



Pontificia Universidad
JAVERIANA
Cali

Sistema de Medición de Datos para animales en ambientes Co- lombianos (RealTinfo)

Entrega #1

Equipo:

Luis Alberto Salazar
Dilan Andrés Correa
Guido Ernesto Salazar

Internet de las Cosas y Computación en la Nube
Carrera de Ingeniería de Sistemas y Computación
17 de mayo de 2023

Índice general

1. Resumen del proyecto	3
1.1. Resumen ejecutivo	3
1.2. Glosario	4
1.3. El equipo de diseño	4
2. Contexto y concepción de la solución propuesta	7
2.1. Definición del problema	7
2.2. Restricciones del contexto	8
2.3. Identificación de necesidades de los usuarios objetivo	9
2.4. Requerimientos y especificaciones de la solución propuesta	9
3. Diseño del sistema base	11
3.1. Diagrama de contexto	11
3.2. Diagramas MSC en el nivel de sistema	12
3.3. Arquitectura SDL y procesos	15
3.4. Verificación del sistema base	24
3.5. Prototipado en C	26
4. Escalamiento del sistema	29
4.1. Diagramas MSC	29
4.2. Arquitectura SDL y procesos	33
5. Reflexión del proceso	40
6. Costos y presupuesto asociados al proyecto	42
7. Referencias	44
7.1. Personas	44
7.2. Proveedores	44

Capítulo 1

Resumen del proyecto

Para las personas que se dedican a la investigación de animales les resulta complicado hacer salidas de campo para la toma de información, ya que esto representa un problema para la precisión de la información y de su presencia asociada al entorno natural de los animales. Es por eso que se presenta un proyecto que diseña un sistema para la toma de información de animales por medio de dispositivos electrónicos, cuya información es transmitida y almacenada en la nube,.

Este proyecto busca mostrar el funcionamiento de dicho sistema, considerando como usuarios finales investigadores y personas interesadas en la información recolectada en una zona específica en el contexto Colombiano, dicha información es obtenida por medio del uso de internet para la transmisión de datos y su recolección.

1.1. Resumen ejecutivo

Como se conoce, los investigadores se encargan de estudiar muchas características de los animales, es por eso que es necesario definir que características específicas se puede extraer de los animales, para ello se pretenden extraer datos como:

- Imagen de los animales.
- Fecha de la captura de los datos.
- Grabaciones de los animales.
- Audio de los sonidos emitidos por los animales y el entorno.

Para la recolección de esta información se pretende utilizar los siguientes dispositivos electrónicos: *Cámaras de Vídeo*, *Micrófonos* y *Sensores de Movimiento*. Algo importante que se debe aclarar de todo esto es que los dispositivos de la toma de datos estarán conectados a un controlador, dicho controlador emitirá y recibirá información por medio de internet hacia y de un servidor que funciona en lo que se conoce como nube, en donde se almacenará la información en una base de datos de forma periódica. Por último, los usuarios finales (como investigadores) pueden acceder a esta información por medio

de una aplicación en un dispositivo de cómputo, ingresando parámetros o rangos para filtrar la información si lo desean.

1.2. Glosario

Algunos términos que se deben definir dentro del contexto del proyecto son:

- **Nube:** es una red enorme de servidores remotos, conectada a través de internet, que funciona como un único ecosistema. Están diseñados para almacenar y administrar datos, ejecutar aplicaciones o entregar contenido.
- **Controlador:** es un circuito o componente electrónico utilizado para controlar otro circuito o componente electrónico.

1.3. El equipo de diseño

El equipo de diseño y trabajo se compone de los siguientes integrantes:



Figura 1.1: Luis Alberto Salazar. Estudiante de Ingeniería en Sistemas.

Luis, se ve a él mismo como una persona responsable con sus proyectos y sueños, considera que su valor más importante que lo caracteriza es la empatía. Uno de sus más grandes rasgos es la capacidad de liderar a las personas en cualquier proyecto o situación, pues siempre busca ayudar a las personas a alcanzar un objetivo en común. Se considera disciplinado y perseverante para llevar a cabo sus metas y proyectos.



Figura 1.2: Dilan Andrés Correa. Estudiante de Ingeniería en Sistemas.

Dilan, es un estudiante que nunca se rinde para realizar sus tareas, se ve a él mismo como una persona que se esfuerza por lo que hace y la empatía hacia los demás. El siempre busca el bienestar de las personas, pues en sus aportes trata de analizar el menor impacto para los demás. Se considera como una persona que ve más allá de los problemas y siempre busca solucionar los dilemas que se le presentan, le gusta escuchar y entender lo que dicen los demás. Se considera buen compañero para trabajar en equipo.



Figura 1.3: Guido Ernesto Salazar. Estudiante de Ingeniería en Sistemas.

Guido, se considera a si mismo como una persona tranquila y crítica de muchas cosas que él ve. Siente que ve más allá de las cosas y se considera como una persona analítica de todas las situaciones que tiene en su vida. Es una persona responsable y comprometida con lo que hace, busca siempre hacer la mejor solución y considerar todos los casos posibles, le encanta resolver problemas y retarse buscando soluciones que sea eficientes y que solucionen en su totalidad el problema, la mejor solución.

Capítulo 2

Contexto y concepción de la solución propuesta

2.1. Definición del problema

Actualmente los biólogos presentan un gran problema para la recolección de datos sobre animales, ya que en muchos casos su presencia puede afectar el resultado obtenido o puede ser impreciso por las condiciones que pueden presentar la toma de los datos.

Con este proyecto se busca facilitar a los biólogos la recolección de los datos, ya que por medio de algunos dispositivos se espera recolectar los datos esperados sobre cualquier problemática. La idea es que exista un equipo de trabajo que pueda llevar control sobre el funcionamiento de los diferentes dispositivos para que el sistema se encuentre correctamente en todo momento, ya que para la recolección de los datos esto es algo fundamental.

Más allá de la facilidad que se tendría para la recolección de los datos, como se mencionaba en un inicio, la presencia del ser humano en la toma de los datos puede muchas veces alterar el resultado, ya que podría ser un factor adicional en un ambiente con animales que no debería estar. De igual forma, podría asegurar que en para la recolección de algunos datos sobre los cuales algunas veces la vida del animal corre peligro, es muy importante la inmediatez del reconocimiento de los datos. Por ejemplo, existe un problema actualmente con los murciélagos y es que son animales muy sensibles de tratar, para la investigación sobre estos, el método que se utiliza para poder capturarlos es el siguiente: se utiliza una red de Nylon que se conoce como Red de Niebla”, que es lo suficientemente delgada para que cuando caigan sobre ella no los hiera, sin embargo, puede suceder que si pasan mucho tiempo atrapados en la red pueden perder la vida. Es por esto que la intervención rápida de los biólogos o investigadores para recogerlo es muy importante. El método actual que se utiliza según las preguntas realizadas a los biólogos de la Universidad Javeriana de Cali, es el mismo mencionado anteriormente, sólo que, como la presencia de seres humanos altera el resultado, lo que se hace es montar la red de Niebla y se hace un registro cada 30 minutos por algún bió-

logo o investigador si la red ha capturado algo. Los biólogos nos reportaron que en muchas ocasiones no se puede obtener con exactitud el tiempo que llevan atrapados en la red, y algunos murciélagos por desgracia pierden la vida. Es por eso, que poder tener un sistema que detecte cuando un murciélago es atrapado podría ayudar con la exactitud de los datos tomados y salvar la vida de los murciélagos que son capturados para investigaciones.

- ¿Qué y quiénes están alrededor de la problemática del proyecto?
- ¿Quiénes van a querer usar la solución de este problema?
- ¿Por qué la van a querer usar?
- ¿Qué necesidades generales existen en el área de trabajo asignada?
- En general, ¿Cuál es el contexto del problema?

2.2. Restricciones del contexto

Actualmente, el proyecto está propuesto para diferentes localizaciones a nivel mundial ya que se puede buscar recolectar la misma información para diferentes animales. Sin embargo, pueden existir algunos problemas para el desarrollo de la recolección de los datos que pueden ser los siguientes:

- Algunos países presentan grupos armados al margen de la ley que tienen custodiadas algunas zonas que contienen a su vez hábitat naturales de animales.
- Si bien es cierto que actualmente existen software que puedan reconocer animales que está enfocando una vídeo cámara, es importante tener en cuenta que la precisión de estos sistemas puede variar dependiendo de varios factores, como la calidad de la imagen o vídeo, la variabilidad de las características de los animales, y la complejidad del entorno en el que se encuentran.
- El proyecto requiere de unas contribuciones económicas de parte de las entidades que quisieran hacer parte, es por esto, que no llegar al presupuesto mínimo podría concluir a la no ejecución del proyecto. No obstante, se debe tener en cuenta, que más allá del precio de los dispositivos a utilizar, la consideración importante sobre ellos es que sean resistentes a los diferentes climas que pueden afrontar.
- Aunque este proyecto está enfocado en la recolección de datos sobre animales para el conocimiento sobre estos, es posible que algunas políticas sobre el cuidado de la biodiversidad en algunos países pueda que no permitan este sistema.

2.3. Identificación de necesidades de los usuarios objetivo

- Investigadores y biólogos que estudian la ecología y el comportamiento de los animales.
- Organizaciones de conservación que monitorean la salud y la población de especies en peligro de extinción.
- Agencias gubernamentales que regulan la caza, la pesca y otras actividades relacionadas con la vida silvestre.
- Empresas que trabajan en la industria de la energía, la minería y otras industrias que pueden afectar los hábitats de los animales.

Como fue mencionado en la definición del problema, los biólogos de la universidad manifestaron que para recolectar datos sobre animales en muchas ocasiones la presencia humana alteraba el posible resultado, es por esto que implementar dispositivos que puedan recolectar datos podría ayudar mucho. De igual forma, poder realizar monitoreos a distancia podría beneficiar la analítica de datos.

También, como se mencionó en la definición del problema, para capturar algunos animales sobre los que se quiere hacer investigaciones, es importante que la captura del animal se haga de la mejor manera y esta no arriesgue la vida del animal.

2.4. Requerimientos y especificaciones de la solución propuesta

- Capacidad de almacenamiento: El sistema debe ser capaz de almacenar grandes cantidades de datos de vídeo y audio de alta calidad de forma eficiente.
- Duración de la batería: Los dispositivos de recolección de datos deben tener una duración de batería suficiente para funcionar durante largos periodos de tiempo.
- Resistencia a las condiciones ambientales: Los dispositivos deben ser capaces de soportar condiciones climáticas diversas, como lluvia, viento, temperaturas altas, entre otros.
- Precisión del reconocimiento de animales: El software de reconocimiento de animales debe tener una alta tasa de precisión en la identificación de especies y en el seguimiento del movimiento y comportamiento de animales.
- Fácil instalación y mantenimiento: Los dispositivos de recolección de datos deben ser fáciles de instalar y mantener, con un mínimo de piezas móviles y componentes que requieran mantenimiento.
- Protección de los datos: Los datos recolectados deben ser almacenados y transmitidos de forma segura y protegida para evitar cualquier acceso no autorizado o robo de información.

- Minimización de gastos energéticos y económicos: Si bien es cierto que se requieren dispositivos de buena calidad, se debe priorizar que el proyecto pueda ser llevado a cabo en un sistema que es óptimo en el ahorro de recursos. Es por esto que se necesita de un controlador que envíe una señal al dispositivo para que este se encienda y así poder ahorrar mayor batería en el transcurso de la ejecución del proyecto y así también, ahorrar gastos.

