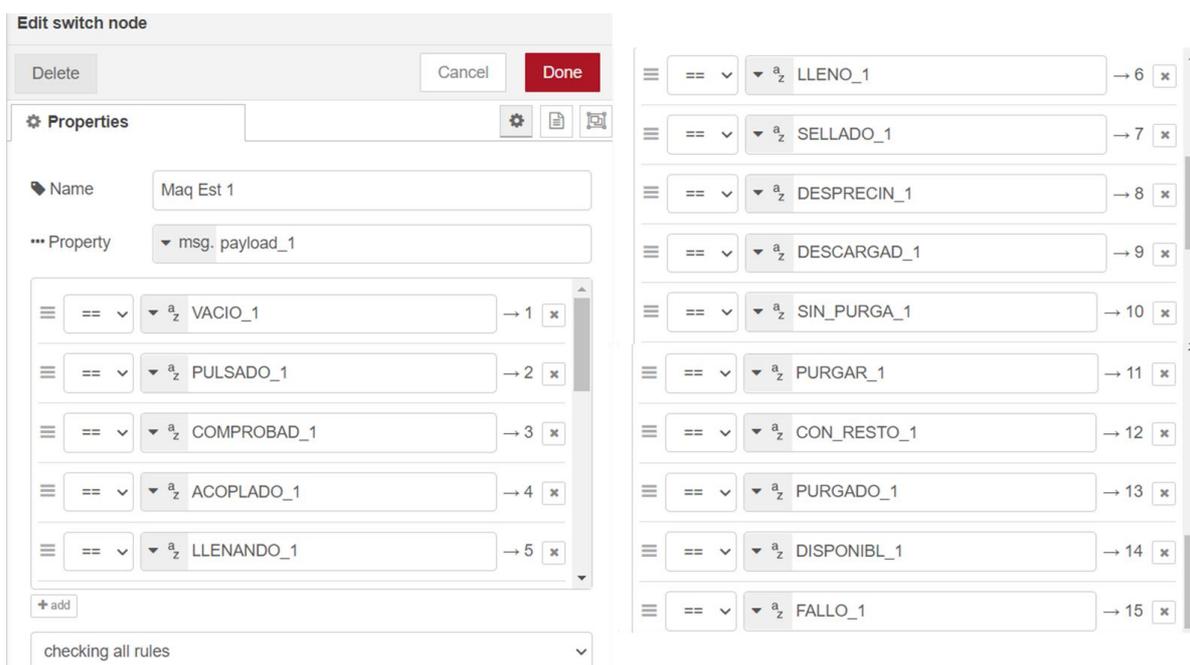


Ilustración 4-21. Propiedades nodo switch denominado *Maq Est 1*.

Tras pasar el flujo por este nodo y distribuirse a una salida u otra en función del valor el estado, ahora se tienen todas las funciones de los distintos estados posibles dónde, en función de una serie de condiciones, se harán unas transiciones u otras.

En primer lugar, se tiene la función `VACIO_1`, estado desde el que se puede transitar al estado `PULSADO` o `CON_RESTO`. Se pasa a estado `PULSADO` siempre que el Scully esté conectado, el acoplador API esté cerrado y la válvula de fondo lleve abierta 3 segundos, activando la variable auxiliar `PulsVF_1`. Para pasar a estado `CON_RESTO`, además de que el Scully esté conectado, condición imprescindible en la mayoría de los estados, el sensor de presencia de líquido tiene que encontrarse encendido, es decir, detectando líquido en el interior de compartimento. Todo esto se encuentran programado en el código de dicha función, mostrado en la ilustración 7-3 del anexo.

El siguiente estado que se tiene en el nodo `switch` es el estado `PULSADO_1`, cuyo objetivo es el de comprobar en la isleta de carga que no queda producto en ninguno de los compartimentos, pudiendo pasar a los estados `CON RESTO`, en el caso de detectar la presencia de líquido, al estado `COMPROBAD`, si el acoplador API se encuentra cerrado y el sensor de líquido se encuentra 8 segundos sin detectar combustible, activando la variable `NoLiq_1`, o al estado `ACOPLADO` si no se detecta la presencia de líquido y el acoplador API esté conectado.

El estado que viene después es `COMPROBAD_1`, cuya única transición es al estado `ACOPLADO`, siendo las condiciones necesarias para ello que el Scully esté conectado y que el acoplador API se encuentre conectado, siendo todos los demás sensores indiferentes. Este estado se alcanza cuando se comprueba que verdaderamente los depósitos están vacíos y se puede proceder a la carga.

A continuación, el estado descrito es `ACOPLADO_1`, que representa el acoplamiento de la manguera de carga al depósito justo antes de empezar a llenarlo con líquido. Si se produce un desacople para, por ejemplo, abortar la operación, no se pasará directamente a `VACIO`, sino que se transitará al estado `DISPONIBL`, bajo las condiciones de que no haya líquido en el compartimento y tanto el acoplador como la válvula de fondo se encuentren cerrados. La otra transición es al estado `LLENANDO`, que comienza cuando, estando el acoplador abierto, es decir, conectado, se detecta la presencia de líquido. El paso a este último estado registra una anomalía en el caso de que dicha transición se realice con algún compartimento en estado `CON RESTO` ya que eso indicaría que el conductor no ha purgado todos los compartimentos antes de iniciar la carga.

Para registrar una anomalía en el evento en el que se produce se le asigna a la variable global `ANOMALIA_i` el valor de SI, esto hará que en un flujo paralelo se ejecute una función que escribe en un registro todas las anomalías que se van produciendo en los distintos compartimentos de la cisterna. Este flujo se estructura así:

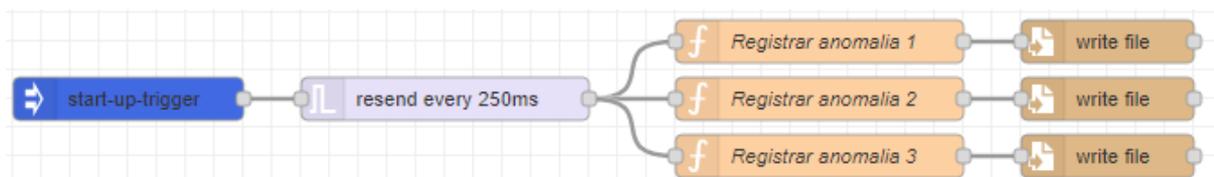


Ilustración 4-22. Flujo para registrar una anomalía.

En cada una de las funciones `Registrar anomalía` se asigna un payload estructurado de manera que en una línea se escribe primero el suceso ocurrido, que es una anomalía en un determinado compartimento, y posteriormente se indica la fecha y la hora en el que se ha producido dicha anomalía, haciendo uso de las variables globales `Fecha` y `Hora` en las que se van almacenando, haciendo la variable `ANOMALIA_i` igual a NO para sólo escribirlo una única vez por cada estado. En el nodo `write file` sólo hay que indicar la dirección dónde se pretende almacenar el archivo junto con el nombre que se le va a asignar, sea el de la Raspberry o el del escritorio del ordenador utilizado por ejemplo.

Siguiendo con la descripción de los estados, se tiene la función `LLENANDO_1`, situación en la que permanece durante todo el proceso de carga y desde la que se transiciona a tres posibles estados. Primero a `LLENO`, cuando se tienen el acoplador, la boca de hombre y la válvula de fondo cerrados y se detecta combustible en el interior del compartimento. Luego a `ACOPLADO`, que se producirá cuando se aborte la operación de carga y no se detecte líquido y el acoplador se encuentre abierto. Y, finalmente, puede pasar a

SELLADO si el Scully está desconectado durante 5 segundos antes de retirar el brazo de carga, teniendo que se detecta líquido en el compartimento y tanto la boca de hombre, como la válvula de fondo como el acoplador se encuentran cerradas, además de que ningún compartimento se encuentre en estado CON RESTO. Las dos últimas transiciones registran una anomalía. El código de esta función se encuentra en la ilustración 7-8 del anexo.

En siguiente lugar se tiene la función para el estado LLENO, que se mantiene una vez que el compartimento ha sido cargado y la cisterna sigue conectada a la isleta de carga a través del Scully. Desde LLENO se puede pasar a SELLADO, bajo las mismas condiciones comentadas en el estado anterior, o a LLENANDO si se detecta líquido, el Scully está conectado y la válvula API se encuentra abierta, registrándose en este caso una anomalía.

A continuación, se tiene el estado SELLADO que es el estado principal de todo el sistema al que tiene que llegar todo compartimento llenado, puesto que indica que ha sido cargado y que no ha sufrido ninguna manipulación, debiendo llegar así al estado de descarga. También se permite la recarga de un depósito sellado por lo que se permite la evolución al estado LLENANDO, que se produce al tener conectado el Scully, estar el acoplador API abierto y al detectarse líquido en el compartimento. La otra transición que se da desde este estado es a DESPRECIN cuando el Scully está desconectado y se ha abierto la boca de hombre, o el acoplador API o la válvula de fondo, registrándose una anomalía también en este caso.

Dado el estado DESPRECIN, la única transición posible es a DESCARGAD, para lo cual hace falta que el camión no esté conectado a la isleta de carga, no tener líquido durante al menos 20 segundos, y tener tanto la válvula de fondo como el acoplador abiertos.

Seguidamente se tiene DESCARGAD_1, estado intermedio que ocurre cuando se ha vaciado el depósito y se encuentra cerrado. La única posible evolución es al estado SIN PURGA cuando se da que el Scully está desconectado, el acoplador y la válvula de fondo están cerradas y además no se detecta líquido.

Luego se encuentra el estado SIN_PURGA_1, que refleja el momento en el que se ha de comprobar la inexistencia de líquido en los compartimentos descargados, para lo cual el camión debe estar en una superficie horizontal, abrir la válvula de fondo durante 5 segundos, y en función de si detecta líquido o no se pasa a VACÍO o a PURGAR.

Si la transición es al estado PURGAR, el conductor está obligado a purgar el compartimento antes de abandonar la estación de servicio, por lo que estando el Scully desconectado, el API abierto y la válvula de fondo abierta, en el momento en el que se deja de detectar líquido durante 5 segundos, se evoluciona al estado VACIO.

En siguiente lugar se encuentra el estado CON_RESTO_1, en el que se permanece hasta que se realiza una purga del compartimento independientemente de si se procede de estado VACIO o PULSADO. Para ello hace falta que se active la señal de conexión del Scully, la válvula de fondo junto con el acoplador estén abiertos, y no se detecte líquido durante 5 segundos, pasando así a PURGADO.

El estado PURGADO refleja que, aunque el depósito ya se ha purgado, sigue conectado a la isleta de carga, y permanecerá en él hasta que se desactive la señal de los sensores del acoplador API y la válvula de fondo, dando lugar al estado DISPONIBL.

En siguiente lugar aparece DISPONIBL, estado de reposo en el que todas las válvulas se encuentran cerradas aunque el Scully aún permanece conectado. Sólo se puede pasar a ACOPLADO en el momento en el que se abre el acoplador.

Por último, se tiene el estado de FALLO, al que se llega de manera independiente de cualquier estado que se haya dado anteriormente cuando hay algún sensor del compartimento en fallo, en cortocircuito o desconectado. El depósito permanece así hasta que se restaure el fallo, dónde se pasará al estado SIN PURGA, teniendo que realizarse un purgado del mismo antes de comenzar a usarlo por cuestiones de seguridad.

Mediante un flujo adicional, se van almacenando los estados y las variables en un archivo hasta que algún compartimento entra en estado de FALLO, en ese momento se restaurará o reparará el fallo en cuestión, teniendo que llamar a la centralita encargada del camión en cuestión ya que es desde ahí desde dónde se ejecuta el reseteo para restaurar todos los estados. Una vez el compartimento en cuestión o todo el sistema ha dejado de estar en este estado, la función *FALLO_i* se encarga de cambiar el estado a SIN PURGA, siguiendo ya con total normalidad el funcionamiento de la máquina de estados y el propio compartimento.

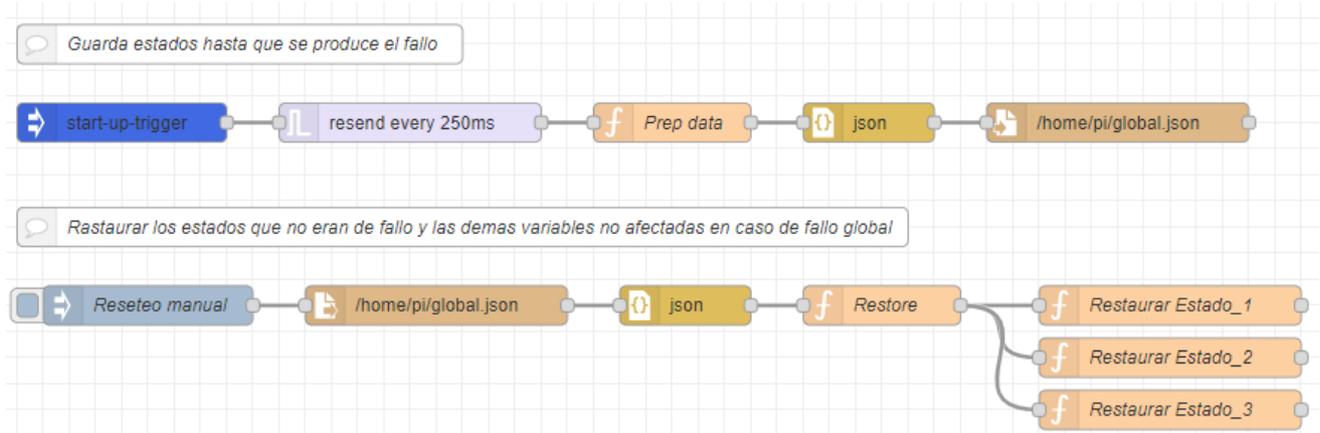


Ilustración 4-23. Flujo Guardar ante Fallo.

