Informe práctica de laboratorio 2

Presentado

a:

Eugenio Tamura Morimitsu

Realizado por:

Edwin Andrés Calvo Rincón

Santiago Hurtado Bustos

Jennifer López Grajales

Juan José Restrepo Rosero

Facultad de Ingeniería y Ciencias

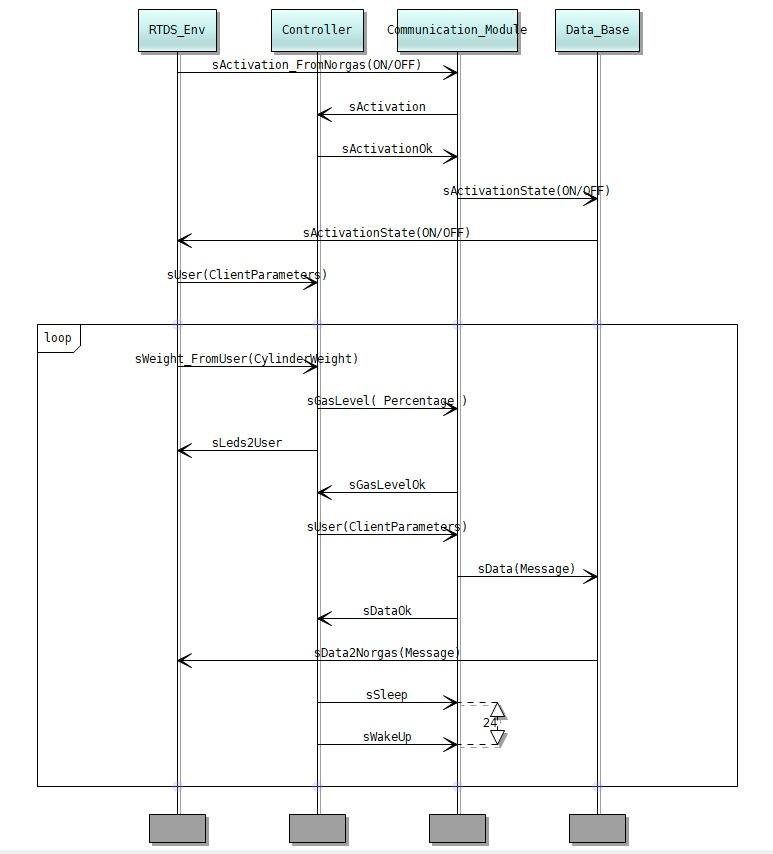
Fecha de entrega: 7/04/2022

Santiago de Cali, Valle del Cauca

**Tabla de contenido:**

1. Diagrama detallado en MSC
   1. RDTS\_Env
   2. Controller
   3. Comunication\_Module
   4. Data\_Base
2. Explicación detallada del SDL
   1. Diagramas de estado SDL para el controller
   2. Diagramas de estado para el Data Base
   3. Comunication\_Module SDL
3. Plan de Pruebas y Simulación
4. Conclucion grupal
5. Conclusiones personales
   1. Jennifer Lopez
   2. Juan Jose Restrepo
   3. Santiago Hurtado
   4. Edwin Calvo

**Diagrama detallado en MSC**



El diagrama MSC TRACE del sistema está conformado por 4 hilos:

**RTDS\_Env**

En este hilo se hace referencia a los stakeholders que interactúan con el sistema como receptores o como los generadores de las señales enviadas a este. Sea de recalcar que RTDS\_Env se compone de la compañía NORGAS y el usuario de GLP que esté en posesión del sistema. Se tienen las señales de entrada sActivation\_FromNorgas, sUser y sWeight\_FromUser. Para las señales de salida se tiene sLeds2User y sData2Norgas. Se especificarán sus funciones y detalles más adelante.