Capítulo 3

Diseño del sistema base

3.1. Diagrama de contexto

El proyecto cuenta con el siguiente diagrama de contexto:

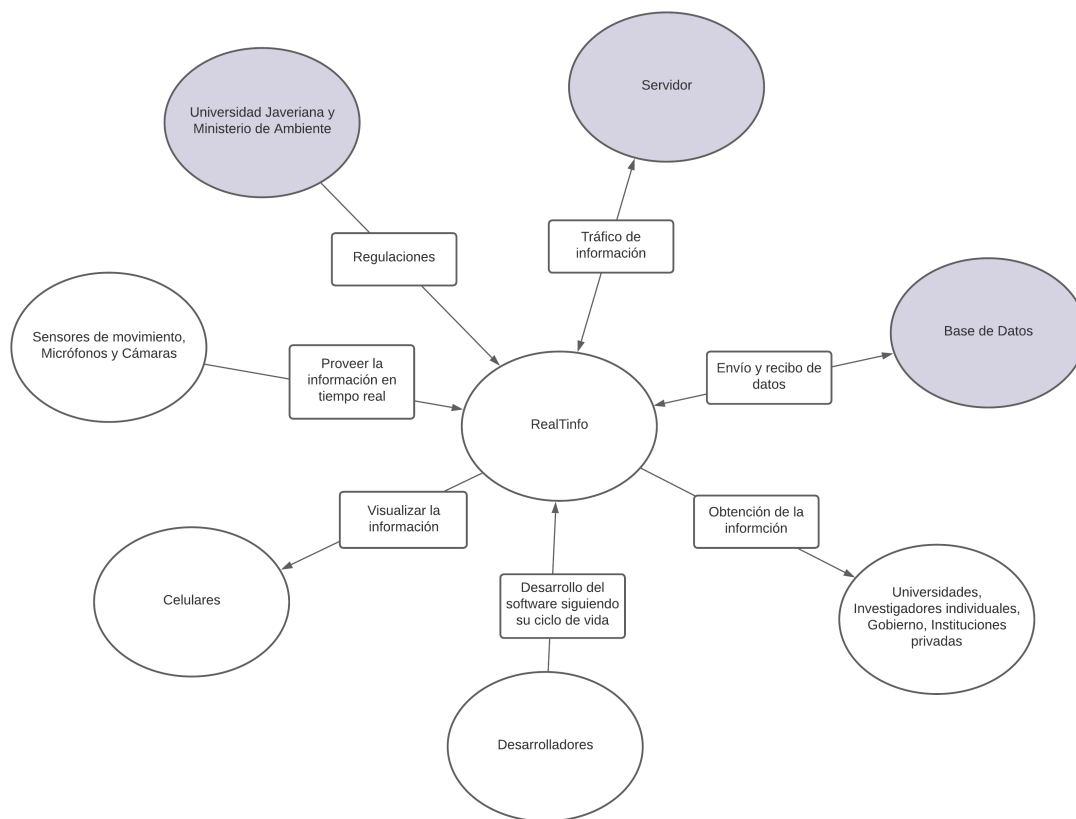


Figura 3.1: Diagrama de Contexto.

El sistema tiene varios actores dentro del diagrama, las razones por las cuales están estos actores son:

- Las regulaciones están dadas por el Ministerio de Ambiente y la Pontificia Universidad Javeriana Cali, ya que son las principales entidades que

limitarán el proyecto con respecto a las restricciones para un correcto desarrollo y ejecución.

- El servidor funciona como un dispositivo que se encuentra en la nube y funciona por medio de internet para pasar la información entre la base de datos, los dispositivos de cómputo y los dispositivos para la toma de la información.
- La base de datos se va a usar para guardar la información que es transmitida.
- Las universidades, investigadores individuales, gobierno, etc. Son los usuarios finales a los cuales potencialmente se les puede transmitir la información.
- Los desarrolladores se encargan de diseñar el sistema y revisar su funcionamiento.
- Los celulares son los dispositivos de cómputo que van a recibir la información de la base de datos que soliciten.
- Por último, están los dispositivos que toma la información y la transmiten.

3.2. Diagramas MSC en el nivel de sistema

El diagrama MSC se compone de varias partes, la primera parte es la siguiente:

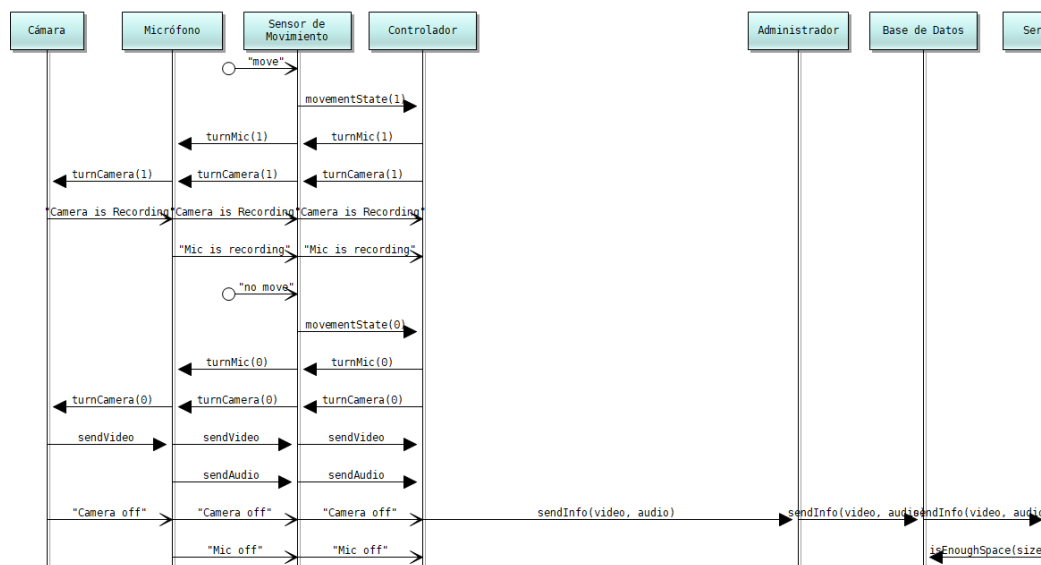


Figura 3.2: Diagrama Message Sequence Chart (MSC).

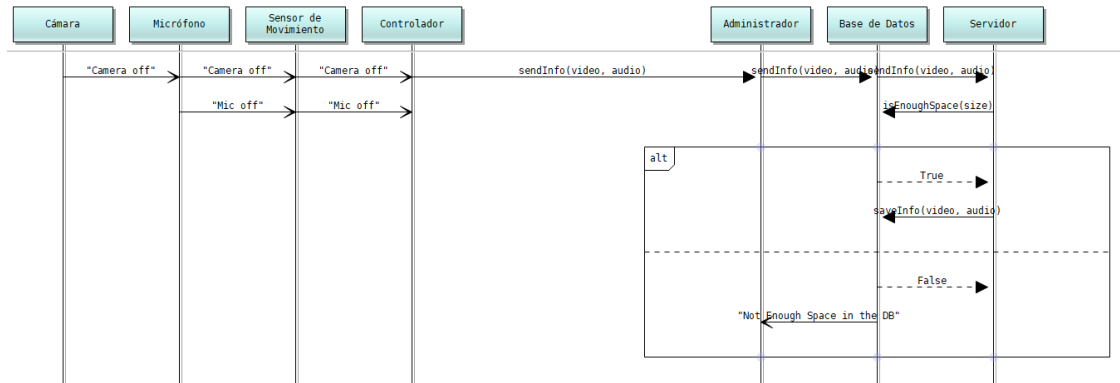


Figura 3.3: Diagrama Message Sequence Chart (MSC).

En esta primera parte en las figuras (3.2) y (3.3), funciona cuando se activa el trigger de movimiento en el sensor de movimiento, cuando esto es activado se envía una señal al controlador y el controlador ordena prender los dispositivos para que empiecen a grabar, una vez que el movimiento se detiene, después de un timer que tiene el sensor de movimiento, se manda una señal para indicar que no hay movimiento al controlador y por ende ordenar apagar la cámara y el micrófono, pero antes estos envían lo grabado para ser almacenado en la base de datos.

En la base de datos se verifica si hay espacio, si hay espacio las grabaciones se guardan y cuando no se pierden. Al mismo tiempo que ocurre lo anterior se apagan los dispositivos para ahorrar su energía.

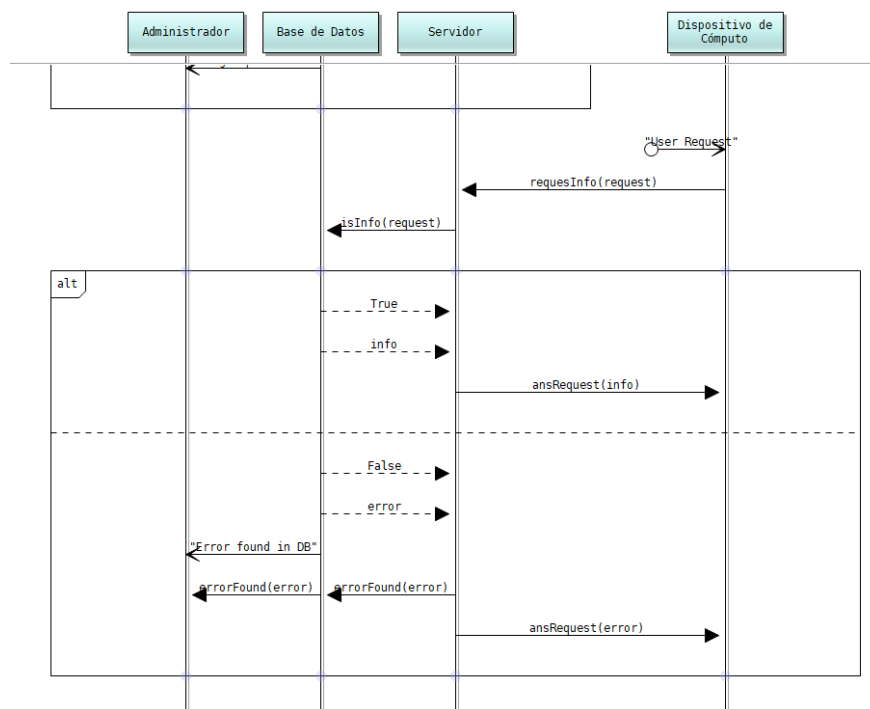


Figura 3.4: Diagrama Message Sequence Chart (MSC).

En esta parte del diagrama se muestra cuando un usuario hace una solicitud para obtener la información del sistema, cuando se pide la solicitud se revisa en la base de datos dos casos: 1) si la información solicitada está, entonces le avisa al servidor que es correcto, entonces se envía la información al usuario para que le sea desplegada, 2) el segundo caso es cuando la información solicitada tiene algún error (no se encuentra, está corrupta, etc.), este error se el envía a un administrador del sistema y se le muestra al usuario que ocurrió un error y la razón.

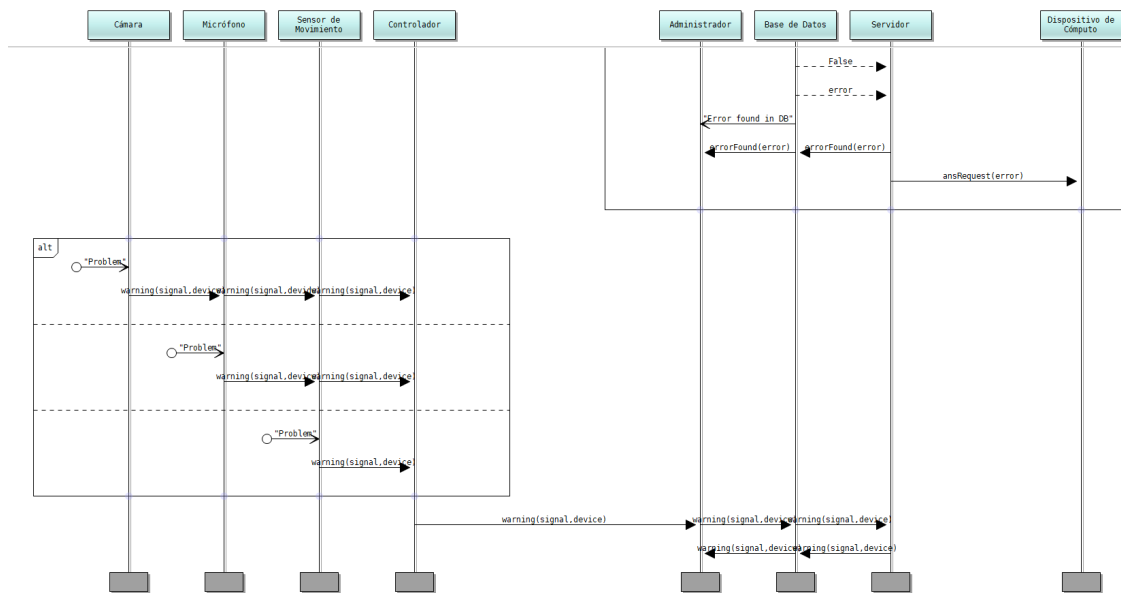


Figura 3.5: Diagrama Message Sequence Chart (MSC).