Llegados a este punto, ya se tiene descrito todo lo que engloba a la máquina de estados, tanto los códigos de programación de cada función como los flujos extras para registrar tanto anomalías como controlar los fallos, además de la lectura de las variables que se hizo al principio del apartado. Ahora se va a proceder a explicar cómo se controlan las salidas que tiene el programa, que únicamente son los leds de cada depósito, el led de presión del circuito principal y la salida por pantalla del estado de cada compartimento.

Los leds de salida tienen tres situaciones diferentes, apagado, encendido fijo y encendido intermitente. Permanecerá encendido de manera fija únicamente cuando se encuentre en el estado SELLADO, estará apagado en estado VACIO, COMPROBAD, PURGADO y DISPONIBL, y parpadeará de manera intermitente en el resto de los estados que son PULSADO, CON RESTO, ACOPLADO, LLENANDO, LLENO, DESPRECI, DESCARGAD, SIN PURGA y PURGAR. Para poder programar estas diferentes opciones se ha realizado flujo mostrado en la ilustración 4-24.

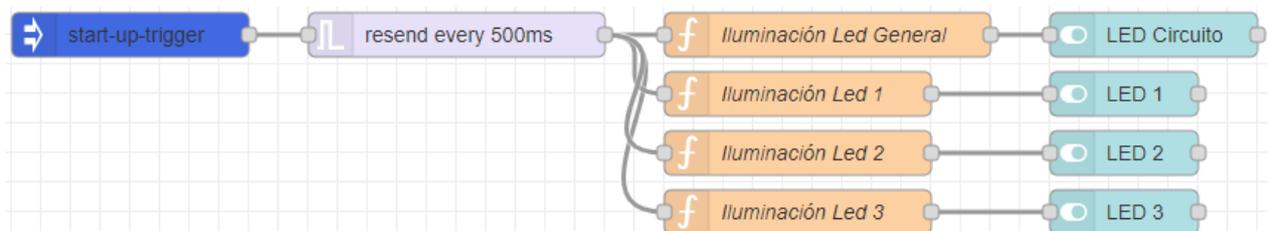


Ilustración 4-24. Flujo para la iluminación de los leds a través del dashboard.

Aquí lo que se está realizando es comprobar en qué estado se encuentran las variables $PN_CirPrin$ y los respectivos $Estado_i$. La importancia de $PN_CirPrin$ recae en que es un requisito esencial la existencia de presión general en la cisterna para activar los pilotos leds de cada compartimento. El led correspondiente a dicha presión tendrá color amarillo mientras que los correspondientes a los compartimentos serán de color verde.

El valor utilizado para ver la existencia de presión general es un valor mayor a 0 ya que se ha programado teniendo en cuenta que la presión reflejada es la presión relativa, de forma que un valor de 0 representaría 1 bar de presión absoluta. Se ha realizado así ya que en la mayoría de las instalaciones de este tipo se opera con la presión relativa. Además, como la entrada de texto para introducir los valores está en modo de números enteros, cualquier valor mayor de 0, indicará la existencia de presión general en el circuito encendiendo así el led, cuya programación de salida al dashboard es:

Ilustración 4-25. Propiedades del nodo *LED Circuito*.

Dónde se define que el color de encendido es el amarillo y mientras no lo esté será blanco para que no se muestre nada, siendo el comando *fa-lightbulb-o* el comando que hace que ese switch se vea en vez de en forma de interruptor, en forma de bombilla simulando lo que sería el led. Para los leds de cada compartimento, las propiedades del son casi iguales únicamente cambiando el color amarillo por el verde, que se encuentra recuadrado en azul en la ilustración anterior, y el grupo en el que se encuentran, que sería el recuadro rojo, siendo el led del compartimento 1 perteneciente al grupo [Sistema de Sellado] Compartimento 1.

La programación respecto a la iluminación de los leds de cada compartimento es similar, por lo que sólo se va a profundizar en la explicación del programa y configuración de uno nada más.

En la función *Iluminación Led 1* primero se tiene en cuenta que el led general esté encendido y posteriormente se ve en qué estado se encuentra la variable correspondiente *Estado_i*, de manera que si encuentra en el estado en el que está encendido fijo la salida correspondiente se hace true, si se encuentra apagado se hace false, y para programar la intermitencia se ha definido una variable local en la que se guarda el valor contrario de la variable *Estado_Led_i*, pudiendo guardar en dicha variable local ese valor opuesto, hecho que ocurre cada medio segundo que se va ejecutando la función permitiendo así el parpadeo del led y estando medio segundo encendido y otro medio segundo apagado, haciendo así la intermitencia.

Por último, para mostrar el estado de cada compartimento a través del dashboard se realiza un flujo igual al de los leds con la diferencia de que la salida es una salida de texto llamada Estado, en el que se mostrará el estado de cada compartimento en ese instante. El *resend* que tendrá será algo inferior al de los leds para ser capaz de detectar las variaciones de estado de manera prácticamente instantánea.

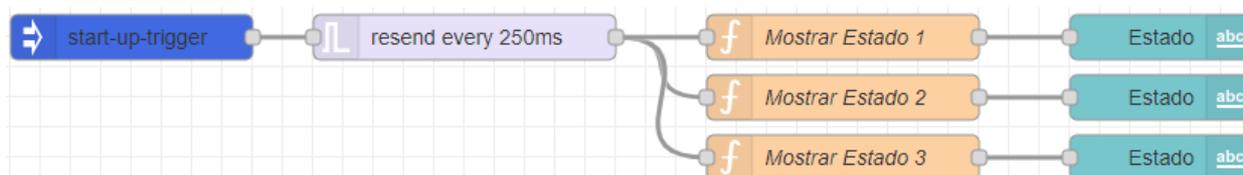


Ilustración 4-26. Flujo para mostrar en el dashboard el estado de cada compartimento.

Cada función *Mostrar Estado i* se encarga de guardar en el *payload* de ese *msg* el valor de la variable *Estado_i* para que cada salida de texto sea capaz de mostrar dicho *payload*.

Adicionalmente, para la ejecución de los flujos en Node-Red y poder hacer todas las simulaciones oportunas, en la lectura de variables se ha declarado una función en la que se inicializan todas las variables en estado false, que serían los valores que tendría de fábrica el Sistema de Sellado hasta que empieza a funcionar y se conecta por primera vez.

Con toda esta explicación, queda desarrollada toda la programación del Sistema de Sellado para su simulación y ejecución por ordenador. Para su implementación en la Raspberry hay que modificar ciertos nodos, fundamentalmente de entrada y de salida ya que el funcionamiento de la máquina de estados es exactamente el mismo.

La conexión de la Raspberry Pi a los sensores mediante cableado no se va a realizar en el contenido de este trabajo, a cerca de lo cual se profundizará en apartados siguientes, pero igualmente se va a configurar como si así se hiciese. Para ello los nodos que se utilizarán tanto para la entrada como para la salida de valores serán los propios nodos que se pueden descargar a través del módulo de pines del propio Node-Red especialmente diseñados para la Raspberry. La descarga de este módulo proporciona estos nodos:

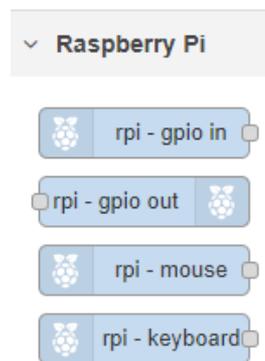


Ilustración 4-27. Nodos del módulo Raspberry Pi.

El primero y el segundo tiene la función de realizar la lectura de entrada y de salida respectivamente de los pines que se le asignen desde dentro de sus propiedades y los otros dos son para realizar entradas a través del ratón y del teclado que estén conectados a la Raspberry a través de las entradas de USB elegidas. En la ilustración 4-27 se muestran las propiedades que tiene el nodo de entrada *rpi-gpio in*, al principio está la tabla con todos los pines de la Raspberry indicándose también no solo su nombre sino sus configuraciones preestablecidas como son el pin de alimentación, de tierra y demás pines numerados. En el apartado BCM GPIO, al hacer clic sobre el pin seleccionado aparece el número equivalente de cada GPIO. Debajo de este, se indica si el pin seleccionado tiene alguna resistencia conectada para indicar el debounce y corregir así el ruido de esta. Justo abajo hay un cuadro el que se puede marcar que se lea el estado inicial del pin al ejecutar el *Deploy* o reiniciar Node-Red. Por último, hay un aviso del propio programa indicando que las salidas de los pines permitidos es únicamente digital con valores de 0 o 1.

Asignando a los pines de entrada, aquellos pines que se encuentran en verde en la tabla de la ilustración 4-28, se asignan los pines de entrada de los sensores teniendo la configuración mostrada en la ilustración posterior, la 4-51, que muestra la lectura del Scully, de la presión del circuito (cuya explicación se hará en un apartado posterior) y de las entradas del compartimento 1. La lectura de entradas del resto de compartimentos es igual únicamente cambiando las variables. En las funciones unidas a las salidas de los nodos de la Raspberry, transforman el valor leído a true o false que es con los valores con los que trabaja el programa desarrollado.

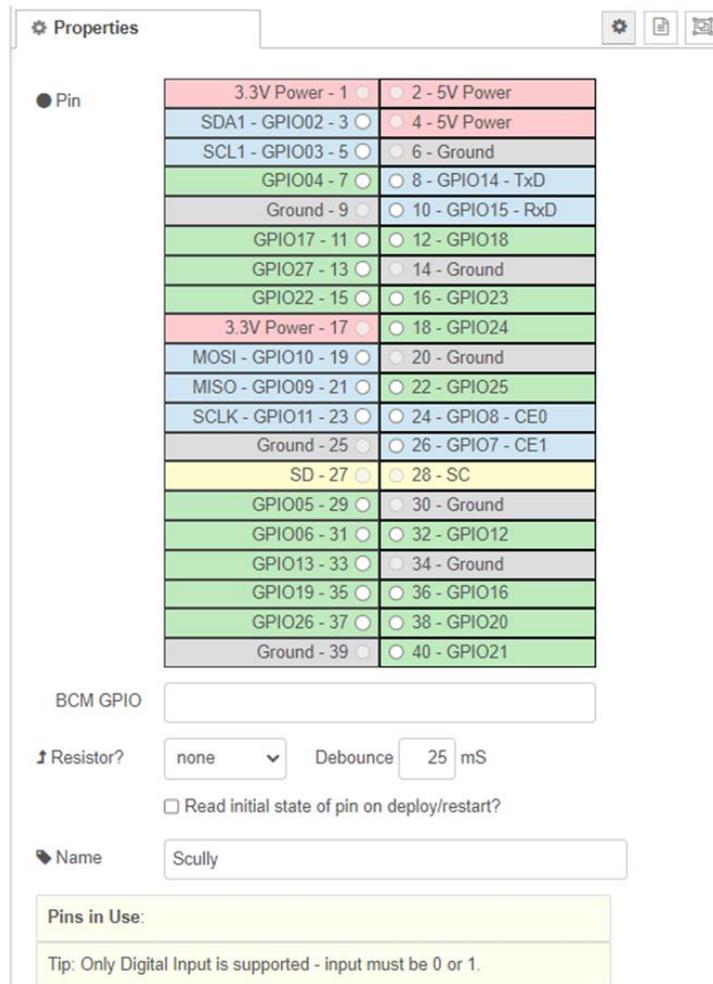


Ilustración 4-28. Propiedades nodo `rpi-gpio in`.