**Controller**

En este hilo se realizan la mayoría de los procedimientos internos del sistema. Se coordina el inicio del sistema con sActivation la cual es un string que pasa primero por el Comunication\_Module que le llega en forma de sActivation\_FromNorgas para ser enviado al controlador, este mensaje proviene de la empresa para activar el sistema de forma remota. Después de verificar el mensaje envía confirmación de inicio de programa a través de sActivationOk que va al Comunication\_Module quien lo envía por sActivationState a Data\_Base para que este se lo envié a Norgas como confirmación de activación del sistema. Al recibir la confirmación Norgas que sería RTDS\_Env le enviará sUser que contiene la identificación de ese sistema y la del usuario que tiene el sistema, para posteriormente usarlos como identificadores de a quien la información de peso del cilindro que se enviará pertenece.

Seguidamente se entra en un bucle donde se realiza primero la toma de peso del cilindro la cual llega al controlador por sWeight\_FromUser. Se transforma los datos en un porcentaje de llenado del cilindro y se transfiere a Comunication\_Module el cual le responderá con sGasLevelOk al controlador como signo de haber recibido la señal. Mientras se hace la confirmación el controlador determina si es necesario prender el LED de alerta de bajo GLP que definimos que sería por debajo de 20% del cilindro de gas lleno. Si el sistema considera por debajo del porcentaje lo activará mandando la señal sLeds2User a RTDS\_Env. Después de la respuesta positiva de sGasLevelOk el controlador responde mandando un mensaje con el identificador y el porcentaje de gas GLP restante en la señal sUser que va al módulo de comunicación que lo envía a Data\_Base para que este se lo envíe a la empresa en RTDS\_Env. El modelo de comunicación enviará sDataOk como confirmación de envío de los datos. Por último se usa la sSleep y sWakeUp para reiniciar el bucle.

**Comunication\_Module**

Este hilo es el encargado de comunicar la compañía con los usuarios que consumen GLP, a través del envió de mensajes con información necesaria para optimizar las operaciones logísticas, dado que uno de los principales problemas que enfrenta NORGAS son las pérdidas económicas, por lo que la interacción entre los stakeholders debe ser efectiva. Este módulo genera y recibe señales del controlador y del data base respectivamente.

**Data\_Base**

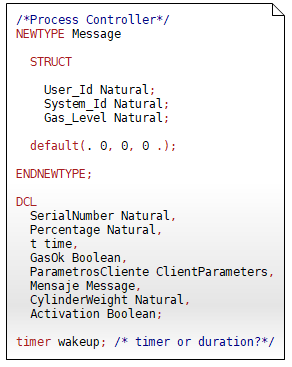
La Data Base se encarga de ser el intermediario entre el sistema y la empresa con respecto al envío de datos pero es unilateral solo recibe datos del sistema y no de la empresa. Se usa de etapa intermedia de comunicación. Dentro de la verificación de activación se puede ver esto donde la llega sActivationState al hilo y después es enviada la misma señal a la empresa. Después dentro del bucle database le llega sData la cual contiene tanto identificación como porcentaje de gas restante en una estructura de datos creada llamada Message. Seguidamente enviará el Mensaje a Norgas pero a través de sData2Norgas.

**Explicación detallada del SDL**

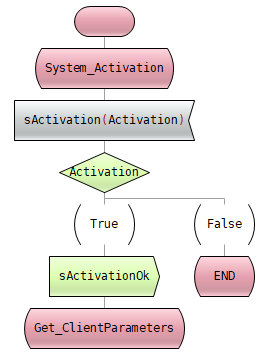
**Diagramas de estado SDL para el controller**

Como se mencionó anteriormente el diagrama MSC está compuesto por 4 módulos fundamentales, los cuales se puede agrupar en dos secciones, sistema de sensado de peso el cual realizará la conversión a GLP para enviarlo a la compañía, y otro que engloba a la empresa NORGAS y a usuarios que se encuentran en su hogar.