En esta parte del diagrama se muestra cuando ocurre un problema con los dispositivos de la toma de información, ocurren varios casos:

- La primera es cuando la cámara tiene un problema por la temperatura, por la batería o el almacenamiento interno de la misma, en cualquiera de los casos se envía una señal y el dispositivo hacia el controlador.
- La segunda es cuando el micrófono no tiene batería suficiente o tiene más almacenamiento, en estos caso se le envía una señal y el dispositivo al controlador.
- Por último, si el sensor de movimiento presenta algún tipo de problema este va a enviar una señal y el dispositivo al controlador

Cuando cualquier señal es enviada al controlador este envía el dispositivo que la envió y el problema (la señal) al administrador para que haga el debido proceso para hacerse cargo de este problema.

3.3. Arquitectura SDL y procesos

Después de haber realizado los diagramas MSC, se pasó a realizar el diagrama SDL sobre la arquitectura y comunicación del sistema, en donde se muestran las interacciones entre las señales con los bloques y procesos generados del sistema.

Primeramente, el sistema cuenta con una arquitectura general y una sección de declaraciones con respecto a las señales, tipos y variables globales, es decir, que se pueden acceder en cualquier sección de la arquitectura, (ver figura 3.6)

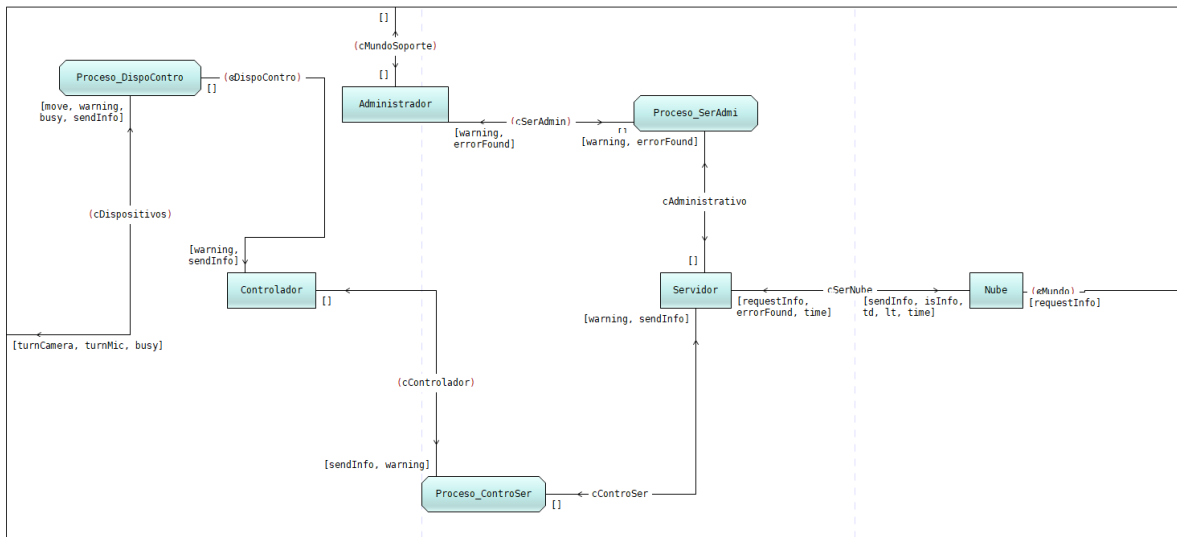


Figura 3.6: Diagrama SDL

Como se puede observar, tiene varios canales para el paso de mensajes y se comunica entre procesos y bloques para el manejo de la información que se solicita.

```
/* Signals of the system */  
SIGNAL move(boolean), warning(device, senal);  
SIGNAL sendInfo(info);  
SIGNAL requestInfo(request), isInfo(request);  
SIGNAL busy, time(t);  
SIGNAL ansRequest(ans);  
SIGNAL errorFound;  
SIGNAL turnCamera(boolean), turnMic(boolean);
```

```
/* Definition of the types */  
NEWTYPE info  
  LITERALS video, audio;  
ENDNEWTYPE;  
  
NEWTYPE device  
  LITERALS camera, microphone, sensor;  
ENDNEWTYPE;  
  
NEWTYPE senal  
  LITERALS baterly, externo, almacenamiento;  
ENDNEWTYPE;  
  
NEWTYPE request  
  LITERALS get, read;  
ENDNEWTYPE;  
  
NEWTYPE ans  
  LITERALS errores, respuesta;  
ENDNEWTYPE;  
  
SYNTYPE t = INTEGER  
  CONSTANTS 1:10  
ENDSYNTYPE;
```

Figura 3.7: Diagrama SDL (declaraciones)

Además, en la figura 3.7, por la parte de las declaraciones, se tienen las señales que manejan la información de manera global en cualquier parte de la arquitectura. También, cuenta con los tipos de datos que manejan las señales. En las declaraciones se muestran las señales con el tipo de parámetro que lleva, si es que llevan.

Por otro lado, el proceso entre los dispositivos y el controlador, se puede observar que se encarga de manejar la información que se transmite entre los dispositivos y el controlado general, ya que pueden haber errores o la información tomada.

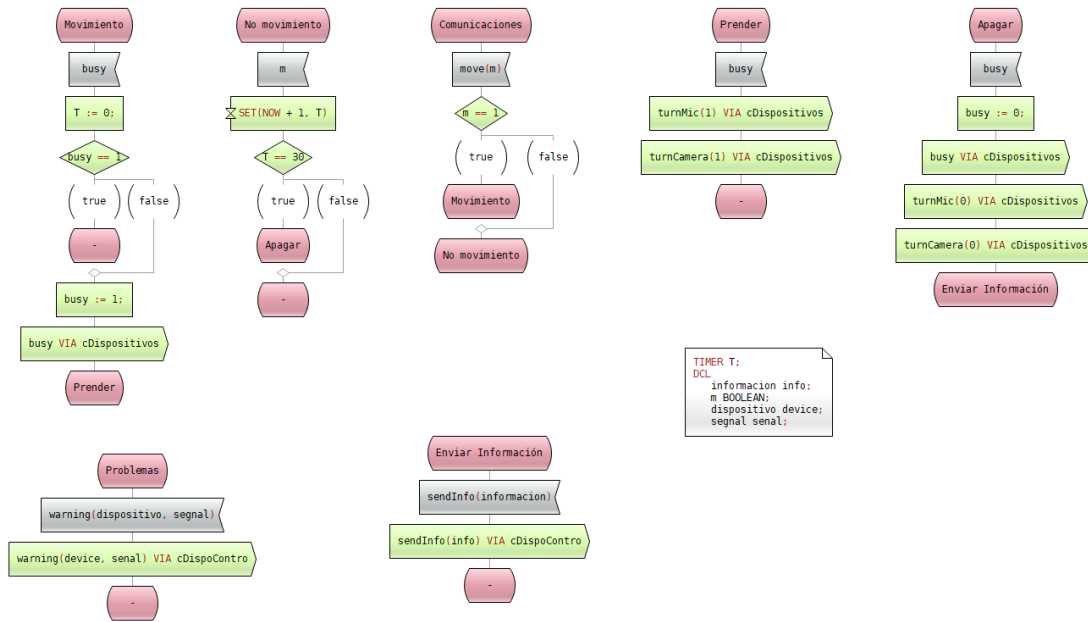


Figura 3.8: Diagrama SDL Proceso Dispositivos-Controlador

En la figura 3.8 se puede ver que hay distintos estados en donde se maneja el paso de la información (problemas o datos grabados), además se tiene un estado para determinar si un animal se encuentra en movimiento o no, con esto se pueden grabar con mayor facilidad los datos y no va a haber problemas con otras señales ya que el canal va a estar ocupado. Entonces, si el canal se encuentra ocupado es porque los dispositivos están grabando, está desocupado en caso contrario.

Para determinar el tiempo que se graba, se tiene un temporizador que va a medir el tiempo de grabación para tomar los datos, cuando se cumpla ese tiempo, los dispositivos se apagan para ahorrar energía, ya que no pueden estar prendidos todo el tiempo.

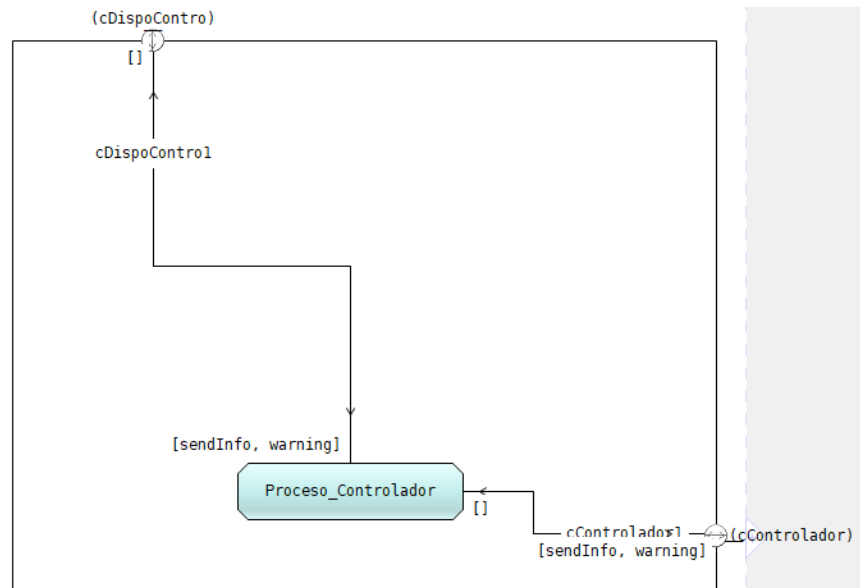


Figura 3.9: Diagrama SDL del Controlador

Dentro del controlador solo se encuentra un proceso que maneja la información que se pasa, ya que la distribuye hasta el servidor. Dentro del proceso se encuentra un diagrama que muestra como recibe la información y como la envía en la figura 3.11.

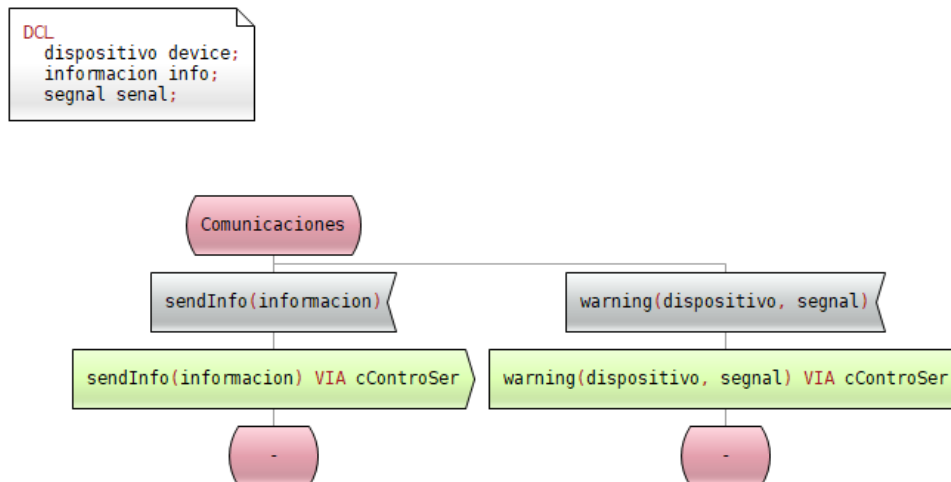


Figura 3.10: Diagrama SDL del Proceso dentro del controlador

Luego, viene el proceso que conecta el controlador con el servidor, de hecho, hace el mismo proceso que el proceso dentro del controlador (ver la figura 3.11).