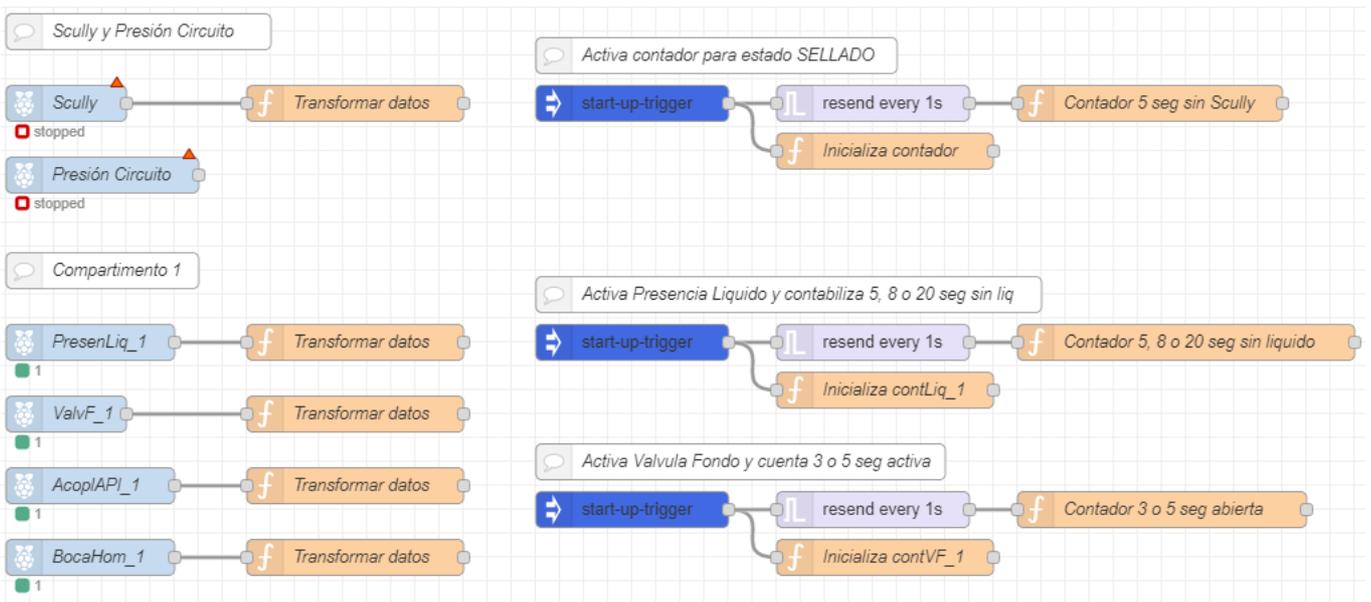


Ilustración 4-29. Flujo de lectura entradas generales y compartimento 1.

La diferencia entre los nodos que tiene el cuadrado verde con el 1 debajo y los que tienen el triángulo rojo junto con el cuadrado rojo y la palabra *stopped*, es que los primeros si tienen asignado un pin, aunque a ese pin no esté llegando nada, y los segundos no tienen ningún pin asignado por lo que el programa muestra esos avisos indicando que no hay posibilidad de hacer ninguna lectura porque no hay nada asignado. Se ha elegido esta configuración concreta para poder mostrar las diferentes situaciones que se pueden dar. Las funciones *Transformar datos* tienen el mismo formato de código entre sí, modificando únicamente la variable que tiene que asignar, haciendo que cuando ese sensor se halle encendido, la variable con la que trabaja el programa referente a ese sensor sea *true*, y por el contrario sea *false*. En la ilustración 7-21 del anexo se encuentra el código de la función *Transformar datos* para la variable *PresenLiq_1*, siendo equivalente para el resto de las variables.

Esas serían las modificaciones referentes a la lectura de entradas para la Raspberry. El otro flujo que variaría es el de las salidas, incluyendo el segundo nodo mostrado en la ilustración 4-27. Haciendo uso también de la advertencia por parte del programa de que las salidas digitales son 0 o 1, se programan las funciones que van antes de esos nodos de salida para poder acoplar bien los funcionamientos de los nodos. Dichas funciones son exactamente iguales a las de la ilustración 7-21 con el único cambio de que en vez de ser *true* o *false*, el *msg.payload* es 1 o 0, todo lo demás es idéntico. El flujo configurado se muestra a continuación:

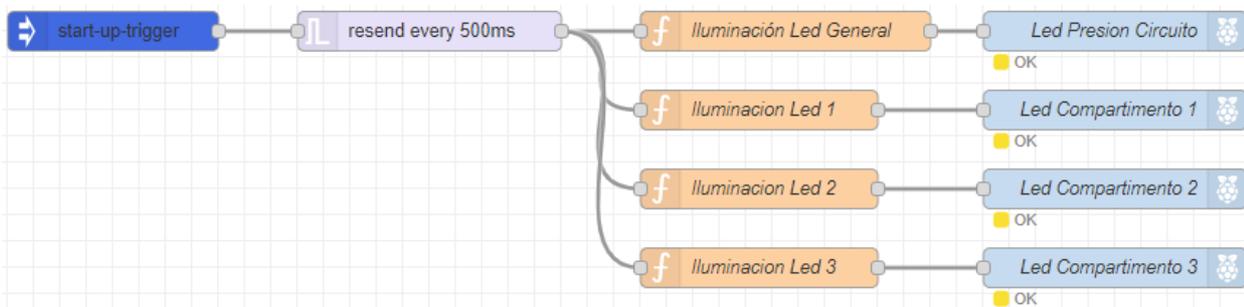


Ilustración 4-30. Flujo para la iluminación de los leds a través de la Raspberry.

La conexión principal realizada a la salida de la Raspberry es la del display que mostrará los estados de los compartimentos fundamentalmente. La pantalla utilizada es una pantalla LCD de 20x4 que se conecta a través de I2C impulsada por un controlador PCF8574.

Para conectarla a la Raspberry, se necesitan cuatro cables hembra hembra que conectaran los 4 pines de salida del módulo, rodeados en la ilustración 4-31 que son GND, VCC, SDA y SCL, con los pines 6, 4, 3 y 5 respectivamente que se muestran en la ilustración 4-28. Para conectarla a través de la programación, se necesita un nodo que hay que descargar desde la pestaña *Manage palettes* específico para este tipo de pantallas, dicho nodo es el nodo LCD-I2C. Para saber cómo es el formato de código que hay que darle para poder escribir no hay más que hacer clic en la pestaña de información del propio nodo dónde nos da un código de ejemplo para escribir en la parte central de la línea 1 y en la línea 4 alineando el texto esta vez a la derecha.

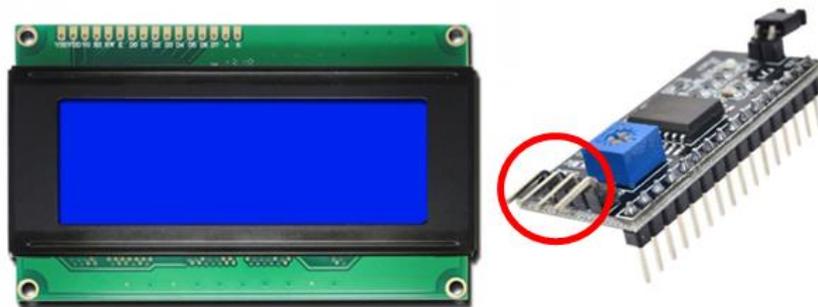


Ilustración 4-31. Display y módulo PCF8574.

Haciendo uso de este formato, se configura la función *Mostrar Estados en Display* que se muestra en el flujo de la ilustración 4-32, cuya aplicación es la de ver si el Scully se encuentra ON u OFF y la de asignar en variables auxiliares el valor de las variables globales *Estado_1*, *Estado_2* y *Estado_3* para mostrar uno en cada línea. Esto lo realizará el programa cada medio segundo, consiguiéndolo gracias al nodo *resend*. Este flujo estará en ejecución desde que se enciende la Raspberry.

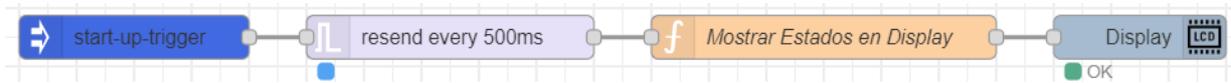


Ilustración 4-32. Flujo para mostrar en el display los estados.

4.4 Pruebas

En este apartado se van a mostrar las distintas puestas en marcha del programa y su visualización a través del dashboard. Hay que destacar que las conexiones de la Raspberry con los sensores de entrada no se ha podido realizar para comprobar el funcionamiento implementado en ella, para solucionarlo se ha programado también a través de un dashboard al igual para realizar los ensayos en el ordenador. Siendo ambos programas prácticamente idénticos, el dashboard que se va a ir representando será el de la propia Raspberry y se adjuntarán diferentes capturas de la salida de los datos por el display.

En las condiciones previas al funcionamiento, el dashboard se encuentra con todos los *switchs* desconectados, en off, la presión en cero, los leds se encuentran apagados, no hay ningún estado en ninguno de los compartimentos y aún no se muestra ni la fecha ni la hora, esto simularía el estado de fábrica previo al conexionado al camión cisterna, teniendo en el dashboard lo mostrado a continuación:

Sistema de Sellado			
General	Compartimento 1	Compartimento 2	Compartimento 3
Presión Circuito	Presencia Líquido	Presencia Líquido	Presencia Líquido
LED Circuito	Boca Hombre	Boca Hombre	Boca Hombre
Fecha Actual	Válvula Fondo	Válvula Fondo	Válvula Fondo
Fecha Actual	Acoplador API	Acoplador API	Acoplador API
Scully	LED 1	LED 2	LED 3
	Estado	Estado	Estado

Ilustración 4-33. Dashboard previo al conexionado de la Raspberry.

Posteriormente se procede con el conexionado al camión, alimentándose la Raspberry con los 5V e iniciándose así el programa por primera vez, mostrando la fecha y la hora y el estado de fábrica para cada uno de los compartimentos, siendo el estado de vacío para todos ellos.

Sistema de Sellado			
General	Compartimento 1	Compartimento 2	Compartimento 3
Presión Circuito	Presencia Líquido	Presencia Líquido	Presencia Líquido
LED Circuito	Boca Hombre	Boca Hombre	Boca Hombre
Fecha Actual	Válvula Fondo	Válvula Fondo	Válvula Fondo
Fecha Actual	Acoplador API	Acoplador API	Acoplador API
Scully	LED 1	LED 2	LED 3
	Estado	Estado	Estado
	VACIO_1	VACIO_2	VACIO_3

Ilustración 4-34. Dashboard recién conectado el sistema.

A partir de este momento el sistema empieza a funcionar con total normalidad, siguiendo el procedimiento anteriormente descrito en la máquina de estado. Se va a empezar suponiendo que el camión acaba de llegar a una isleta de carga, estando el Scully conectado y una presión en el circuito de 1 bar. En ese momento, el primer compartimento se encuentra vacío y se va a proceder a llenar, al igual que en el segundo compartimento, sin embargo, para el tercero se detectará combustible en el interior del compartimento.

Sistema de Sellado		Compartimento 1		Compartimento 2		Compartimento 3	
General							
Presión Circuito	1	Presencia Líquido	<input type="checkbox"/>	Presencia Líquido	<input type="checkbox"/>	Presencia Líquido	<input checked="" type="checkbox"/>
LED Circuito		Boca Hombre	<input type="checkbox"/>	Boca Hombre	<input type="checkbox"/>	Boca Hombre	<input type="checkbox"/>
Fecha Actual	23/06/2022	Válvula Fondo	<input checked="" type="checkbox"/>	Válvula Fondo	<input checked="" type="checkbox"/>	Válvula Fondo	<input type="checkbox"/>
Fecha Actual	11:01:42	Acoplador API	<input type="checkbox"/>	Acoplador API	<input type="checkbox"/>	Acoplador API	<input type="checkbox"/>
Scully	<input checked="" type="checkbox"/>	LED 1		LED 2		LED 3	
		Estado	COMPROBAD_1	Estado	COMPROBAD_2	Estado	CON_RESTO_3

Ilustración 4-35. Dashboard cuando se tiene los estados COMPROBAD, COMPROBAD y CON RESTO.

Para los dos primeros compartimentos, la válvula de fondo se ha abierto, y al pasar 3 segundos se ha pasado al estado COMPROBAD, el tercer compartimento se encuentra en estado CON RESTO ya que ha detectado la presencia de líquido en su interior.

El siguiente paso es el de purgar el tercer compartimento, abriendo el acoplador y la válvula de fondo hasta que se deja de detectar líquido, junto con el de conectar el acoplador API en uno de los dos primeros compartimentos para proceder a su llenado, el elegido será el primero, pasando a estado ACOPLADO e iniciando el parpadeo de su led correspondiente.

Sistema de Sellado		Compartimento 1		Compartimento 2		Compartimento 3	
General							
Presión Circuito	1	Presencia Líquido	<input type="checkbox"/>	Presencia Líquido	<input type="checkbox"/>	Presencia Líquido	<input type="checkbox"/>
LED Circuito		Boca Hombre	<input type="checkbox"/>	Boca Hombre	<input type="checkbox"/>	Boca Hombre	<input type="checkbox"/>
Fecha Actual	23/06/2022	Válvula Fondo	<input type="checkbox"/>	Válvula Fondo	<input checked="" type="checkbox"/>	Válvula Fondo	<input checked="" type="checkbox"/>
Fecha Actual	11:02:46	Acoplador API	<input checked="" type="checkbox"/>	Acoplador API	<input type="checkbox"/>	Acoplador API	<input checked="" type="checkbox"/>
Scully	<input checked="" type="checkbox"/>	LED 1		LED 2		LED 3	
		Estado	ACOPLADO_1	Estado	COMPROBAD_2	Estado	PURGADO_3

Ilustración 4-36. Dashboard con compartimentos en estado ACOPLADO, COMPROBAD y PURGADO.