Con base a lo anterior se procederá a construir la máquina de estados de cada módulo, con el propósito de explicar cómo interactúan las señales al interior del sistema a través del proceso de intercambio de información. Inicialmente se definen las diferentes variables que transportaran las señales, por ejemplo “SerialNumber”, “GasOk”, “Mensaje”, entre otras. Donde a cada variable le corresponde un tipo de dato, “Natural”, “Boolean” y “Message” respectivamente. Lo anterior se puede ver en la siguiente imagen:

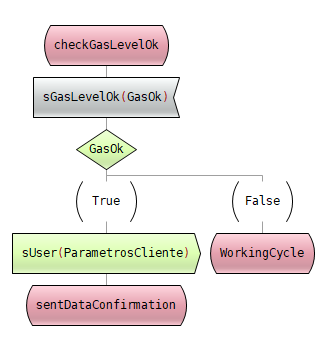


Es importante aclarar que este módulo es el “cerebro” de la operación, por lo tanto recibirá la mayoría de las señales generadas internamente sobre el circuito.



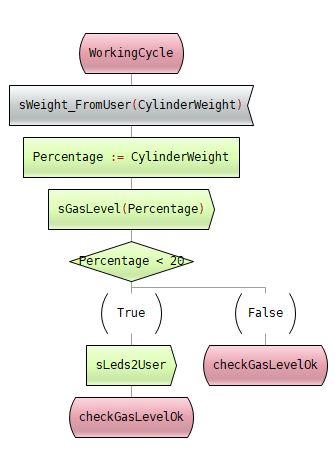
El sistema inicia con el estado ***System\_Activation***  el cual recibe la señal ***sActivation***, dicha señal es de gran importancia dado que es una extensión de la señal que le envía la compañía al módulo de comunicación, para que el controlador active o desactive el sistema. Posterior a ello se evalúa la condición del valor de “Activation”, el valor de está variable conlleva únicamente a dos posibles escenarios:

* El primero será cuando el argumento de la señal es equivalente a un ***True***. En caso de que esto suceda se generará una señal de salida ***sActivationOk***. Esta señal nos sirve para confirmar que el dispositivo ha sido activado por la empresa satisfactoriamente. Posterior a ello pasaremos al estado ***Get\_clientParameters***.
* La segunda opción será cuando el valor de la señal tome el valor lógico de ***False***, lo que conlleva a que la operación del sistema termine completamente.



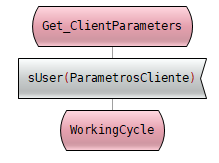
El siguiente diagrama de estado es ***checkGasLevelOk***, el cual tiene como señal de entrada a ***sGasLevel***, con un argumento que puede tomar dos únicos valores: ***True*** or ***False***. Dependiendo el valor que tome el argumento ***Gasok***, ocurrirán las siguientes situaciones:

* Si el valor obtenido es ***True***, entonces tendremos una señal de salida llamada ***sUser***, la cual tendrá como parámetro de entrada, identificación del usuario e identificación del sistema., las cuales se condensan en una variable llamada ***ParámetrosCliente***. Después de almacenar los datos en la variable, se pasará a una etapa de confirmación del envío de los datos.
* En caso de que la señal tome un valor ***False*** indicará que el sistema debe volver a recolectar los datos, identificación del usuario e identificación del sistema a través del estado ***WorkingCycle***.

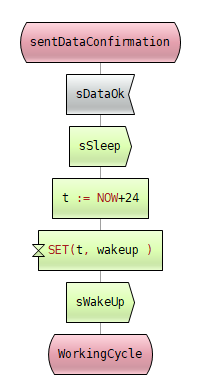


El proceso de trabajo es de gran importancia dado que es el ciclo durante el cual se realizará la toma de los datos de peso, su respectivo envío a la compañía y la visualización por medio de un bombillo a los demás usuarios que se encuentran en sus hogares.