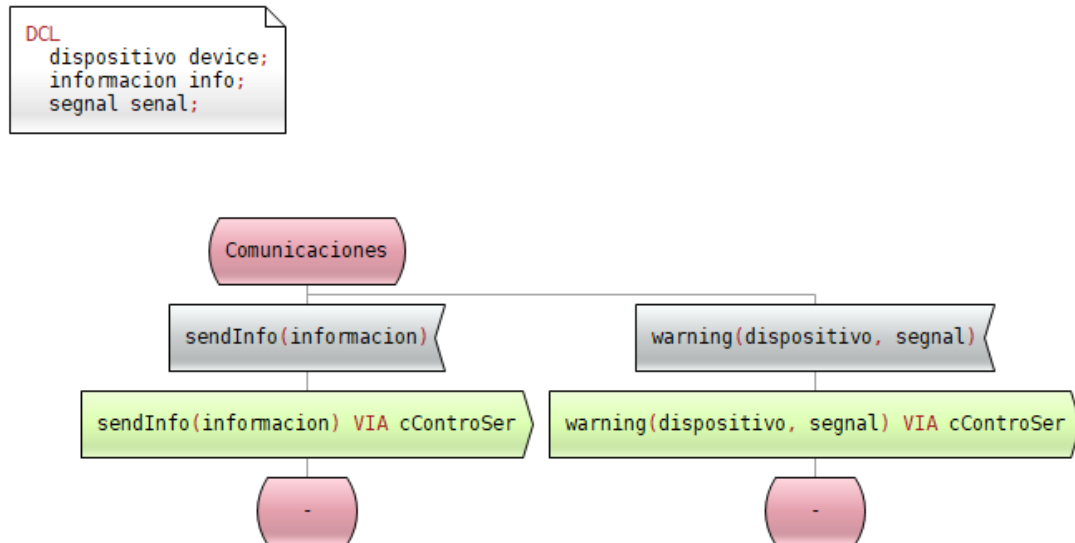


Figura 3.11: Diagrama SDL del Proceso que conecta el controlador con el Servidor.

Ahora bien, en la parte del servidor encuentra un único proceso que conecta con tres secciones: administrador, controlador y la nube (ver la figura 3.12).

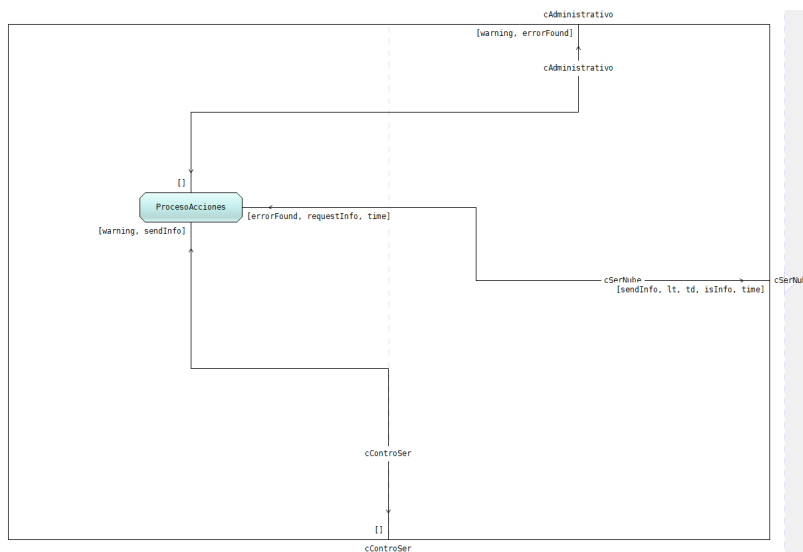


Figura 3.12: Diagrama SDL del Servidor

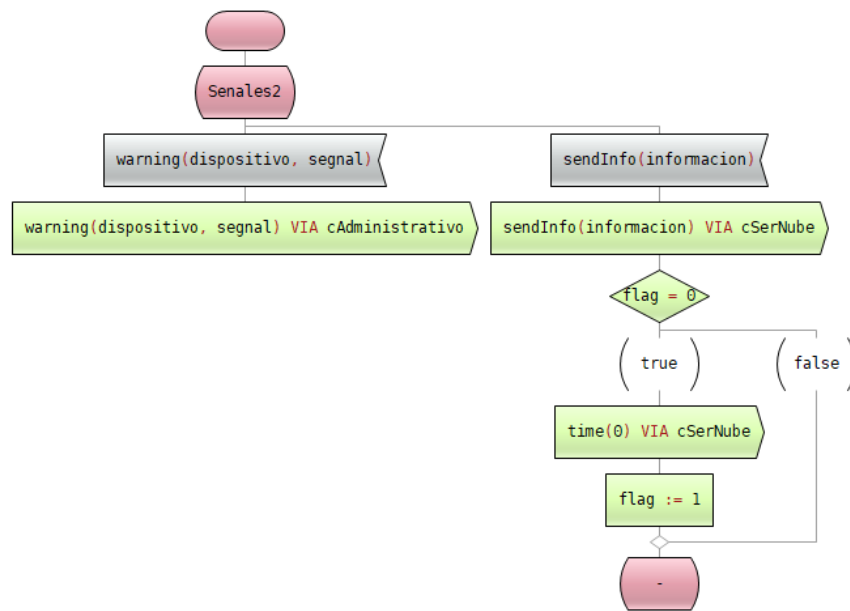


Figura 3.13: Diagrama SDL del Proceso dentro del Servidor (parte 1)

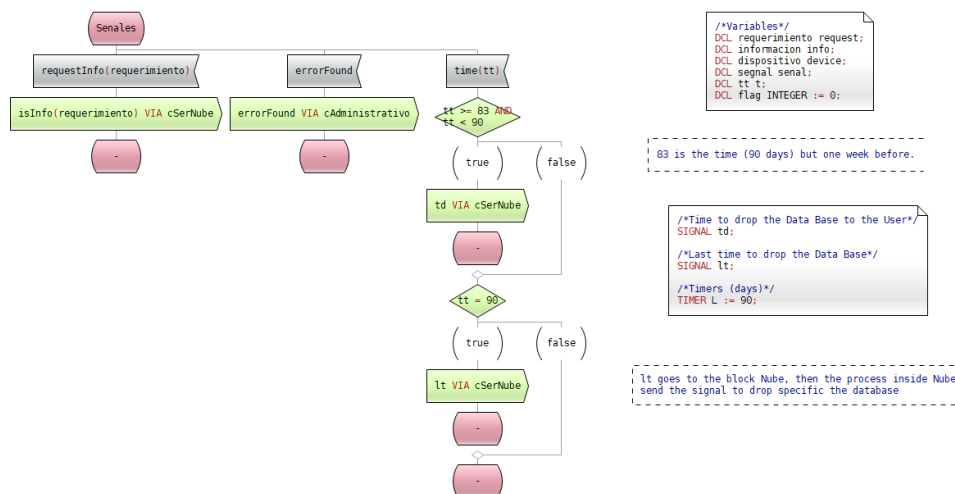


Figura 3.14: Diagrama SDL del Proceso dentro del Servidor (parte 2)

En las figuras 3.13 y 3.14 se muestran los estados para pasar las señales (errores o datos) hasta administrador o hasta la nube. Además, el servidor se encarga de contar el tiempo (periodo) con el que cuentan los usuarios para hacer uso del sistema, ya que después de cumplir el tiempo la base de datos eliminará la información que tiene almacenada.

Es por eso, que el servidor se encarga de avisar al usuario cuando le quede una semana, con esto el usuario cuenta con una semana para descargar la información o guardarla en otra parte, si en el tiempo en el que tuvo para eso no descargó la información ya no tendrá acceso ni oportunidad para obtenerlas. Esto se hace con el propósito de que la base de datos no se llene en

algún momento y se pueda prevenir esos errores, para seguir almacenando la información sin problemas.

Cuando se cumpla el tiempo de una semana antes de que se venza el periodo que tienen se va a enviar una señal al usuario para avisar que se va a eliminar, cuando se venza todo el plazo después de una semana de anticipación, se envía una señal a la base de datos para borrar la información.

En la figura 3.15, se ve el proceso que conecta el servidor con el administrador, el cual se encarga solamente de pasar la información hasta el administrador.

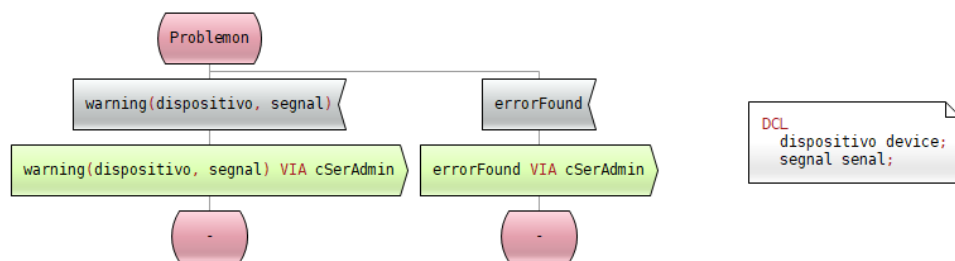


Figura 3.15: Diagrama SDL del Proceso entre el Servidor y el Administrador

En las figuras 3.16 y 3.17, se ve el proceso que está dentro del administrador y lo que se encarga de hacer, que es el paso de la información hacia el mundo (internet) y lo que manejan como administradores, que reciben los errores y logs de la nube.

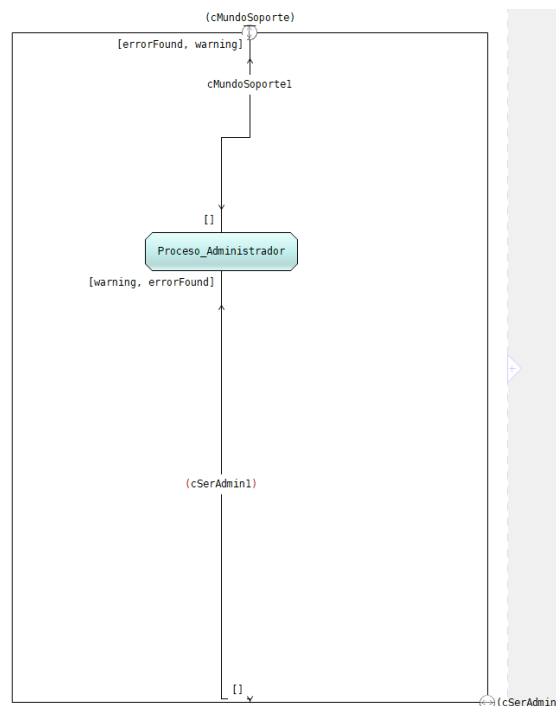


Figura 3.16: Diagrama SDL del Administrador

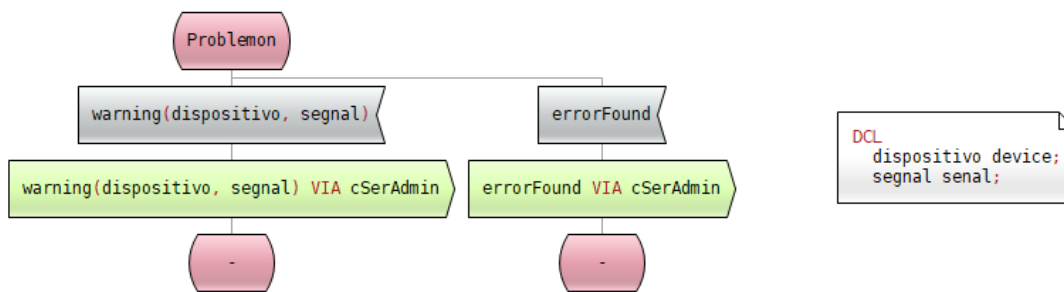


Figura 3.17: Diagrama SDL del Proceso dentro del Administrador

Por último, se encuentra el bloque de la nube (ver la figura 3.18), este bloque contiene un proceso para manejar la información entre el servidor y el usuario con la base de datos. Además, la nube con tiene más cosas que se pueden hacer con los datos en un nivel más específico como el uso de Inteligencia Artificial, sin embargo, no es necesario hacerlo para el contexto de este proyecto.