En cuanto el primer compartimento pasa a detectar líquido, entra en estado LLENANDO, mientras que el tercer compartimento al cerrar la válvula de fondo y el acoplador API pasa a estado DISPONIBL permitiendo que se le conecte el acoplador únicamente para proceder a su llenado.



Ilustración 4-37. Dashboard mostrando los estados LLENANDO; COMPROBAD Y DISPONIBL.

Durante el proceso de llenado del primer compartimento, su led se encuentra parpadeando al igual que cuando el acoplador se encontraba acoplado pero sin empezar a llenar. Por eso en la imagen en estado LLENANDO se aprecia el led encendido mientras que en la anterior no se aprecia, manifestando así el parpadeo del led.

En cuanto se desconecta el API del primer compartimento, este pasa a estado LLENO, en el que permanecerá hasta realizar el sellado. En el segundo compartimento se ha conectado el acoplador ahora, estando en estado ACOPLADO.



Ilustración 4-38. Dashboard cuando se tienen los estados LLENO, ACOPLADO Y DISPONIBL.

Ahora se va a decidir no llenar el segundo compartimento sino el tercero, por lo que el acoplador pasará de uno a otro, siendo el estado del segundo DISPONIBL y el estado del tercero ACOPLADO.



Ilustración 4-39. Dashboard cuando los compartimentos están LLENO, DISPONIBL y ACOPLADO.

Una vez se ha llenado el compartimento número 3, se desconecta el acoplador API.



Ilustración 4-40. Dashboard con compartimentos en estado LLENO, DISPONIBL y LLENO.

Es ahora cuando se proceden a sellar los dos compartimentos que están llenos para poder así viajar hasta la estación de servicio de destino, por lo que se desconecta el Scully y pasados 5 segundos pasan a estado SELLADO.



Ilustración 4-41. Dashboard con dos compartimentos en SELLADO y uno DISPONIBL.

Tras llegar a la estación de servicio dónde se realizará la descarga, se procede a desprecintar los depósitos para su descarga, abriendo las válvulas de fondo y los acopladores.



Ilustración 4-42. Dashboard teniendo los estados DESPRECIN, DISPONIBL y DESPRECIN.

Cuando el sensor de presencia de líquido deja de detectar combustible, el estado cambia a DESCARGADO hasta que se cierran tanto los acopladores API como las válvulas.



Ilustración 4-43. Dashboard tras la finalización de la descarga de dos depósitos.

Tras cerrar los elementos comentados anteriormente, ambos compartimentos pasan a estado SIN PURGA, momento en el que se tendrá que comprobar la inexistencia de líquido en los depósitos descargados.

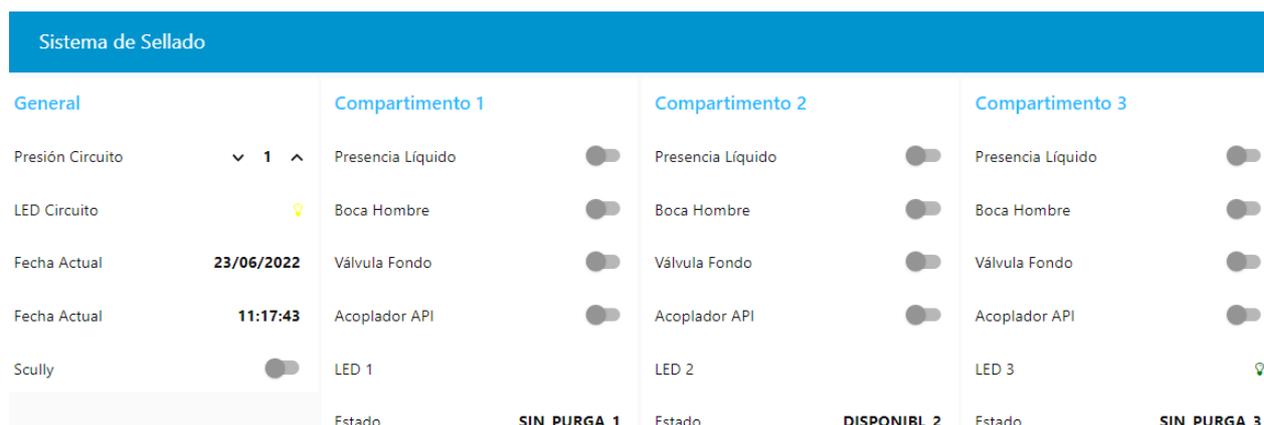


Ilustración 4-44. Dashboard mostrando los estados SIN PURGA, DISPONIBL Y SIN PURGA.

La operación de purga se realiza manteniendo abierta la válvula de fondo 5 segundos. Tras realizarlo, en el primer compartimento no se detecta líquido, por lo que pasa a estar en estado VACIO mientras que en el segundo sí, pasando a estado PURGAR.

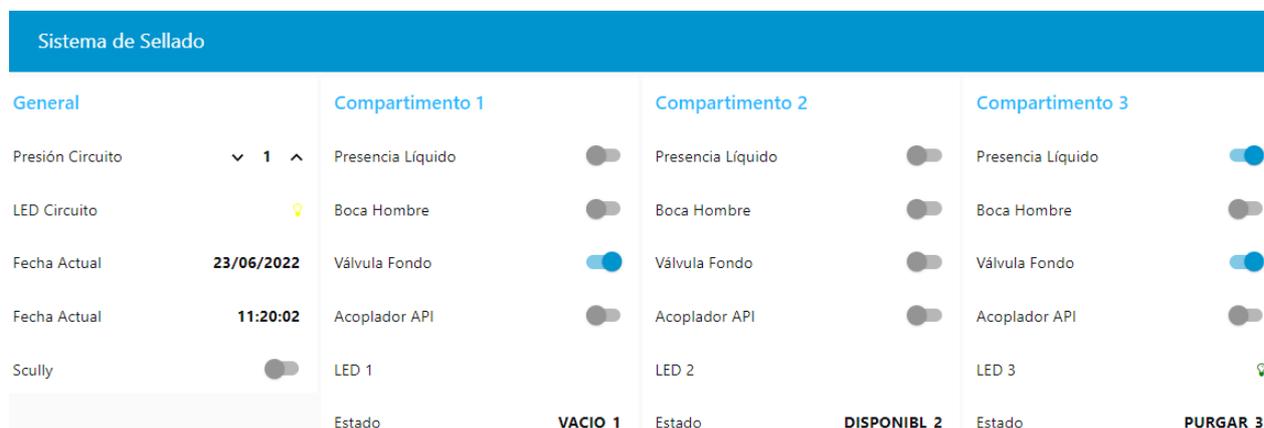


Ilustración 4-45. Dashboard tras la primera operación de purga.

Ahora, para purgar el depósito, además de abrir la válvula de fondo correspondiente, hay que abrir el acoplador API del compartimento en cuestión, en este caso el del tercero. Esto se mantiene así hasta que el sensor de presencia de líquido deje de detectar líquido durante al menos 5 segundos, pasando así a VACIO.

Sistema de Sellado		Compartimento 1		Compartimento 2		Compartimento 3	
General		Presencia Líquido	<input type="checkbox"/>	Presencia Líquido	<input type="checkbox"/>	Presencia Líquido	<input type="checkbox"/>
Presión Circuito	▼ 1 ▲	Boca Hombre	<input type="checkbox"/>	Boca Hombre	<input type="checkbox"/>	Boca Hombre	<input type="checkbox"/>
LED Circuito	💡	Válvula Fondo	<input type="checkbox"/>	Válvula Fondo	<input type="checkbox"/>	Válvula Fondo	<input type="checkbox"/>
Fecha Actual	23/06/2022	Acoplador API	<input type="checkbox"/>	Acoplador API	<input type="checkbox"/>	Acoplador API	<input type="checkbox"/>
Fecha Actual	11:21:19	LED 1	<input type="checkbox"/>	LED 2	<input type="checkbox"/>	LED 3	<input type="checkbox"/>
Scully	<input type="checkbox"/>	Estado	VACIO_1	Estado	DISPONIBL_2	Estado	VACIO_3

Ilustración 4-46. Dashboard tras haber purgado los depósitos necesarios.

A continuación se van a mostrar varias fotos tomadas del display que está conectado a la Raspberry mostrando los datos correspondientes a los estados de los compartimentos y del Scully.

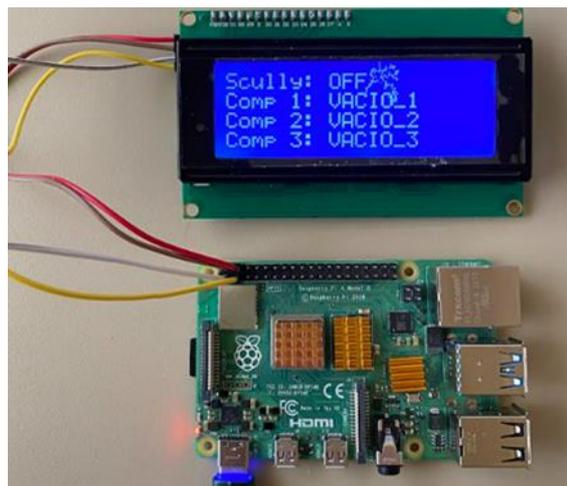


Ilustración 4-47. Conexión de display a Raspberry con estados iniciados.



Ilustración 4-48. Estados mostrados por el display.

4.5 Hardware adicional para implementación real

Para la puesta en práctica del proyecto contenido en este trabajo de fin de máster, y como avance al contenido que se ha desarrollado, para llevar a cabo la implementación real de la Raspberry como unidad central del Sistema de Sellado, se necesitarán una serie de elementos, fundamentalmente de protección y conexionado, para poder realizar las conexiones de manera correcta y segura.

- Convertidor DC 24 V a 5 V. Es un tipo de convertidor de potencia que transforma corriente continua de un nivel de tensión a otro mediante el almacenamiento temporal de la energía de entrada, entregándola luego en la salida al voltaje deseado. Se utiliza para alimentar la Raspberry con los 5 V que necesita, ya que el camión se alimenta a 24 V y eso es una tensión excesivamente alta para la nueva unidad central.



Ilustración 4-49. Convertidor DC 24V a 5V [25].

- Relé de estado sólido. Es un dispositivo interruptor electrónico que conmuta el paso de la electricidad cuando una pequeña carga de corriente es aplicada a sus terminales de control. Consisten en un sensor que responde a una señal de control, un interruptor electrónico de estado sólido que conmuta el circuito y un mecanismo de acoplamiento a partir de la señal de control que activa este interruptor sin partes mecánicas. Como ventajas que argumentan el uso de este tipo de relés está que al no tener partes mecánicas no le afecta la humedad, haciendo su vida útil mayor, no le afecta el desgaste al realizar conmutaciones de altas corrientes y tampoco producen interferencias electromagnéticas, aunque por contra, tiene pérdidas de calor y cuando se averían suelen quedarse activados, al contrario de los mecánicos.



Ilustración 4-50. Relé sólido de 4 canales [26].

- Caja antideflagrante. Son unas cajas de conexiones con o sin terminales para la instalación de equipos eléctricos tales como interruptores, señales, equipos de control remoto o transformadores. Están fabricadas para ser instaladas en las zonas clasificadas como 1, 2, 21 y 22, en las que hay peligro de explosión o áreas donde hay polvo combustible o gas, y especialmente diseñadas para los sectores donde las instalaciones requieren seguridad adicional. Son ideales para interconectar cables y acomodar una gran variedad de equipos e instrumentos, que en este caso sería la Raspberry. El equipo eléctrico está encerrado en el interior de una envolvente capaz de resistir una posible explosión y no transmitir la inflamación al ambiente. Además permite el montaje de actuadores, su fabricación a

medida y la posibilidad de varios modelos con ventana, lo que puede resultar especialmente útil para acoplar el display.



Ilustración 4-51. Caja antideflagrante con ventana externa [27].

- Barreras de seguridad intrínseca. Son dispositivos de protección de equipos eléctricos que limitan la energía que se suministra a un circuito para que no se produzca una posible chispa de manera que nunca se alcance el valor mínimo de la energía de ignición. Además, la temperatura de la superficie del equipo tampoco alcanzará la temperatura de ignición del combustible. Esto se consigue gracias a la limitación del voltaje y de la corriente interna tomando en consideración los posibles defectos que puedan producirse. Su uso fundamentalmente recae en separar la zona de control de la zona clasificada como peligrosa por lo que aplicado al Sistema de Sellado, estas barreras de seguridad intrínseca harían falta para separar todos los sensores que están en contacto con el combustible o en el propio camión cisterna estando conectadas a los propios sensores, a la entrada digital de la Raspberry y a la alimentación del camión de 24 V.



Ilustración 4-52. Barrera de aislamiento intrínseco [28].