La señal de entrada es ***sWeight\_FromUser***, que tiene como argumento de entrada el peso del cilindro. Posterior a ello se realiza un proceso de conversión, (Peso a porcentaje). Una vez obtenido ese porcentaje, será enviado al interior del sistema, a través de la señal ***sGasLevel***Dependiendo del valor que tome ***percentage***, se encenderá la alerta visual para indicar la cantidad de GLP. Al finalizar la ejecución de alguna de las alternativas, nos encontraremos en el ***checkGasLevelOk***.



El proceso anterior es relativamente sencillo, dado que solo recibe la señal ***sUser***, con los parámetros del cliente



Este módulo permite llevar el conteo de 24 horas, con el propósito de establecer correctamente los tiempos de hibernación del sistema. Para esto, requiere que la señal de entrada sea igual a ***sData***. Una vez terminada la toma de los datos, se debe esperar el tiempo mencionado anteriormente para llevar al próximo estado ***WorkingCycle***.

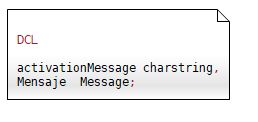
El inicio de los estados será dado por el estado ***System\_Activation***.

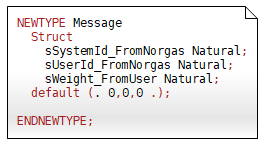
**Diagramas de estado para el Data Base**

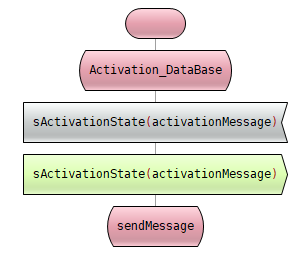
Este módulo del sistema estará compuesto por dos variables, ***ActivationMessage***y ***Mensaje*** las cuales son booleana y estructura respectivamente.

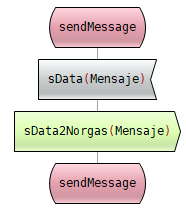
La variable tipo ***ActivationMessage***se utiliza para confirmarle a la empresa de Norgas que si le activo el dispositivo, mientras que la variable tipo estructura fue creada, la cual se compone de los siguientes parámetros:

* **Mensaje:** está señal fue creada con el propósito de agrupar tres señales de identificación, las cuales son consideradas indispensables para el correcto funcionamiento del sistema,siendo estas la identificación del sistema sSystemId\_FromNorgas, la identificacion del usuario sUserId\_FromNorgas y la cantidad de GLP restante en peso sWeight\_FromNorgas.
* **sSystemId\_FromNorgas:** dicha señal representa el código de identificación de cada sistema asignado a su correspondiente usuario.
* **sUserId\_FromNorgas:** representa el código de identificación del usuario al cual se le dió el sistema.
* **sWeight\_FromNorgas:** hace referencia a la cantidad de GLP restante en el cilindro medido por el sistema.