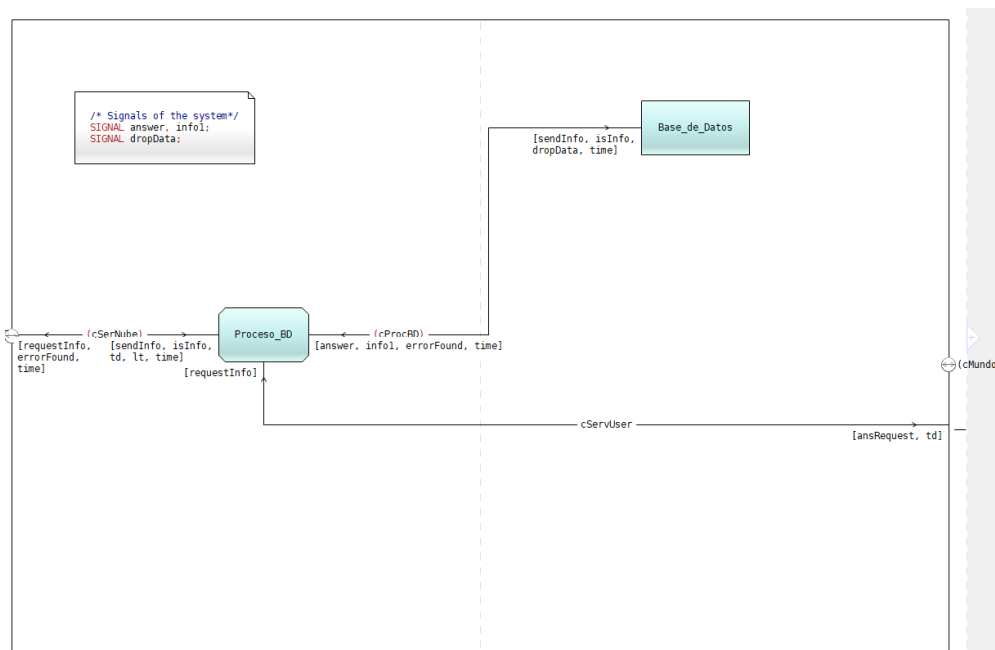


Figura 3.18: Diagrama SDL de la Nube

Dentro del proceso (figura 3.19) se reciben señales y envían a los diferentes canales dependiendo de la señal ya sea para guardar la información, para avisar al usuario de borrar la base de datos, para borrar la base de datos o para verificar una petición que llegue por parte del usuario, pero, cuando se trata de la petición esta puede tener dos valores, que sea falsa o verdadera, en caso de ser verdadera retorna al usuario lo que solicitó y en caso de ser falsa retorna el error al usuario y al administrador.

En la última parte, se encuentra la base de datos que contiene un proceso

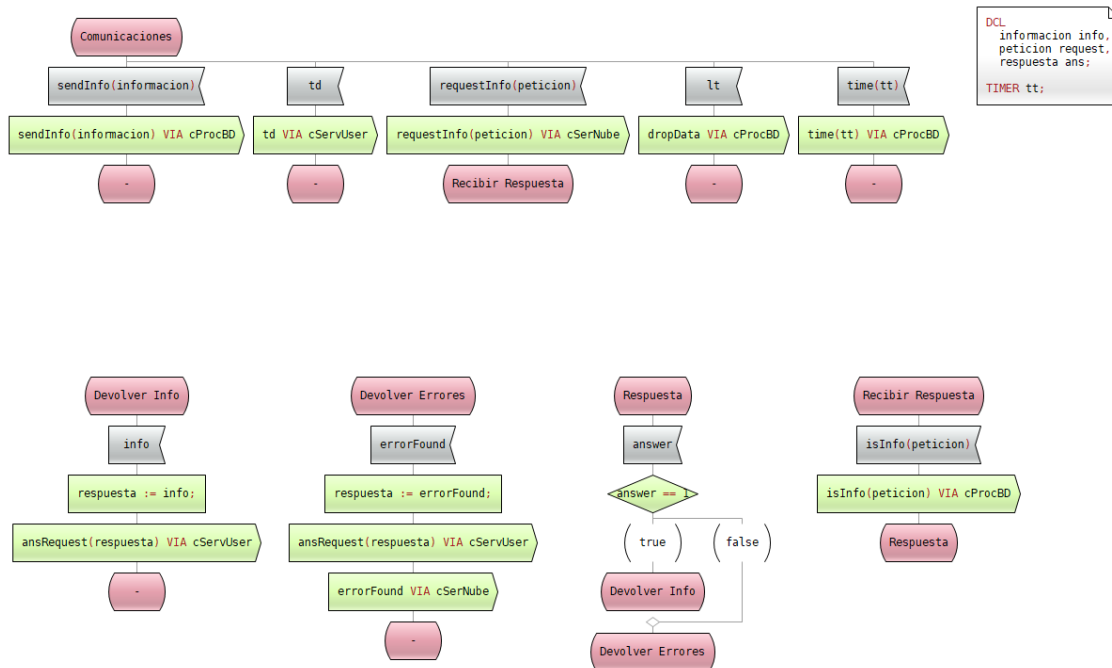


Figura 3.19: Diagrama SDL del Proceso dentro de la Nube

para responder a la información que se pide (figura 3.20). En dicho proceso, se guarda la información que se recibe y se da una respuesta dependiendo si el requerimiento es correcto o no y además, el tiempo que se cuenta para evaluar el periodo (ver figura 3.21).

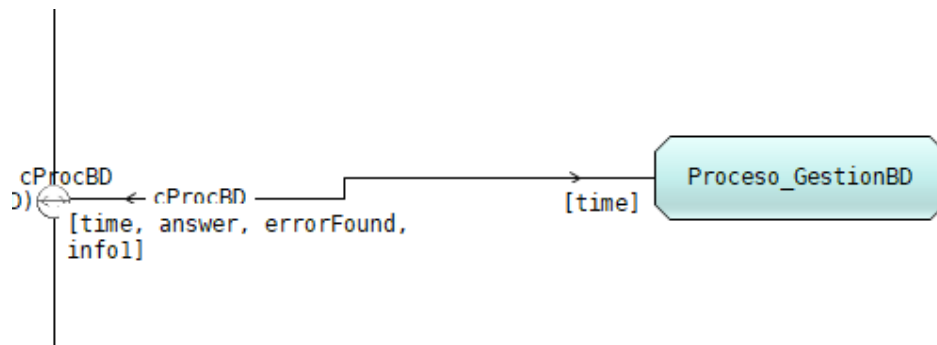


Figura 3.20: Diagrama SDL de la Base de Datos

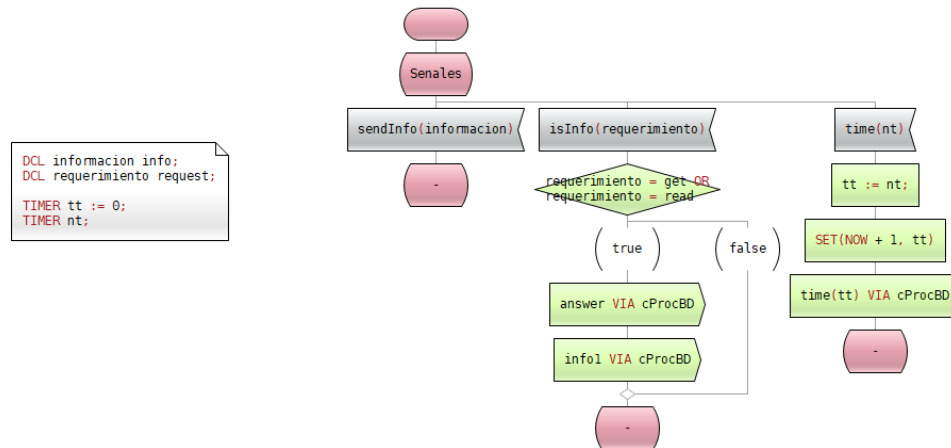


Figura 3.21: Diagrama SDL del Proceso dentro de la Base de Datos

Aunque se muestra el diagrama SDL, aún tiene fallas y faltan cosas por terminar para hacer el plan de pruebas, ya que al no poder solucionar los errores no se hizo la ejecución del plan de pruebas.

3.4. Verificación del sistema base

Presente el plan de pruebas (escenarios y casos de prueba) y los resultados de las diferentes pruebas realizadas al diseño (trazas de ejecución MSC) con los que se muestra la validez del diseño frente a los requerimientos definidos previamente.

tst Plan								exec	
Nombre	Modulo	ID	pre	inputs	post	Salida	obs	result	obs
Guardar Info BD (aceptado)	Base de Datos	BD01	S01	Audio, Video	-	-	Se Guarda la info en la BD	NAN	-
Guardar Info BD (rechazado)	Base de Datos	BD02	S01	Audio, Video	-	-	No se guarda la info en la BD	NAN	-
Esta la Info	Base de Datos	BD03	S02	User_request	S03	Info	La consulta fue exitosa	NAN	-
No esta la Info	Base de Datos	BD04	S02	User_request	S04, S05	Error, "Error found in DB"	La consulta no fue exitos	NAN	-
Drop DB	Base de Datos	BD05	S10	SQL Sentence	-	-	Se hace el drop en la BD	NAN	-
Movimiento Sensor	Controlador	C01	-	mov	C03, C04	-	Se activa la señal del movimiento	NAN	-
Apagar movimiento	Controlador	C02	C01	time, ~mov	C05, C06, C07	-	Se apagan la camara y microfono	NAN	-
Prender Microfono	Controlador	C03	C01	mov	-	busy, "Mic is recording"	Se prende el microfono y la señal de busy se activa	NAN	-

Figura 3.22: Plan de Prueba 1/4

tst Plan								exec	
Nombre	Modulo	ID	pre	inputs	post	Salida	obs	result	obs
Prender Camara	Controlador	C04	C01	movi	-	busy, "Camera is recording"	Se prende la camara y la señal de busy se activa	NAN	-
Apagar microfono	Controlador	C05	C02	~mov	C07	~busy, "Mic off", Audio	Se apaga el microfono y la señal de busy se desactiva	NAN	-
Apagar Camara	Controlador	C06	C02	~mov	C07	~busy, "Camera off", Video	Se apaga la camara y la señal de busy se desactiva	NAN	-
Enviar Info (aceptado)	Controlador	C07	C02, C05, C06	Audio, Video	S01			NAN	-
Enviar Info (rechazado)	Controlador	C08	C02, C05, C07	Audio, Video	-	-	Se espera que no haya error a la hora de enviar la información	NAN	-
Error Camara	Controlador	C09	-	Problem	C12	warning	La camara tiene algun error	NAN	-
Error Microfono	Controlador	C10	-	Problem	C12	warning	El microfono tiene algun error	NAN	-