5 CONCLUSIONES

Tras hacer el desarrollo de los elementos necesarios para poder llevar a cabo la realización de la unidad central del Sistema de Sellado así como todo el funcionamiento lógico que tiene detrás, se han llegado a una serie de conclusiones que son las siguientes:

- La finalidad de este trabajo era la de poder desglosar el funcionamiento que tiene la Main Unit del Sistema de Sellado aportada por el anterior proveedor, permitiendo realizar una programación exclusiva y semejante para este sistema concreto. Gracias a la utilización de un servidor web como es Node-Red se ha podido hacer ese desarrollo teniendo como base el funcionamiento de un único compartimento, ya que este se puede replicar para tantos como sea necesario, comprobando así que la ejecución es correcta y adecuada.
- El uso de Node-Red tiene varias ventajas sobre otros posibles programas de trabajo. La principal recae en la facilidad de operación del programa debido a que es una herramienta de desarrollo basada en diagramas de flujo para realizar una programación visual, permitiendo al usuario su manejo sin tener excesivos conocimientos sobre programación. Otra ventaja es la infinidad de nodos que Node-Red pone a disposición del usuario ya que a través de la pestaña de *Manage palletes* se pueden instalar todos y cada uno de los nodos que se requieran para poder hacer una aplicación específica, asimismo la base para programar los códigos de las funciones es JavaScript, uno de los lenguajes de programación más utilizados y asentados en la actualidad. Por último pero no por ello menos importante, al ser Node-Red un software libre que ofrece la posibilidad de compartir y modificar el código fuente, se promueve la colaboración entre usuarios y permite la edición y configuración del flujo realizado con la única necesidad de tener acceso a internet, hecho que hoy en día es posible prácticamente en todos los lugares de mundo. Con esto se permite la solución de fallos o corrección de posibles errores de forma remota teniendo así el mejor funcionamiento posible del programa y su instalación y ejecución en un amplio rango de dispositivos y entornos.
- Respecto al apartado económico, por la necesidad de privacidad por parte de la empresa no se van a citar cifras exactas pero si aproximadas. Para la implementación del proyecto empezado a desarrollar en este trabajo se hará uso de la Raspberry Pi 4 introducida en capítulos anteriores, de un display como el también comentado, una caja de pilotos led, una caja antideflagrante que contenga a la Raspberry y permita el manejo del display, el convertidor para la alimentación, un relé con tantos canales como sensores haya o varios con tantos canales como sean necesarios, que para el caso de 3 compartimentos serían 16, al igual que el número de barreras de seguridad intrínseca para los sensores. La suma de todos estos componentes, con los precios de oferta solicitados a los proveedores de la empresa, oscila en torno a los tres mil euros. Este precio es única y exclusivamente de los componentes enumerados por lo que habría que añadir la mano de obra referente a la programación de Node-Red y al montaje de todo el sistema previo a la instalación, ya que el coste de la instalación es un cobro que se realiza por separado. El conjunto de interfases y unidad central utilizada hasta ahora tiene un coste en torno a los cuatro mil quinientos euros, por lo que el ahorro aproximado sería de un 30%. Este ahorro permitiría a MONTREL S.A. ofrecer el Sistema de Sellado a sus clientes con un coste proporcionalmente menor dando así un impulso a las ventas de este sistema.
- Como mejoras al trabajo aquí desarrollado se contempla la optimización del programa desarrollado en Node-Red para poder adaptarlo fácilmente en función del número de compartimentos ya que el que se ha realizado en este trabajo está adaptado a 3 compartimentos, la investigación acerca de los fallos que pueden producirse en cada uno de los sensores pudiendo identificarlos para categorizarlos y diferenciar tanto sus soluciones como su repercusión en el programa, y la utilización de otro servidor web para la escritura de las anomalías permitiendo así su acceso desde la central en cualquier

momento. Todas estas mejoras permitirían tener un programa totalmente depurado de forma que su comercialización estaría totalmente adaptada a cada una de las posibilidades que necesitara el cliente.

De esta forma y tras todas las conclusiones obtenidas gracias a la realización de este trabajo, se puede concluir que el desarrollo de la unidad central del Sistema de Sellado se puede realizar en las instalaciones de MONTREL S.A. obteniendo así un ahorro de en torno a un 20% que no solo se vería repercutido en el coste que tiene para la empresa, sino que esto conllevaría una disminución del precio de venta permitiendo ahorrar también a los clientes y pudiendo impulsar el Sistema de Sellado en el mercado.

6 BIBLIOGRAFÍA

- [1] “Lo que no sabías sobre TIP y los camiones cisterna - TIP España.” <https://www.tipeurope.es/knowledge-and-news/noticias/lo-que-no-sabias-sobre-tip-y-los-caminones-cisterna> (accessed Apr. 27, 2022).
- [2] “Proyectos petroleros: 6 etapas para su ejecución - EALDE.” <https://www.ealde.es/proyectos-petroleros-etapas-ejecucion/> (accessed May 10, 2022).
- [3] “Las petroleras incrementan su demanda de platino por la apertura de nuevas refinerías - Oroinformación.” <https://oroinformacion.com/la-industria-del-petroleo-sigue-incrementando-su-demanda-de-platino-por-la-apertura-de-nuevas-refinerias/> (accessed May 10, 2022).
- [4] “Early Tank Trucks.” <http://www.petroleumhistory.org/OilHistory/pages/trucks/earlytrucks.html> (accessed May 10, 2022).
- [5] “Consumo de carburantes, gasolina y gasóleo en España, estadísticas y datos.” <https://www.epdata.es/datos/consumo-carburantes-gasolina-gasoleo-espana-estadisticas-datos/326/espana/106> (accessed May 10, 2022).
- [6] P. Delgado, C. Tutor, and M. T. Dep, “Trabajo Fin de Grado Grado en Ingeniería en Tecnologías Industriales Desarrollo de Mejoras en Sistema de Sellado Electrónico de Camiones Cisterna,” 2014.
- [7] “Cabezas tractoras VOLVO FH 16 usados y nuevos en venta, comprar cabezas tractoras VOLVO FH 16 de segunda mano en Truck1 es.” <https://www.truck1.es/cabezas-tractoras/volvo/fh-16> (accessed May 25, 2022).
- [8] “ACOPLAMIENTO SECO”.
- [9] “Válvulas de carga y descarga API, válvula neumática Interlock y acoples| Bombas Metal, S.L.” <http://www.new.bombasmetal.es/piusi-gasoil/equipamiento-cisternas/valvulas-de-fondo-ficha.html> (accessed May 25, 2022).
- [10] “Manhole cover DN500 PAF – airportcomponents.com.” <https://www.airportcomponents.com/en/product/manhole-cover-dn500-paf/> (accessed May 25, 2022).
- [11] “Cosas que por seguridad no debe hacer en una estación de servicio - Global Estaciones de Servicio.” <https://globalestacionesdeservicio.com/cosas-que-por-seguridad-no-debe-hacer-en-una-estacion-de-servicio/> (accessed May 25, 2022).
- [12] “De oro negro a oro verde: Singapur cambiará de matriz petrolera a centro ecológico.” <https://www.mascontainer.com/de-oro-negro-a-oro-verde-singapur-cambiara-de-matriz-petrolera/> (accessed May 25, 2022).
- [13] “Presentacion Sistema Sellado Montrel.”
- [14] “¿Qué es el ADR en el transporte de mercancías peligrosas? – Blog sobre logística y transporte | Logismarket.” <https://blog.logismarket.es/que-es-adr-transporte-mercancias-peligrosas/> (accessed May 25, 2022).
- [15] “Asociación Electrotécnica Española.” <http://www.electrotecnia.org/informacion> (accessed May 25, 2022).
- [16] “▷ Qué son las normas UNE EN ISO | CTMA Consultores.” <https://ctmaconsultores.com/normas-une-iso/> (accessed May 25, 2022).
- [17] F. J. Rodríguez Iglesias, “Nuevo Sistema de Sellado Cepsa,” 2006

- [18] “Directiva ATEX y ámbitos de aplicación en atmósferas explosivas.” <https://www.promam.es/directiva-atex-y-aplicacion/> (accessed May 25, 2022).
- [19] “FMC MultiFlow - Metered Delivery - MA-TEKNIK.” <https://ma-teknik.dk/shop/fmc-multiflow-metered-delivery/> (accessed May 25, 2022).
- [20] “Presentacion Sistema Sellado CEPSA.”
- [21] “Sistemas de conexión a tierra, prevención de rebose e identificación de vehículos para la industria química - Scully Signal.” <https://scully.com/mercados/sistemas-de-conexion-a-tierra-prevencion-de-rebose-e-identificacion-de-vehiculos-para-la-industria-quimica/?lang=es> (accessed May 25, 2022).
- [22] “Node-RED.” <http://127.0.0.1:1880/#flow/178d11e115851b4f> (accessed Jun. 13, 2022).
- [23] “Node.js.” <https://nodejs.org/es/> (accessed Jun. 22, 2022).
- [24] “Compra una Raspberry Pi 4 Modelo B – Raspberry Pi.” <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/> (accessed Jun. 13, 2022).
- [25] “DTJ1548S05 | Convertidor dc-dc 15W, Salida 5V dc, 3A, 0.005 No No | RS Components.” [https://es.rs-online.com/web/p/convertidores-dc-dc/1935876?cm_mmc=ES-PLA-DS3A-_-google-_-CSS_ES_ES_Fuentes_de_alimentacion_y_transformadores_Whoop-_- \(ES:Whoop!\)+Convertidores+DC-DC-_-1935876&matchtype=&pla-339639780897&gclid=CjwKCAjwwdWVBhA4EiwAjcYJEMAP1fiNCdJU7V3ObV3yr5TB56COTOh9oxLYg0ABYqCBSqg-gS4Z_RoCGBQAvD_BwE&gclsrc=aw.ds](https://es.rs-online.com/web/p/convertidores-dc-dc/1935876?cm_mmc=ES-PLA-DS3A-_-google-_-CSS_ES_ES_Fuentes_de_alimentacion_y_transformadores_Whoop-_- (ES:Whoop!)+Convertidores+DC-DC-_-1935876&matchtype=&pla-339639780897&gclid=CjwKCAjwwdWVBhA4EiwAjcYJEMAP1fiNCdJU7V3ObV3yr5TB56COTOh9oxLYg0ABYqCBSqg-gS4Z_RoCGBQAvD_BwE&gclsrc=aw.ds) (accessed Jun. 24, 2022).
- [26] “Relé sólido de 4 canales.” <https://www.e-ika.com/rele-solido-de-4-canales> (accessed Jun. 24, 2022).
- [27] “Caja Antideflagrante Ex d Ventana EJB · Atex Delvalle.” <https://www.atexdelvalle.com/es/cajas-antideflagrantes-ex-atex/cajas-de-conexiones-con-ventana-rectangular> (accessed Jun. 24, 2022).
- [28] “Quick Select Product Guide Process Interfaces Interface Technology Remote I/O Systems Fieldbus Infrastructure”, Accessed: Jun. 24, 2022. [Online]. Available: www.pepperl-fuchs.com

7 ANEXO

7.1 Códigos funciones compartimento 1

```
var cont = global.get("contador");

if (global.get("Scully")==false && (global.get("Estado_1")==LLENANDO_1 || global.get("Estado_1")==SELLADO_1 ||
global.get("Estado_1")==LLENO_1 || global.get("Estado_2")==LLENANDO_2 || global.get("Estado_2")==SELLADO_2 ||
global.get("Estado_2")==LLENO_2 || global.get("Estado_3")==LLENANDO_3 || global.get("Estado_3")==SELLADO_3 ||
global.get("Estado_3")==LLENO_3))
{
    cont=cont+1;
    global.set("contador",cont);
    msg.payload=global.get("contador");
    if (global.get("contador")>=5)
    {
        global.set("NoScully",true);
    }
}
else
{
    global.set("contador", 0);
    msg.payload=global.get("contador");
    global.set("NoScully",false);
}
return msg;
```

Ilustración 7-1-. Programación nodo *function* denominado *Contador 5 seg sin Scully*.



Ilustración 7-2. Código y propiedades de la función *ACTUALIZA ESTADO*.

```
if(global.get("Scully") == true)
{
    if (global.get("AcoplAPI_1") == false && global.get("PulsVF_1") == true)
    {
        global.set ("Estado_1","PULSADO_1");
    }
    if (global.get("PresenLiq_1") == true)
    {
        global.set ("Estado_1","CON_RESTO_1");
    }
}
msg.payload_1=global.get("Estado_1")
return msg;
```

Ilustración 7-3. Código de la función *VACIO_1*.

```

if(global.get("Scully") == true)
{
  if (global.get("AcoplAPI_1") == true && global.get("PresenLiq_1") == false)
  {
    global.set ("Estado_1","ACOPLADO_1");
  }
  if (global.get("PresenLiq_1") == true)
  {
    global.set ("Estado_1","CON_RESTO_1");
  }
  if (global.get("AcoplAPI_1") == false && global.get("NoLiq_1") == true)
  {
    global.set ("Estado_1","COMPROBAD_1");
  }
}
msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-4. Código de la función *PULSADO_1*.