****

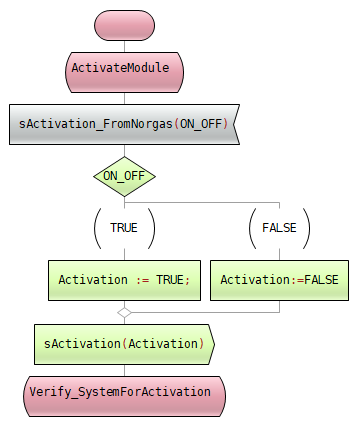
****

****

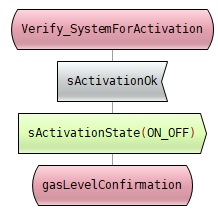
****

**Comunication\_Module SDL:**

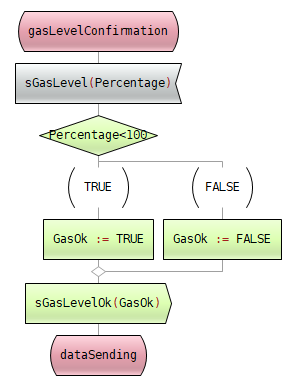
Este módulo es el encargado de establecer la comunicación tanto con el controlador del sistema como con la base de datos a la que se enviarán los datos tomados del nivel de gas del cilindro. Para llevar esto a cabo, se empieza con el estado ***ActivateModule*** en donde se efectúa la activación del módulo mediante una señal que proviene de la empresa de NORGAS llamada ***sActivation\_FromNorgas*** la cual contiene un parámetro de tipo booleano ***ON\_OFF****.* Dicha señal se comunicará con el controlador del sistema ***Controller*** a través del respectivo canal de comunicación que hay entre ellos dos **cController\_Communication***,* en donde el mensaje enviado por la compañía se transmitirá al controlador por medio de la la señal ***sActivation*** cuyos valores serán ***True*** o ***False,*** y con estos se determinará si el sistema podrá seguir operando o no, por medio de la etapa en la que se verifica dicho estado. Está se conoce como ***Verify\_SystemForActivation.***

******

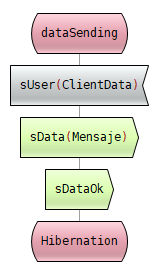
Para el estado de verificación de la correcta activación del sistema, se recibirá una señal que comprobará que el sistema fue ajustado de manera adecuada ***sActivationOk***, además esta etapa es en la que se mantendrá una constante comunicación entre el módulo de comunicación y el controlador, pues hay un intercambio continuo de señales entre los dos hilos. Por otro lado, está la señal ***sActivationState*** la cual tiene como parámetro el booleano ***ON\_OFF*** y se envía al ***RTDS\_Env****,* por medio del canal de comunicación que hay entre la base de datos y el módulo de comunicación. Una vez se haya terminado la etapa de activación, se pasa a la toma y confirmación de la correcta medición del nivel de gas de la pipeta.



Para este estado de confirmación del nivel de gas, se requiere tener una señal que contenga dicha lectura. Es por eso que se empleará de una señal llamada ***sGasLevel*** la cual tiene un parámetro de la medición tomada y se mostrará en un porcentaje de llenado. Este porcentaje será evaluado por medio de una condición con la cual se busca mantener que la medición del nivel de gas sea adecuada; para ello, se buscará que la lectura arrojada sea coherente con un valor de la realidad, es decir, que nunca sea superior al 100% de capacidad. Una vez evaluada esta condición, la salida se almacenará en una variable llamada ***GasOk*** y será pasada como parámetro a una señal que evaluará dicha coherencia medida, está señal es llamada ***sGasLevelOk***, de ahí en adelante,



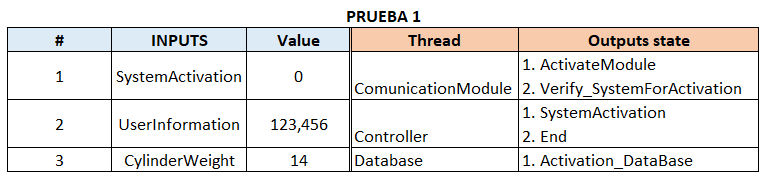
Una vez se haya tomado la medición del nivel de gas actual en el cilindro, sigue la etapa del envío de dicha medida al usuario, junto con su identificación personal y la del cilindro con el que cuenta. Para ello, se hace uso de una señal que toma los datos del cliente que son ***User\_Id, System\_Id y GasLevel*** y estos se guardan en una estructura que se llama ***Mensaje,*** para posteriormente ser enviada como parámetro a la señal ***sData.*** No obstante, se tiene que chequear que los datos leídos son correctos previamente al envío, para ello se usa la señal de confirmación ***sDataOk*** y una vez se haya confirmado que todo es correcto, se mandan los datos a la empresa y se pasa a un estado de hibernación.