Figura 3.23: Plan de Prueba 2/4

tst Plan								exec	
Nombre	Modulo	ID	pre	inputs	post	Salida	obs	result	obs
Error Sensor de Movimiento	Controlador	C11	-	Problem	C12	warning	El sensor de movimiento tiene algun error	NAN	-
Error dispositivos (Controlador)	Controlador	C12	C09, C10, C11	warning, device	S06	warning, device	El controlador envia el problema al servidor	NAN	-
Request Info	Dispositivo de Cómputo	DC01	-	User_request	S02	-	El usuario pide una información	NAN	-
Guardar Info	Servidor	S01	C07	Audio, Video	BD01	-	Se envia lo coleccionado a la BD	NAN	-
Is Info	Servidor	S02	DC01	User_request	BD03, BD04	-	El servidor solicita la infor del usuario a la base de datos	NAN	-
AnsRequest	Servidor	S03	BD03	Info	-	-	Se le muestra al usuario la información solicitada	NAN	-
Ans Request Error	Servidor	S04	BD04	Error	-	-	Se le muestra al usuario la información del error	NAN	-
Admin Error Request Usuario	Servidor	S05	BD04	"Error found in DB", Error	-	-	Se le muestra al admin la información del error	NAN	-

Figura 3.24: Plan de Prueba 3/4

tst Plan								exec	
Nombre	Modulo	ID	pre	inputs	post	Salida	obs	result	obs
Error dispositivos (Servidor)	Servidor	S06	C12	warning, device	-	-	Se le muestra al admin el error del dispositivo	NAN	-
Time to drop	Servidor	S07	-	time	S08	-	Llega la ultima semana de la suscripción	NAN	-
Time to drop2	Servidor	S08	S07	time	-	"The Data will erase from the BD, please save it"	Se le envia al usuario mensajes diarios de que la información va a ser eliminada	NAN	-
Last Time to Drop	Servidor	S09	-	time	S10	-	Llego el ultimo dia de suscripción	NAN	-
Drop Time	Servidor	S10	S09	-	BD05	-	Se envia la secuencia de drop a BD	NAN	-

Figura 3.25: Plan de Prueba 4/4

3.5. Prototipado en C

Para el proyecto se realizó un primer prototipo en el lenguaje C con diferentes hilos de procesamiento. Estos hilos se separaron en las diferentes funcionalidades del proyecto, dichas funcionalidades fueron divididas en 8 módulos (tabla 3.1), donde cada módulo tiene sus estados y señales (mensajes), los cuales fueron de utilidad para poder realizar la comunicación entre los hilos. Para dicha comunicación se creó una cola de mensajes que permitió encolar hasta ocho diferentes instrucciones.

Módulo	Tarea
pServer	Encargado de la comunicación entre el usuario y la información recolectada por los dispositivos.
pCamera	Controlador de la cámara
pMicro	Controlador del micrófono
pMoveSensor	Controlador del sensor de movimientos
pController	Controlador general de los dispositivos de recolección de información
pDatabase	Comunicación con la Base de Datos
pAdmin	Módulo del administrador del sistema para el monitoreo de los dispositivos
pNube	Módulo de comunicación en la nube

Tabla 3.1: Hilos del sistema

Para la realización del proyecto se desarrollaron unas señales. Cada señal se envía por los canales que conectan los procesos con la finalidad de realizar ciertos tipos de acciones y enviar mensajes, donde algunas de estas señales llevan información a través de parámetros. Toda esta información se puede visualizar en la tabla 3.2 a modo de resumen para visualizar las señales que se usaron en la elaboración del prototipo del sistema. Estas señales se crearon al seguir los diagramas SDL.

TO_DEVICES_SIGNAL	TO_DEVICES	TO_SERVER_BD	TO_BD_CLOUD
turnOn	sendInfo(const char*)	requestInfo(const char*)	ansRequest(const char*)
turnOff	move(integer)	errorFound	dropData
move_detected(integer)	busy	tiempo(integer)	
error(const char*, const char*)	warning(const char*, const char*)	answer(integer)	
		info1	
		td	
		lt	
		isInfo(const char*)	

Tabla 3.2: Mensajes del sistema con sus parámetros

Dentro del código se uso una estructura para la realización del paso de mensaje a través de los distintos canales. Esta estructura se compuso de cuatro atributos. El primer atributo fue el nombre de la señal identificado por el nombre signal. El segundo atributo fue un valor numérico para aquellas señales que envían un entero como tipo de dato indentificado como value. El tercera atributo fue el nombre del dispositivo como una cadena de caracteres (este atributo es importante para la señal de warning en el canal TO_DEVICES y la señal de error en el canal TO_DEVICES_SIGNAL) identificado como device. Por ultimo, el cuarto atributo fue una información en formato de cadena de texto indentificado como info.

```

1 typedef struct {
2     int signal;
3     int value;
4     char device[20];
5     char info[150];
6 } msg_t;

```

Para una segunda entrega del prototipo se pudo visualizar una mejora con respecto a esta estructura. Para ello, se piensa realizar un cambio en el segundo atributo value para que este sea un tipo de dato void*. De esta manera el valor podrá tomar cualquier tipo de dato (la identificación del tipo de dato se hará a través del nombre de la señal). Para aquellas señales que usan mas de un parámetro se piensa realizar una estructura con dos atributos correspondientes al dispositivo y la información.

```

1 typedef struct {
2     int signal;
3     void* value;
4 } msg_t;
5
6 typedef struct {
7     char device[20];
8     char info[150];
9 } Info_Data;

```

Después de haber finalizado de escoger la estructura que represento a los mensajes dentro del prototipo, se prosiguió a codificar las señales con un valor numérico. Como resultado de la codificación se obtuvo los valores descritos en la tabla 3.3.

TO_DEVICES_SIGNAL	TO_DEVICES	TO_SERVER_BD	TO_BD_CLOUD
turnOn (0)	sendInfo (0)	requestInfo (0)	ansRequest (0)
turnOff (1)	move (1)	errorFound (1)	dropData (1)
move_detected (2)	busy (2)	tiempo (2)	
error(3)	warning (3)	answer (3)	
		info1 (4)	
		td (5)	
		lt (6)	
		isInfo (7)	

Tabla 3.3: Señales del sistema con su identificador en el código

El siguiente paso en la elaboración del prototipo fue la codificación de los estados dentro de los procesos. Previamente, en el SDL, en la comunicación del sistema se pudo observar la creación de estados dentro de los procesos. Para la elaboración de dichos estados, se crearon seis estructuras dentro del prototipo que permitieron seguir el flujo de trabajo establecido en los diagramas. Es importante recalcar que tres de los procesos del prototipo comparten la misma funcionalidad, por esto no se elaboraron ocho estructuras de estados. Para este paso, también se realizó una codificación que se puede observar en las tablas 3.4 y 3.5.

PROCESO_CONTROLADOR_STATES	PROCESO_SERBDNUBE_STATES	PROCESO_DEVICES_STATES
Comunicaciones_dispocontrol (0)	Comunicaciones_serBDNube (0)	Comunicaciones_devices (0)
Apagar (1)	RecibirRes (1)	
Prender (2)	Respuesta (2)	
RecibirInfo (3)	DevolverInfo (3)	
	DevolverErrores (4)	

Tabla 3.4: Estados del sistema 1/2

PROCESO_ACCIONES_STATES	PROCESO_ADMIN_STATES	PROCESO_BD_STATES
Comunicaciones_acciones (0)	Comunicaciones_admin (0)	Comunicaciones_bd (0)

Tabla 3.5: Estados del sistema 2/2

Finalmente, se realizó el código de los ocho módulos definidos en la tabla 3.1, siguiendo los diagramas SDL desarrollados.

Capítulo 4

Escalamiento del sistema

A partir del sistema base modelado en PragmaDev Studio, se construye un nuevo sistema, que considera múltiples instancias de cada agente (proceso SDL).

4.1. Diagramas MSC

A comparación de la primera versión de los diagramas MSC, durante la realización del proyecto se realizaron varias modificaciones para hacer un mejor diseño en la arquitectura del mismo.

El primer cambio es que ahora en vez de usar un diagrama general MSC se hicieron varios diagramas para mostrar todo el funcionamiento de una manera más organizada y correcta, para esto se separó el diagrama principal en 5 diagramas MSC.

Dentro de los diagramas, para hacer la acción del movimiento, se utiliza el diagrama de la figura 4.1 , el cual solo tiene un fragmento en caso de que el trigger de movimiento se active.

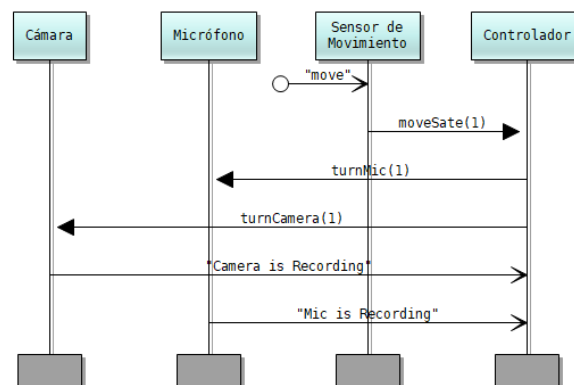


Figura 4.1: Diagrama MSC cuando se registra un movimiento.

Lo mismo aplica en la figura 4.2, la cual muestra el mismo proceso pero cuando ya no se está moviendo algo en la zona registrada, además cuando se envía al controlador este redirige la información al servidor y este envía la información para salvarla en la base de datos.

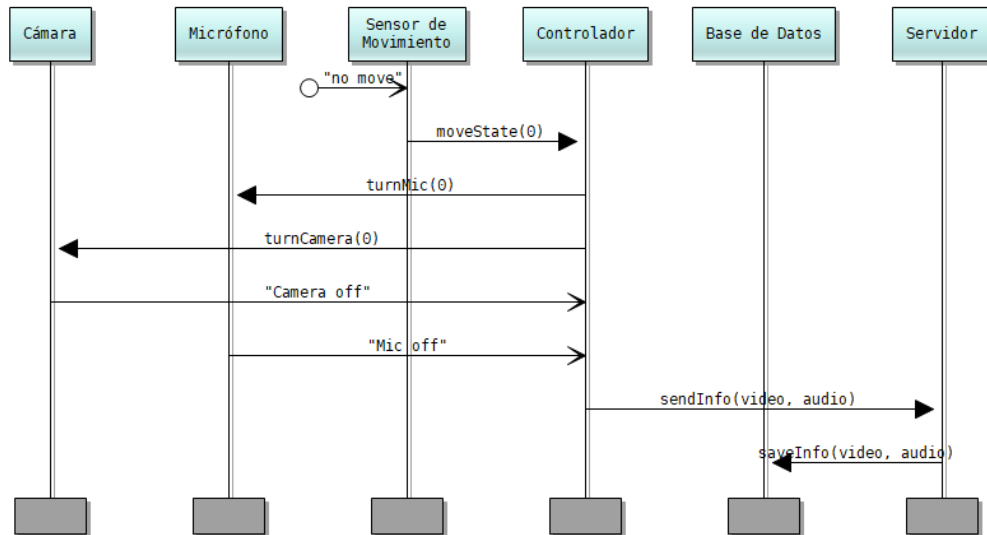


Figura 4.2: Diagrama MSC cuando no se registra un movimiento.

Cuando un usuario va a hacer un requerimiento, este presenta las mismas acciones que el modelo MSC original, como se puede ver en la figura 4.3.

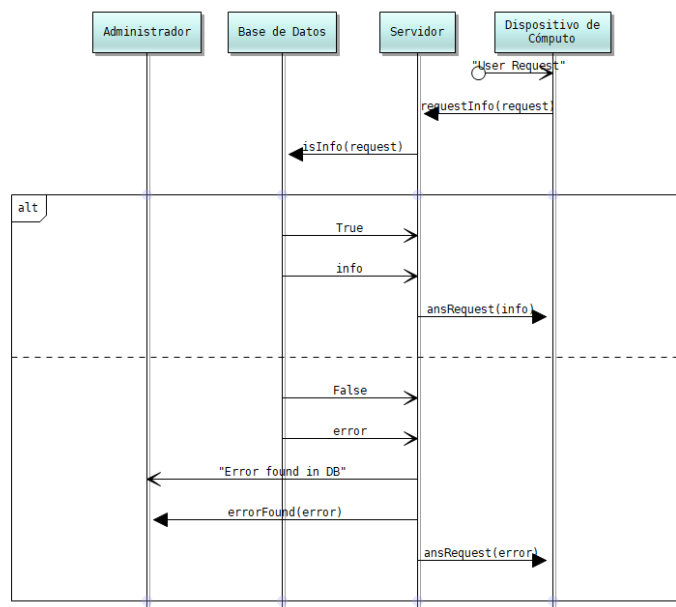


Figura 4.3: Diagrama MSC cuando se registra un requerimiento.