```

if(global.get("Scully") == true)
{
  if (global.get("AcoplAPI_1") == true )
  {
    global.set ("Estado_1","ACOPLADO_1");
  }
}
msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-5. Código de la función *COMPROBAD_1*.

```

if(global.get("Scully") == true)
{
  if (global.get("PresenLiq_1") == true && global.get("AcoplAPI_1") == true)
  {
    global.set ("Estado_1","LLENANDO_1");
    if (global.get("Estado_1") == "CON_RESTO_1" || global.get("Estado_2") == "CON_RESTO_2"
    || global.get("Estado_3") == "CON_RESTO_3")
    {
      global.set ("ANOMALIA_1","SI");
    }// no se puede llenar habiendo algun compartimento en estado con resto
  }
  if (global.get("PresenLiq_1") == false && global.get("AcoplAPI_1") == false
  || global.get("ValvF_1") == false)
  {
    global.set ("Estado_1","DISPONIBL_1");
  }
}
msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-6. Código de la función *ACOPLADO_1*.

```

var aux = "ANOMALIA EN EL COMPARTIMENTO 1"
if (global.get ("ANOMALIA_1")== "SI")
{
    payload={"Suceso":aux," Fecha":(global.get("Fecha"))," Hora":(global.get("Hora"))};
    global.set("ANOMALIA_1", "NO");
    msg.payload=payload;
}
return msg;

```

Ilustración 7-7. Código de la función *Registrar anomalía 1*.

```

if(global.get("Scully") == true)
{
    if (global.get("PresenLiq_1") == true && global.get("AcoplAPI_1") == false
        && global.get("BocaHom_1") == false && global.get("ValvF_1") == false )
    {
        global.set ("Estado_1","LLENADO_1");
    }
    if (global.get("PresenLiq_1") == false && global.get("AcoplAPI_1") == true)
    {
        global.set ("Estado_1","ACOPLADO_1");
        global.set ("ANOMALIA_1","SI");
    }
} //no se puede sellar directamente de llenando
}

if (global.get("NoScully")==true && global.get("PresenLiq_1") == true &&
    global.get("AcoplAPI_1") == false && global.get("BocaHom_1") == false &&
    global.get("ValvF_1") == false && (global.get("Estado_1") != "CON_RESTO_1" &&
    global.get("Estado_2") != "CON_RESTO_2" && global.get("Estado_3") != "CON_RESTO_3"))
{
    global.set ("Estado_1","SELLADO_1");
    global.set ("ANOMALIA_1","SI");
} //
msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-8. Código de la función *LLENANDO_1*.

```

if(global.get("Scully") == true)
{
    if (global.get("PresenLiq_1") == true && global.get("AcoplAPI_1") == true)
    {
        global.set ("Estado_1","LLENANDO_1");
        global.set ("ANOMALIA_1","SI");
    }
}

if (global.get("NoScully")==true && global.get("PresenLiq_1") == true &&
    global.get("AcoplAPI_1") == false && global.get("BocaHom_1") == false &&
    global.get("ValvF_1") == false && (global.get("Estado_1") != "CON_RESTO_1" &&
    global.get("Estado_2") != "CON_RESTO_2" && global.get("Estado_3") != "CON_RESTO_3"))
{
    global.set ("Estado_1","SELLADO_1");
} //
msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-9. Código de la función *LLENO_1*.

```

if(global.get("Scully") == true)
{
    if (global.get("PresenLiq_1") == true && global.get("AcoplAPI_1") == true)
    {
        global.set ("Estado_1","LLENANDO_1");
    }
}

if (global.get("Scully") == false && (global.get("AcoplAPI_1") == true ||
global.get("BocaHom_1") == true || global.get("ValvF_1") == true))
{
    global.set ("Estado_1","DESPRECIN_1");
    global.set ("ANOMALIA_1","SI");
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-10. Código de la función *SELLADO_1*.

```

if (global.get("NoLiq_1") == true && global.get("Scully") == false &&
global.get("AcoplAPI_1") == true && global.get("ValvF_1") == true)
{
    global.set ("Estado_1","DESCARGAD_1");
    global.set ("ANOMALIA_1","SI");
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-11. Código de la función *DESPRECIN_1*.

```

if (global.get("Scully") == false && global.get("PresenLiq_1") == false &&
global.get("AcoplAPI_1") == false && global.get("ValvF_1") == false)
{
    global.set ("Estado_1","SIN_PURGA_1");
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-12. Código de la función *DESCARGAD_1*.

```

if (global.get("Scully") == false && global.get("PulsVF_1") == true)
{
    if (global.get("PresenLiq_1") == false)
    {
        global.set ("Estado_1","VACIO_1");
    }
    if (global.get("PresenLiq_1") == true)
    {
        global.set ("Estado_1","PURGAR_1");
    }
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-13. Código de la función *SIN_PURGA_1*.

```

if (global.get("Scully") == false && global.get("AcoplAPI_1") == true &&
    global.get("ValvF_1")==true && global.get("NoLiq_1")==true)
{
    global.set ("Estado_1","VACIO_1");
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-14. Código de la función *PURGAR_1*.

```

if (global.get("Scully") == true && global.get("AcoplAPI_1") == true &&
    global.get("ValvF_1")==true && global.get("NoLiq_1")==true)
{
    global.set ("Estado_1","PURGADO_1");
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-15. Código de la función *CON_RESTO_1*.

```

if (global.get("Scully") == true && global.get("AcoplAPI_1") == false &&
    global.get("ValvF_1")==false)
{
    global.set ("Estado_1","DISPONIBL_1");
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-16. Código de la función *PURGADO_1*.

```

if (global.get("Scully") == true && global.get("AcoplAPI_1") == true)
{
    global.set ("Estado_1","ACOPLADO_1");
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-17. Código de la función *DISPONIBL_1*.

```

if (global.get("Estado_1")!="FALLO_1")
{
    global.set ("Estado_1","SIN_PURGA_1");
}

msg.payload_1=global.get("Estado_1")
return msg;

```

Ilustración 7-18. Código de la función *FALLO_1*.

```

if(global.get("PN_CirPrin") > 1.5)
{
    msg.payload= true;
}
else
{
    msg.payload= false;
}
return msg;

```

Ilustración 7-19. Código de *Iluminación Led General*.

```

var est_led_1;

if (global.get("PN_CirPrin") > 1.5)
{
  if (global.get("Estado_1") == "VACIO_1" || global.get("Estado_1") == "COMPROBAD_1" ||
      global.get("Estado_1") == "PURGADO_1" || global.get("Estado_1") == "DISPONIBL_1")
      {
        | msg.payload= false;
      }
  if (global.get("Estado_1") == "SELLADO_1")
      {
        | msg.payload= true;
      }
  if (global.get("Estado_1") == "CON_RESTO_1" || global.get("Estado_1") == "ACOPLADO_1" ||
      global.get("Estado_1") == "LLENANDO_1" || global.get("Estado_1") == "LLENO_1" ||
      global.get("Estado_1") == "PURGAR_1" || global.get("Estado_1") == "SIN_PURGAR_1" ||
      global.get("Estado_1") == "DESCARGAD_1" || global.get("Estado_1") == "DESPRECIN_1" ||
      global.get("Estado_1") == "FALLO_1")
      {
        est_led_1=(!global.get("Estado_Led_1"));
        global.set("Estado_Led_1", est_led_1);
        msg.payload=global.get("Estado_Led_1");
      }
}
else
{
  msg.payload= false; // si no hay presion todos los leds apagados
}
return msg;

```

Ilustración 7-20. Código de la función *Iluminación Led 1*.

```

if(msg.payload)
{
  global.set("PresenLiq_1",true);
}
else
{
  global.set("PresenLiq_1",false);
}

return msg;

```

Ilustración 7-21. Código función *Transformar datos* para la variable *PresenLiq_1*.

```

var Scu = global.get("Scully");
if (global.get("Scully")==true)
{
  var Scu= "ON";
}
else
{
  var Scu= "OFF";
}

var Est1 = global.get("Estado_1");
var Est2 = global.get("Estado_2");
var Est3 = global.get("Estado_3");