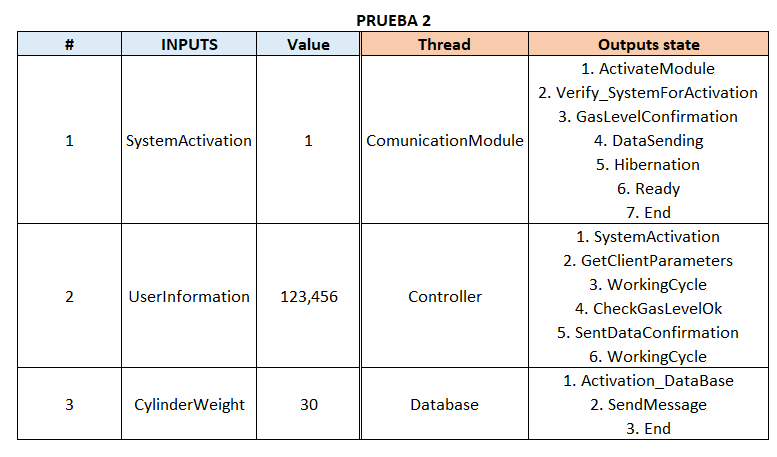
En este último estado, es donde el sistema pasa a “dormir”, es decir, disminuirá su consumo de potencia y permanecerá así hasta que hayan pasado 24 horas para realizar el siguiente proceso de medición y envío de los datos nuevos a la compañía. Cuando esto suceda, el sistema pasa a un estado de ***Ready,*** en el cual se enviará una señal que represente que es momento de “despertarse”, conocida como***sWakeUp*** y así comenzar de nuevo con su operación diaria.

**Plan de pruebas y simulación**

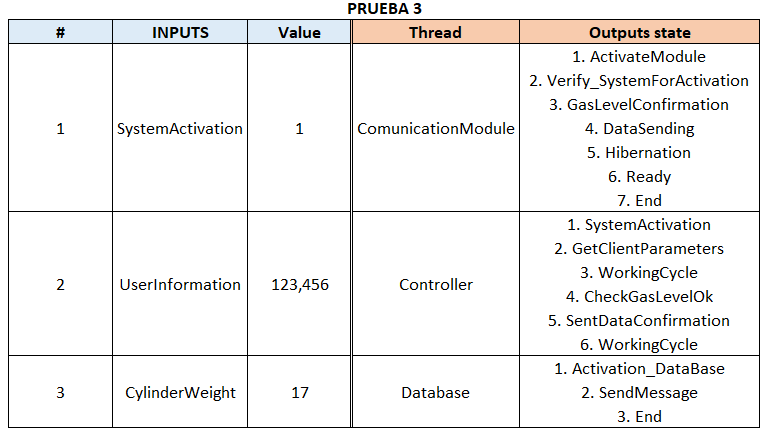
La simulación del sistema es posible corroborarla en el archivo adjunto al informe, donde se encuentran además todos los diagramas y pruebas realizadas. En esta parte planteamos los resultados obtenidos en el plan de pruebas que se le realizó al sistema. Para llevarlo a cabo, se realizaron 3 pruebas de la siguiente manera: Se variaba el valor de las entradas y se observaba el orden de los estados de salida en cada hilo. Cabe aclarar que la entrada de ***UserInformation*** permaneció constante a lo largo de las pruebas debido a que esta no tenía una gran influencia en la ejecución del programa, por tanto, los estados de salida serían los mismas en caso de variar esta entrada específicamente.

****

En la primer prueba se tomó el caso en el que el sistema no fuese activado, es decir, que la señal de SystemActivation fuera cero. Para este caso, los estados de salida en cada hilo son muy pocos, pues no se realizaría casi ningún proceso, solamente se iniciarían y acabarían, como en el ejemplo del ***Controller.***



En esta segunda prueba, el sistema sí fue activado y se envió un peso superior a 30 kg, lo que significaría que no se necesita encender el led, pues el usuario aún contaría con gas suficiente. De aquí se puede observar que se realizan muchos más estados en comparación al caso anterior, y se debería también tener en cuenta que internamente la señal de ***sLeds2User*** no fue enviada, por tanto, el led no se encendió.