Al momento de que haya un problema en el sistema por parte de los dispositivos, que sería un evento importante y que se puede predecir, no cambió respecto al primer planteamiento del sistema, como se puede ver en la figura 4.4.

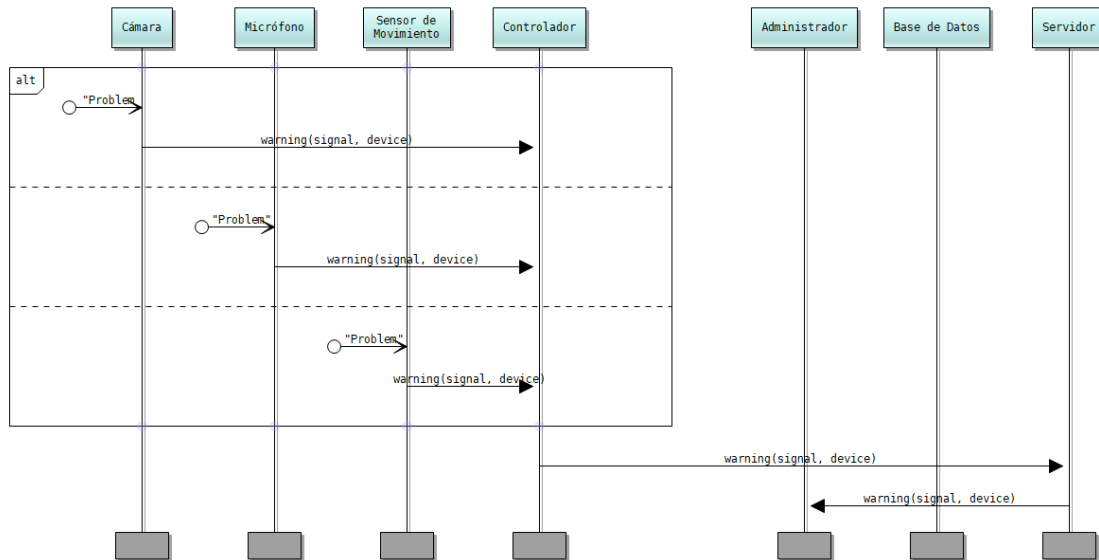


Figura 4.4: Diagrama MSC cuando se registra un problema en el sistema.

Por último, en la figura 4.5 y en la figura noc2, está el caso en el que se elimina la información de la base de datos, esto permite que la información durante el periodo de investigación para los usuarios no se llene la base de datos y no se tenga que utilizar recursos, es por eso que se le envía al usuario un mensaje sobre la información que se eliminará para que descargue la información que necesita y una vez cumplido el tiempo para guardar la información (una semana aproximadamente) se elimina la información de la base de datos, para que el usuario o los siguientes usuarios puedan hacer uso nuevamente de la base de datos y esta no se llene ni tenga problemas porque base de datos se encuentra sin espacio.

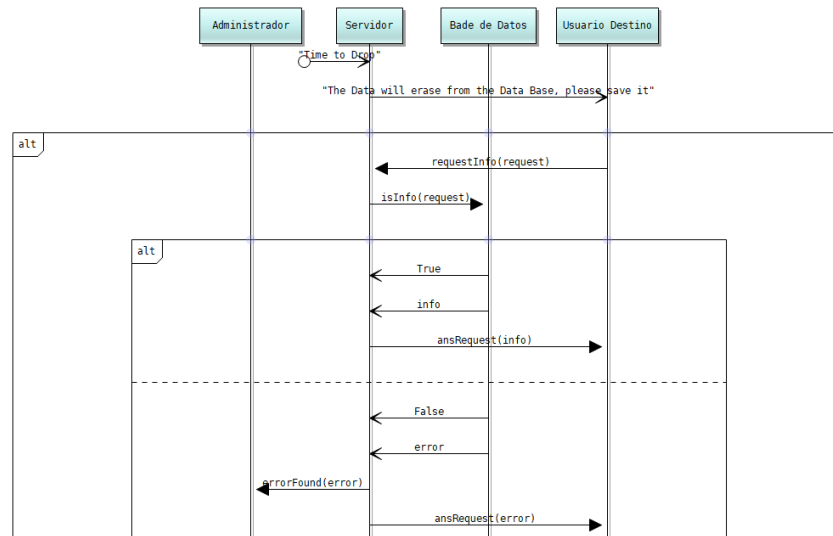


Figura 4.5: Diagrama MSC cuando se va a eliminar la información (1/2).

Por un lado, si el usuario decide traer la información para guardarla localmente una vez le llega el mensaje sobre la información que será eliminada, este envía un requerimiento cuyo manejo ya se mostró anteriormente en la figura 4.3, para esto si el requerimiento es verdadero devuelve la información y si es falso devuelve un error tanto para el usuario como para el administrador en la figura 4.6

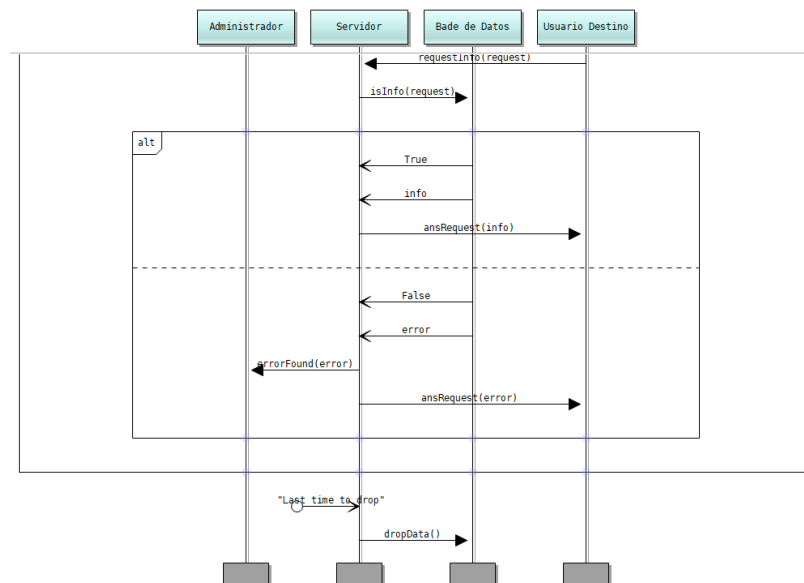


Figura 4.6: Diagrama MSC cuando se va a eliminar la información (2/2).

4.2. Arquitectura SDL y procesos

Incluya diagramas (PragmaDev Studio) con la arquitectura (estructura) del sistema escalado y de los respectivos procesos SDL (comportamiento).

Para la arquitectura SDL se hicieron varios cambios, principalmente fue quitar algunos procesos que no eran necesarios y solo se dejaron bloques en la arquitectura principal, cada bloque tiene un proceso asociado exceptuando el bloque nube, en donde tiene un proceso y otro bloque asociado (base de datos), esta arquitectura se puede ver en la figura 4.7.

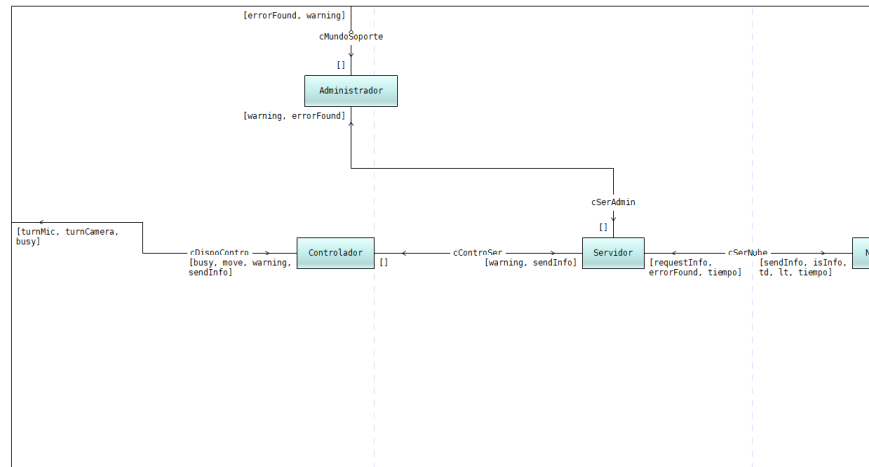


Figura 4.7: Diagrama SDL de la arquitectura general del sistema (1/2).

En la figura 4.8 se encuentra el resto de señales que salen de la nube a internet en general y es de donde también vienen las señales de los usuarios.

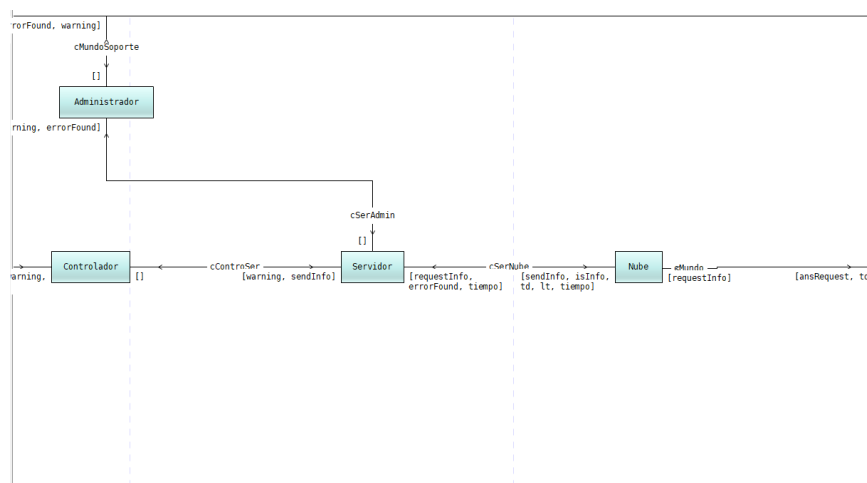


Figura 4.8: Diagrama SDL de la arquitectura general del sistema (2/2).

En la parte del controlador solo se encuentra el proceso que administra las entradas, salidas y los procedimientos que tiene como se puede ver en la figura 4.9.

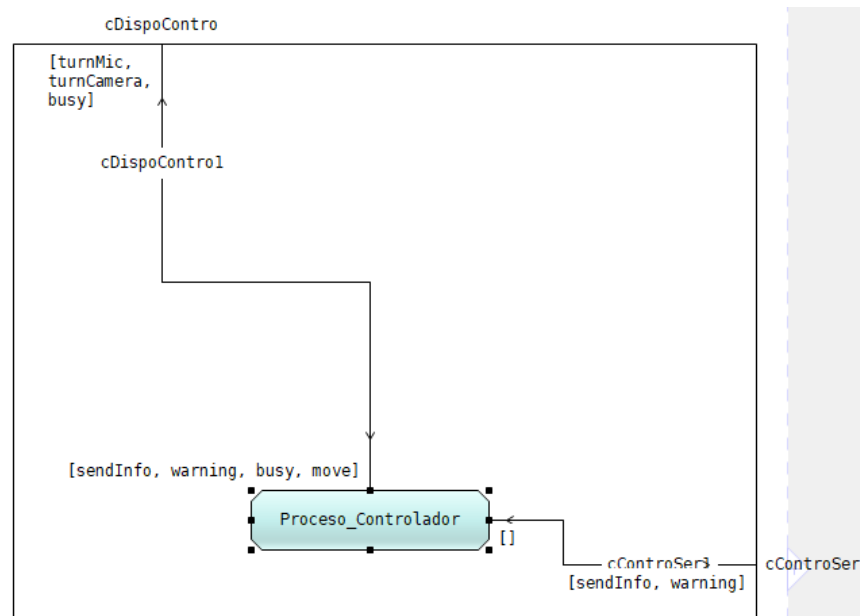


Figura 4.9: Diagrama SDL del controlador.