msg.payload = [
  {
    "clear": false,
    "text": "Scully: " + Scu,
    "alignment": "left"
  },
  {
    "clear": false,
    "text": "Comp 1: " + Est1,
    "alignment": "left"
  },
  {
    "clear": false,
    "text": "Comp 2: " + Est2,
    "alignment": "left"
  },
  {
    "clear": false,
    "text": "Comp 3: " + Est3,
    "alignment": "left"
  }
]
return msg;

```

Ilustración 7-22. Código función *Mostrar Estados en Display*.

7.2 Diagramas de flujo

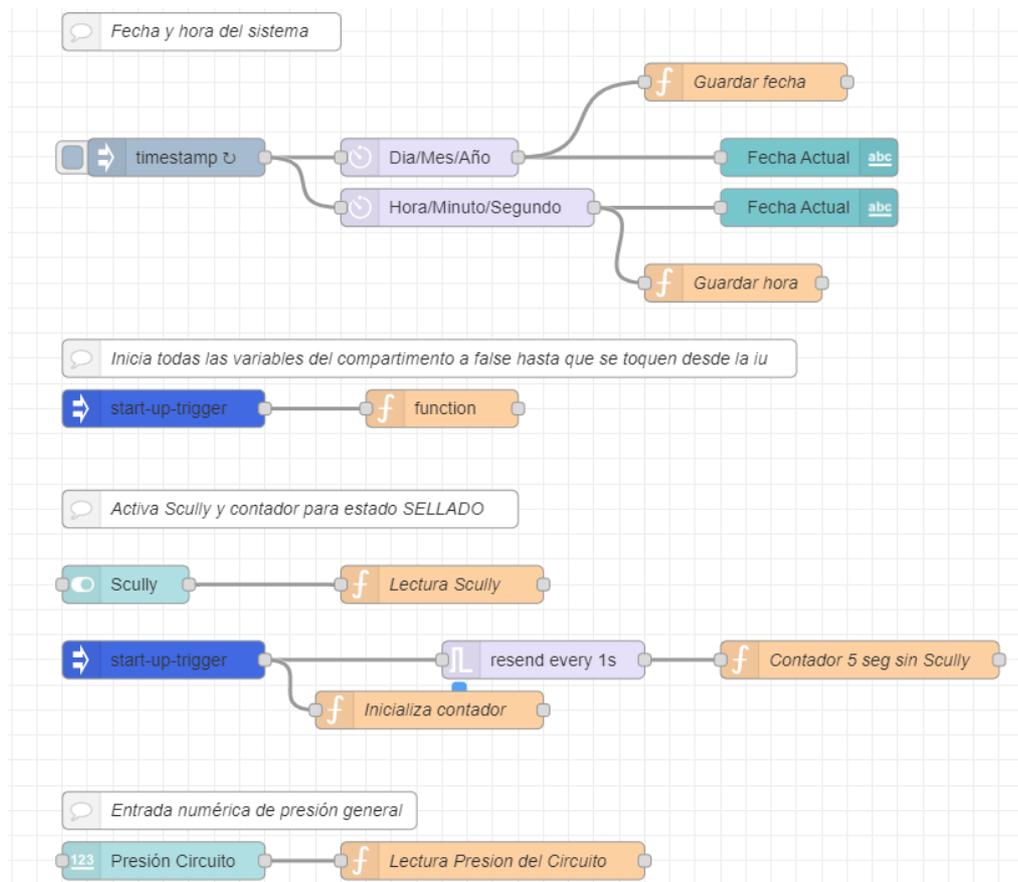


Ilustración 7-23. Flujo Entradas Generales.

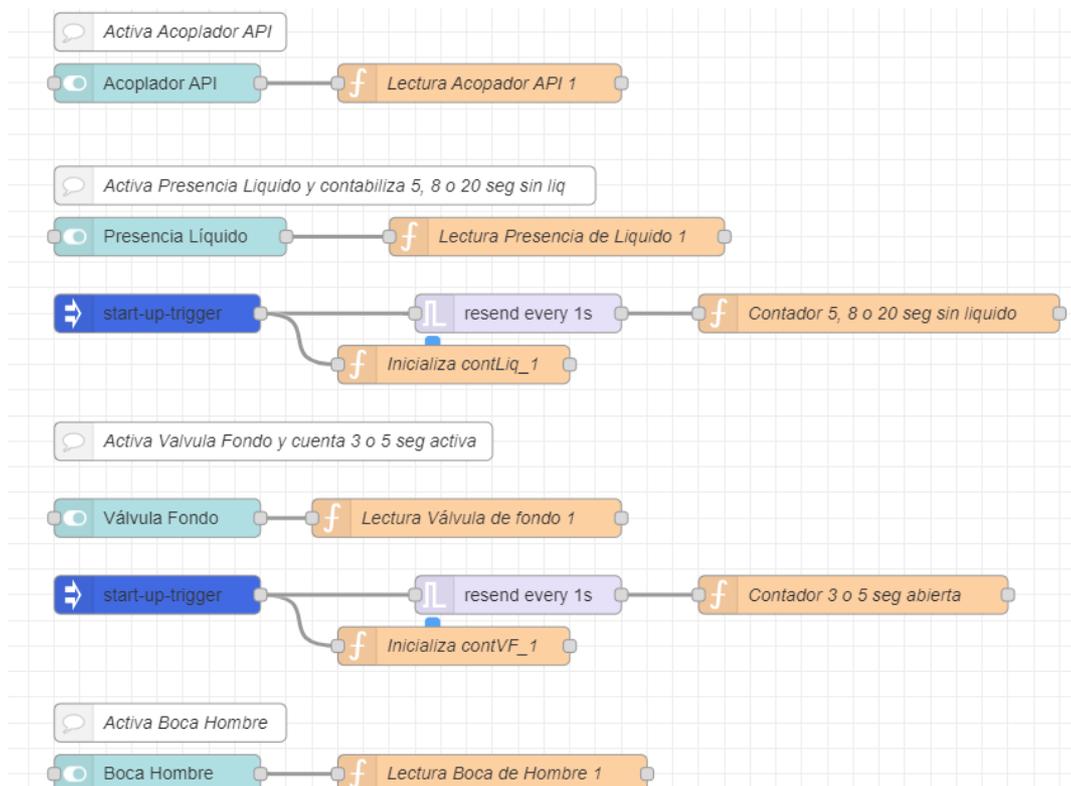


Ilustración 7-24. Flujo Entradas Comp 1.

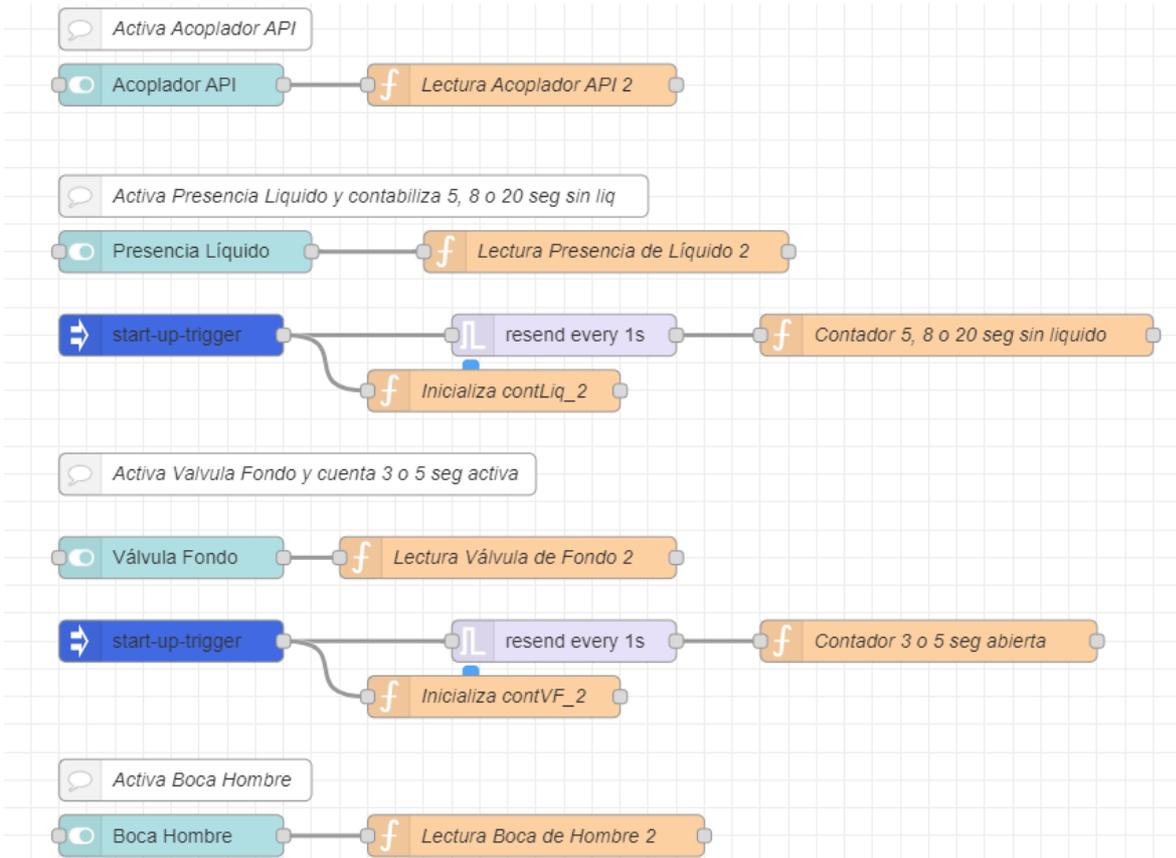