Finalmente, en la prueba 3 se establece el caso de un sistema activado y un peso del cilindro inferior a 20 kg, por consiguiente, la señal que controla el encendido del LED sería activada y el usuario tendría el indicador visual que advierte sobre un nivel de GLP bajo. Por otra parte, se puede observar que los estados de salida son los mismo que en la prueba anterior, pero internamente algunas señales no serían las mismas, tal como para el caso de ***sLeds2User*** que antes era 1 y en este caso se quedaría en cero. Por último, para el hilo del ***controller*** el sistema se queda en ***WorkingCycle*** haciendo referencia a la iteración del ciclo, pues en este estado se está esperando por el peso del cilindro para poder continuar.

**Conclusiones personales**

**Jennifer Lopez**

En el proceso de realización de este avance puede constatar un mejor manejo de las herramientas como PragmDev. Además se pudo constatar desde la parte del PIP una mejora en el sistema planteado ya que se analizó más a detalle cómo serían las interacciones en el sistema y más importante las interacciones con los actores. Antes se analizaba puramente desde nuestro punto de vista o fuertemente influenciado por el de la empresa sin ver que era lo necesario, como la frecuencia de toma de mediciones la cual pasó de 2 al dia a 1 al dia y la posibilidad de mejorarlo para que en caso de inactividad no usar tantos recursos. Otro aspecto fue la visualización del usuario al cual pasamos de 6 LEDs a 1 LED y a considerar la activación del sistema de alerta a un nivel de gas más bajo.

**Juan José Restrepo**

En esta etapa del curso, se nos ha permitido tener un mejor entendimiento sobre la problemática que estamos abordando en el Proyecto de Integración Profesional y gracias al software de modelamiento de PragmaDev ha sido posible mejorar la solución propuesta a este reto, pues se hizo un análisis con mayor profundidad en donde se buscaba ver las posibles interacciones que tiene el sistema con todas sus etapas de operación y las señales involucradas en cada proceso que se lleva a cabo. Una de estas mejoras a destacar fue la adición del hilo de la base de datos, en el cual se comprendió mejor su relación e interacción con el sistema de pesado y el papel de la empresa. Además, se mejoró la comprensión de la parte de la frecuencia con la que se enviarán los datos y cómo se hará para que el sistema sepa el momento en el que tendrá que medir el nivel de gas y los factores que deberá tener en cuenta antes de enviarlo a la empresa.

**Santiago Hurtado**

Para esta entrega se evidenció un mejoramiento en cuanto al modelamiento del sistema de medición de peso, en el software de PragmaDev debido a que se tuvo una mayor claridad de los factores que influenciaban sobre este mismo y cómo interactuaban unos con los otros. Gracias a esto, se tuvo un mejor orden al momento de modelar el sistema de tal forma que se llegó a un modelo claro que cumplía con las funcionalidades esperadas incluso en casos específicos que a diferencia de la entrega anterior, no se habían tenido muy en cuenta, por lo que nuestro modelamiento resultó ser más acorde a la realidad.

**Edwin Calvo**

Para esta entrega, logré apropiarme de mejor manera de los conocimientos tanto en MSC y en SDL, de modo que comprendí mucho mejor el funcionamiento de las simulaciones, los tipos de datos, la utilidad de los canales y demás. Por otra parte, también me parece que aportó a los aspectos de rediseño en el Proyecto de Integración Profesional sobre el cual estoy trabajando junto a mi equipo, volviéndonos más críticos sobre las características del sistema y las señales que interactúan con el mismo.

**Conclusión grupal**

Con la realización de esta parte del PIP fue posible evidenciar un gran avance el cual benefició bastante el desarrollo de este proyecto dado que por medio de la herramienta PragmaDev fue posible mejorar el diseño de nuestro sistema planteado al problema de medición del nivel de gas por medio del peso del cilindro, de tal manera que como grupo pudimos identificar nuevos factores y casos que no se habían tenido en mente anteriormente, permitiendo así tener la realización de un diagrama más simplificado pero concreto del producto que se planea implementar.