Ilustración 7-25. Flujo Entradas Comp 2.

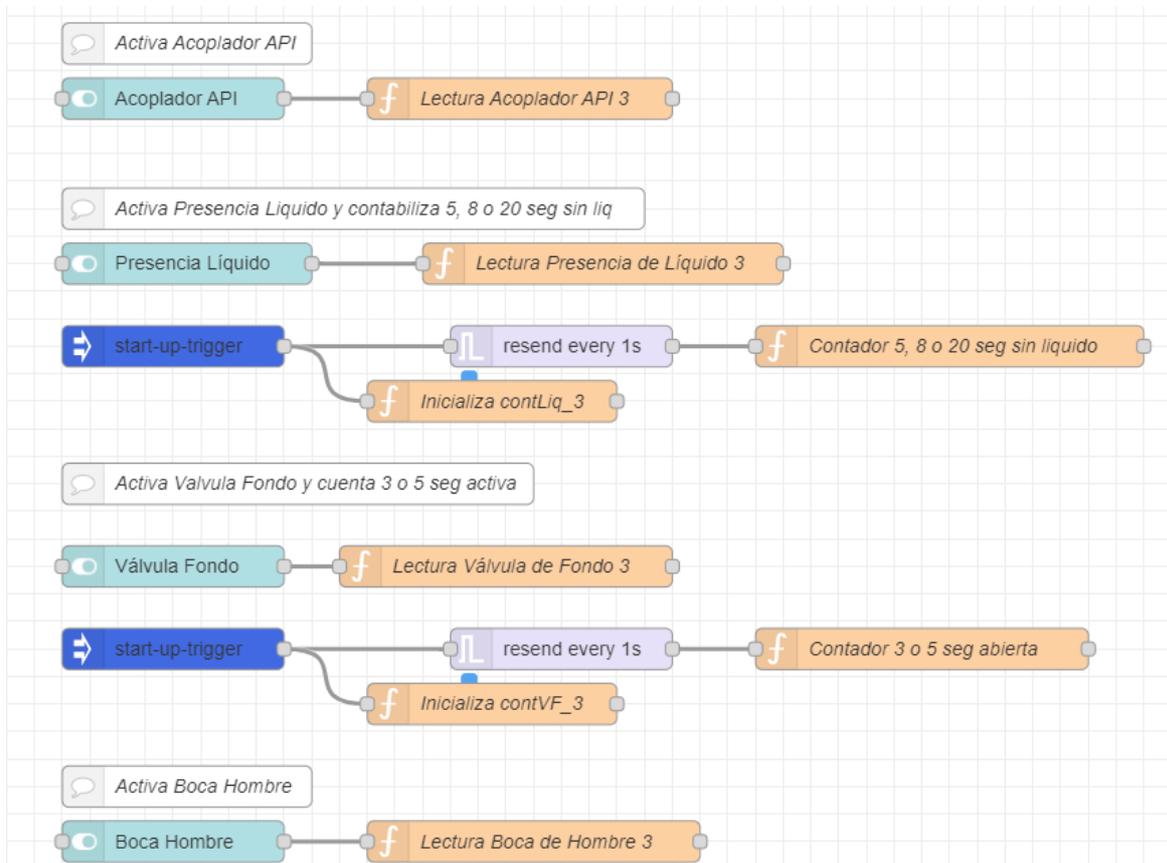


Ilustración 7-26. Flujo Entradas Comp 3.

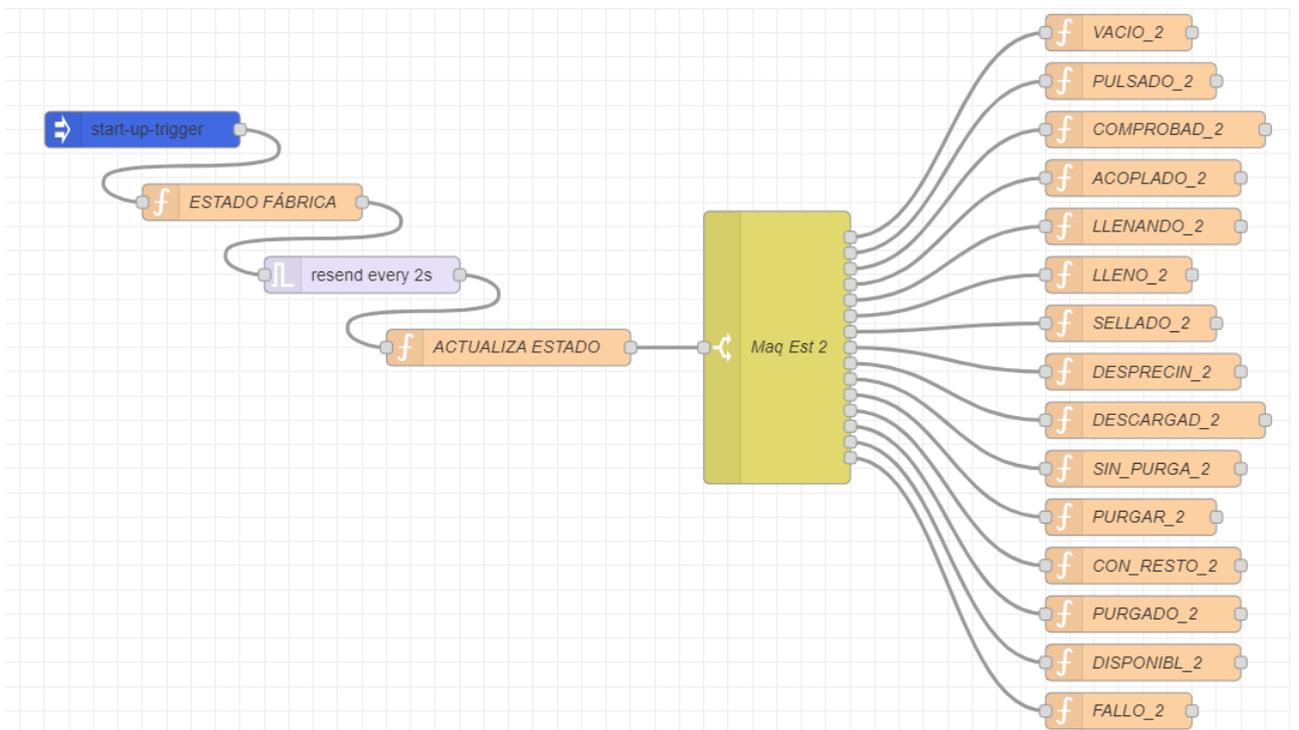


Ilustración 7-27. Flujo Máquina Estados Comp 2.

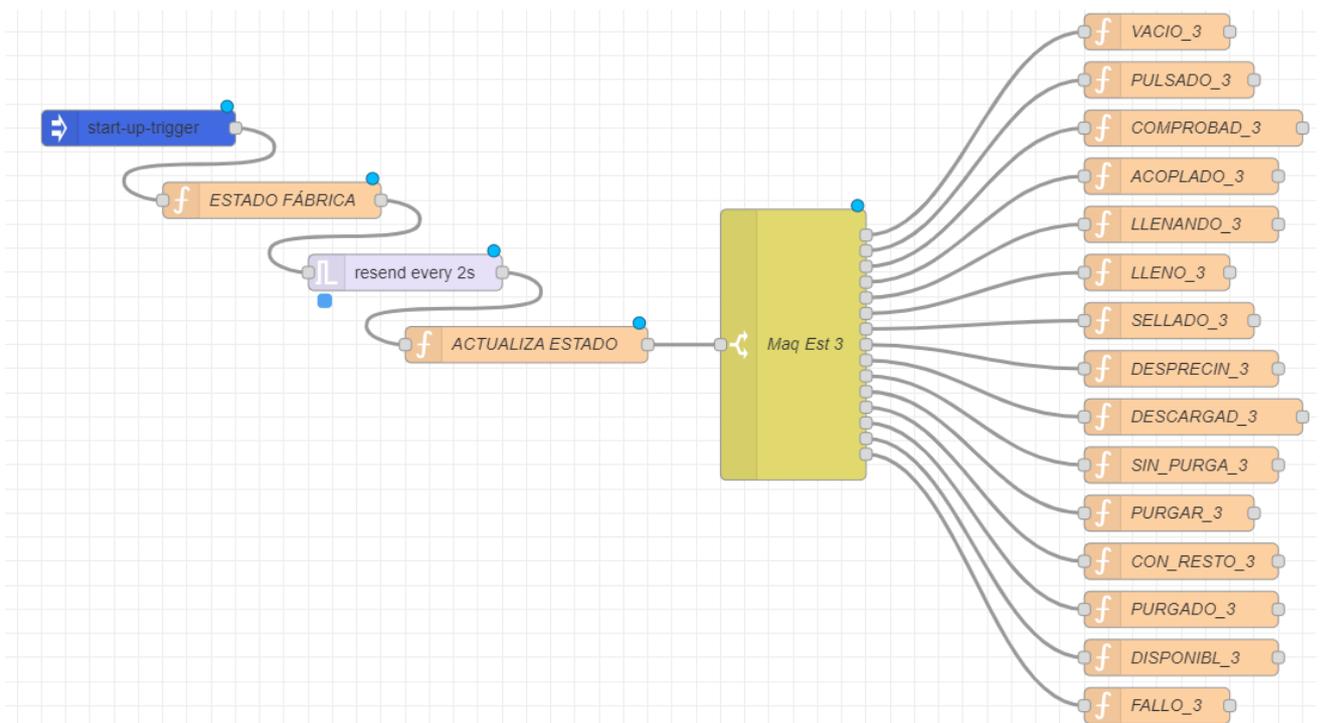


Ilustración 7-28. Flujo Máquina Estados Comp 3.

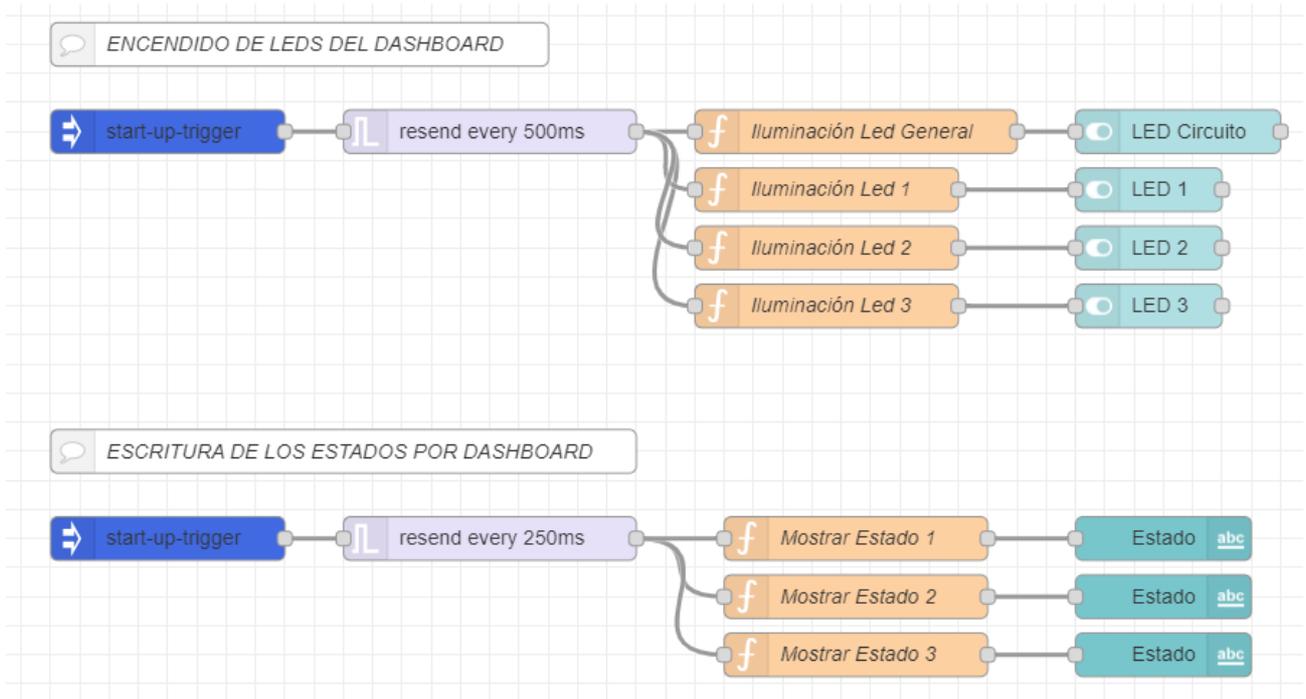


Ilustración 7-29. Flujo Salida Leds y Display.

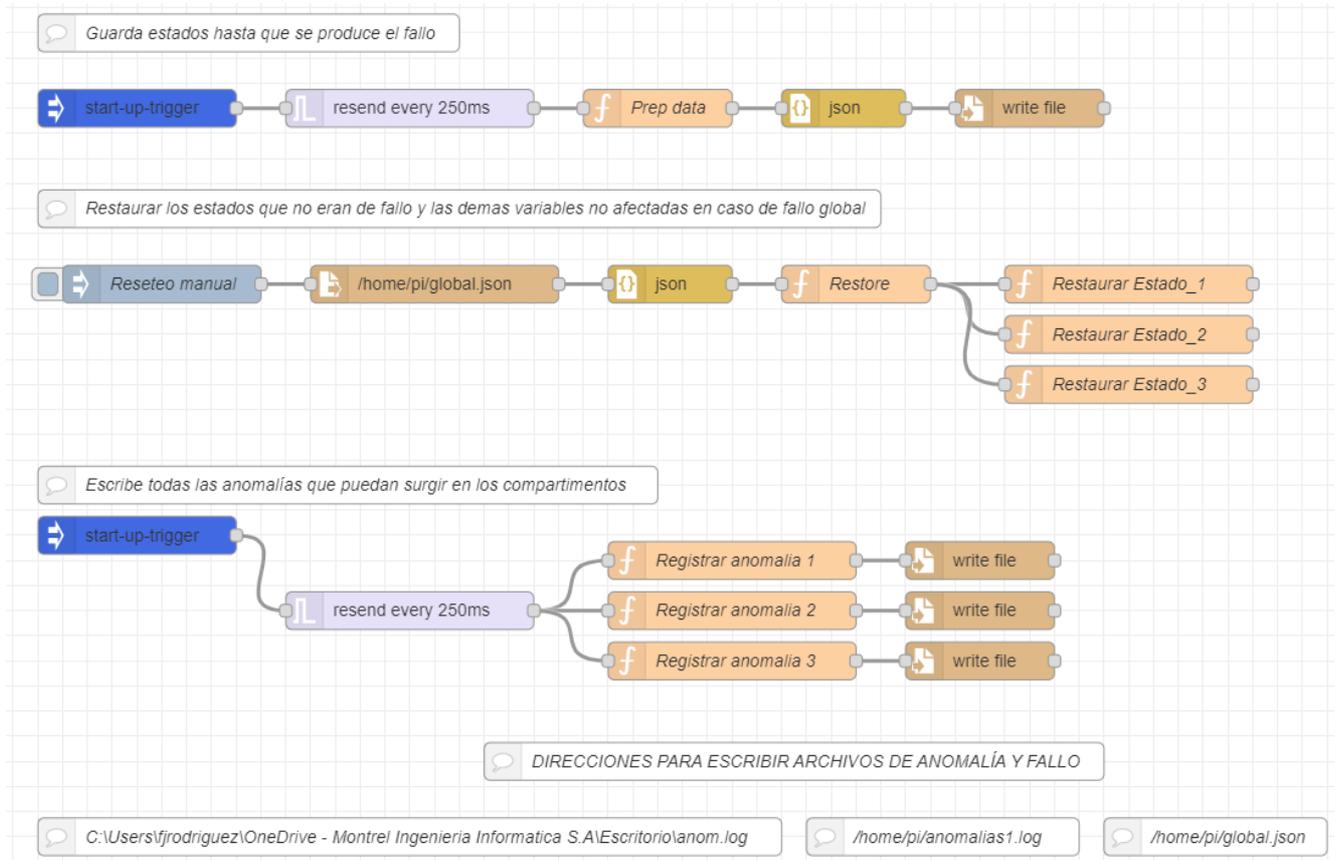


Ilustración 7-30. Flujo Anomalías y Fallo.

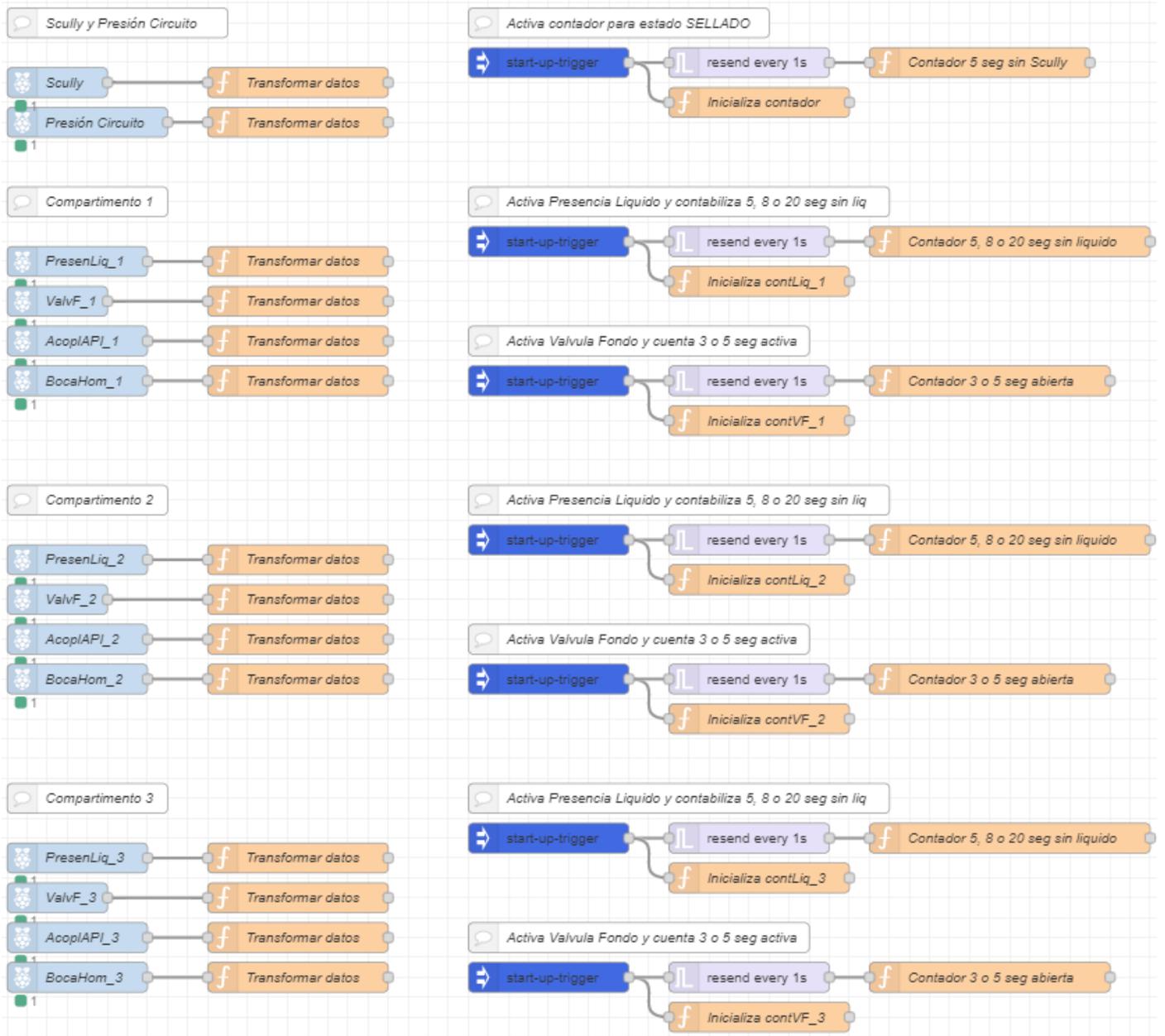


Ilustración 7-16. Flujo Entradas Raspberry.

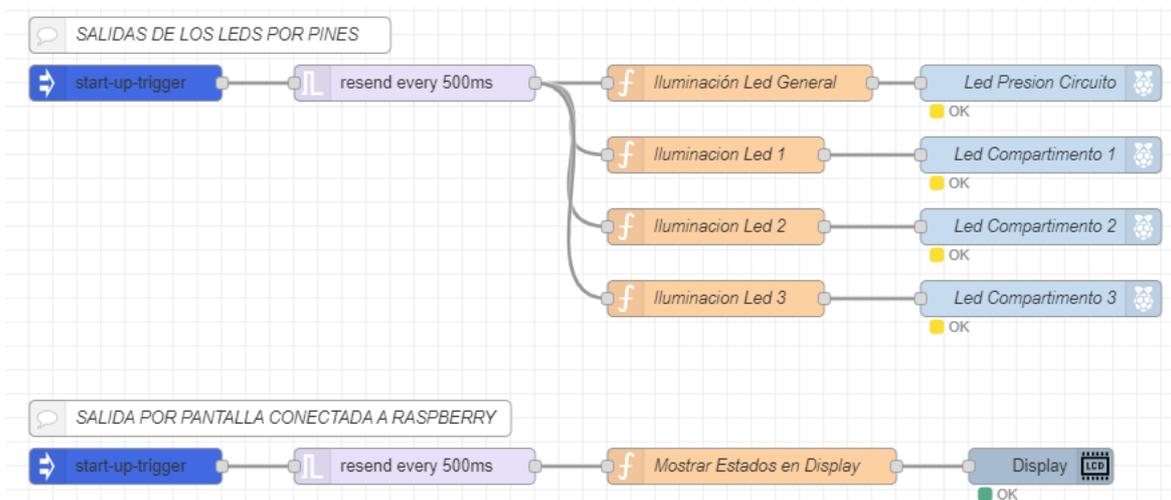


Ilustración 7-17. Flujo Salidas Raspberry.