Dentro de dicho proceso se encuentra la máquina de estados de la figura 4.10. Hubieron muchos cambios con respecto al modelo original, ya que se arreglaron errores del "syntax" se consideraron más casos, juntando los procesos externos entre los dispositivos, el controlador y entre el controlador y el servidor, con la reducción de estos procesos en uno solo que maneja el controlador, refleja un buen comportamiento respecto al diagrama MSC que se estableció al principio.

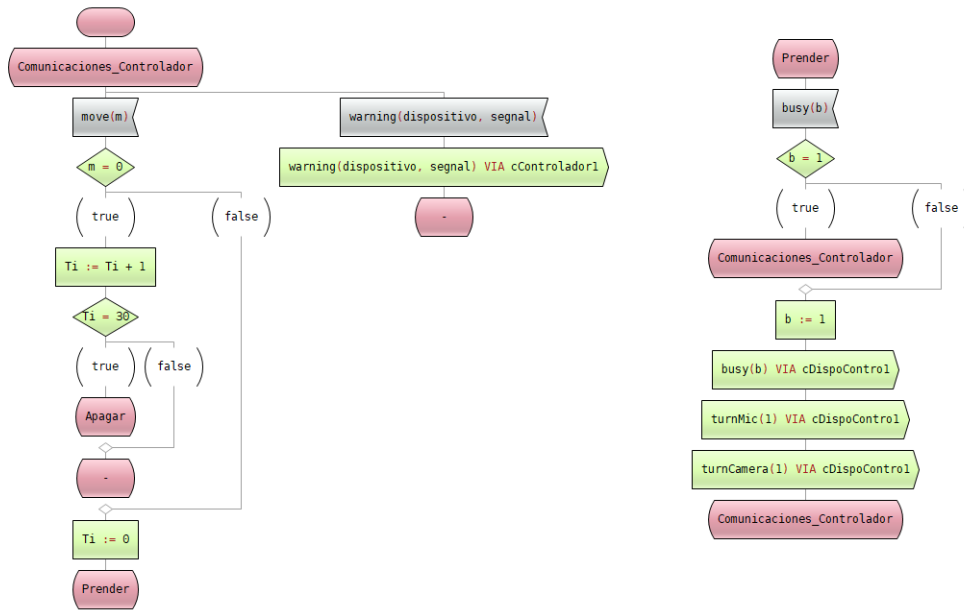


Figura 4.10: Diagrama SDL del proceso del controlador (1/2).

La otra parte de este proceso se muestra en la figura noc, sin embargo, es importante decir que no se están usando timers, sino tipos de variables que simulan temporizadores para medir el tiempo de como si fueran segundos con respecto al rango del movimiento registrado.

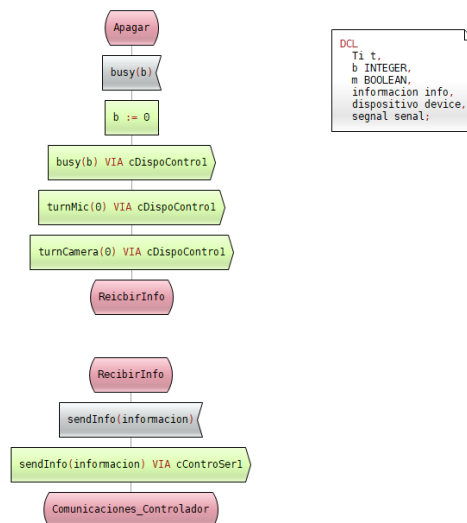


Figura 4.11: Diagrama SDL del proceso del controlador (2/2).

Ahora bien, en la parte del servidor no cambio nada, sigue estando igual al modelo principal dentro del servidor, lo único que cambia es el contenido del proceso dentro del servidor en la figura 4.12 y en la figura 4.13 .

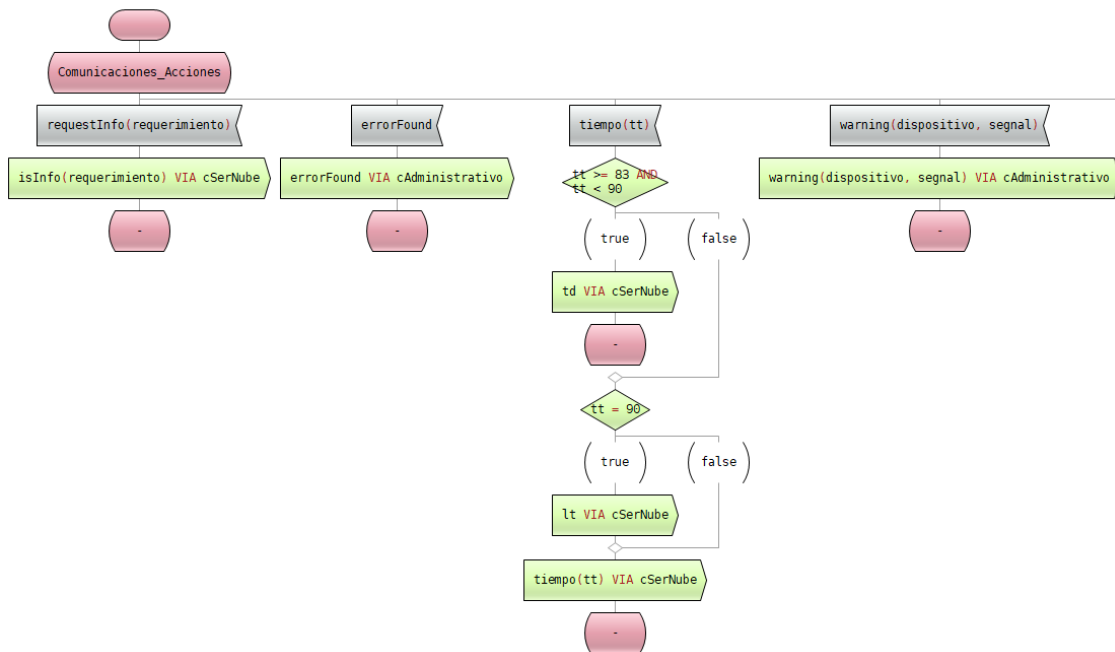


Figura 4.12: Diagrama SDL del proceso del servidor (1/2).

De hecho, en la figura 4.13, se ve como se usan dos estados de *sendInfo* en los cuales el primero

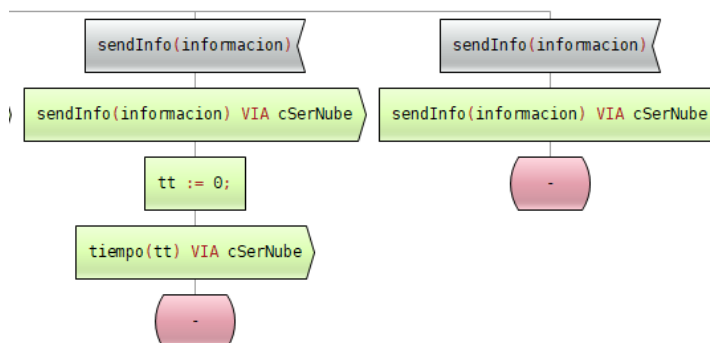


Figura 4.13: Diagrama SDL del proceso del servidor (2/2).

En la parte del administrador tampoco cambió, solo se redujeron procesos y se usó el mismo del planteamiento inicial de la arquitectura. Por otro lado la nube cambió ya que se incluyeron más cosas por el timer y se solucionaron otros errores en las entradas y salidas de los procesos. La arquitectura del servidor es la siguiente figura 4.14.

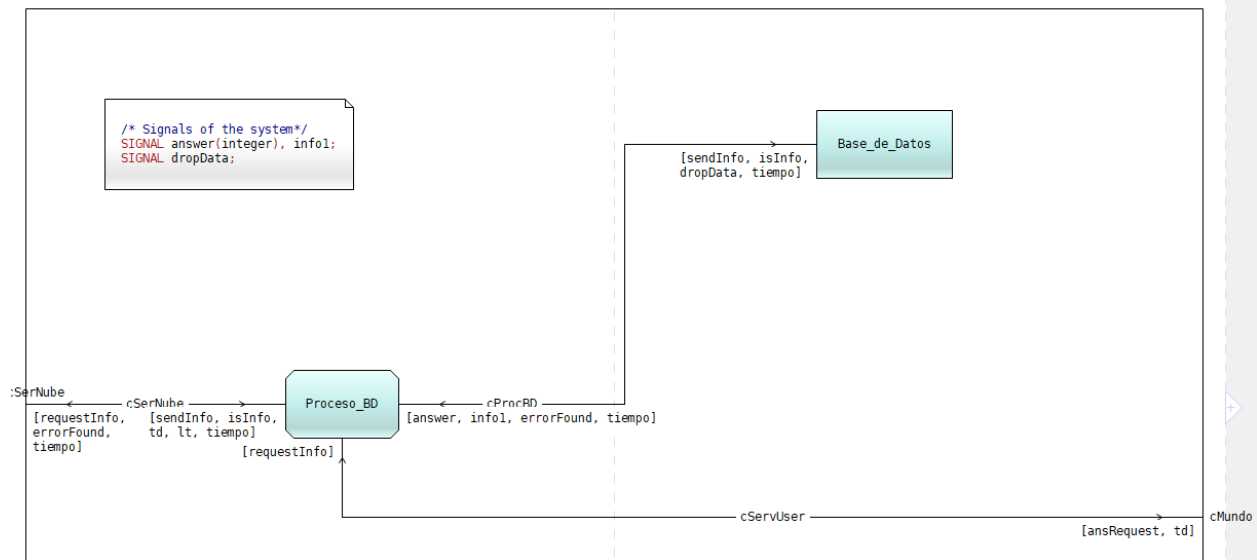


Figura 4.14: Diagrama SDL de la nube.

Dentro del proceso que conecta el servidor con la base de datos se ve lo siguiente en la figura 4.15.

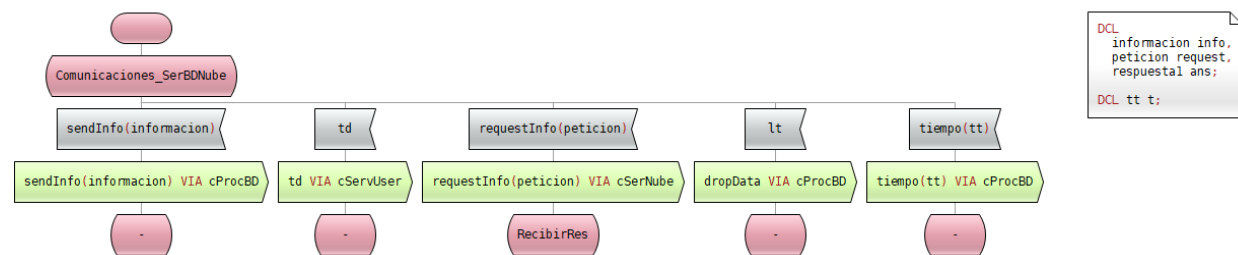


Figura 4.15: Diagrama SDL del proceso que conecta la nube con el servidor (1/2).

Donde los estados para recibir las respuestas de la base de datos para enviarlos y distribuirlos dependiendo si son errores o información correcta obtenida del requerimiento. De hecho, para el usuario se devuelve un tipo de respuesta dependiendo si es un error o si es una consulta válida (la información solicitada), si hay un error se el devuelve al usuario el error y al servidor para redirigirlo al administrador, esto se ve en la figura 4.16.

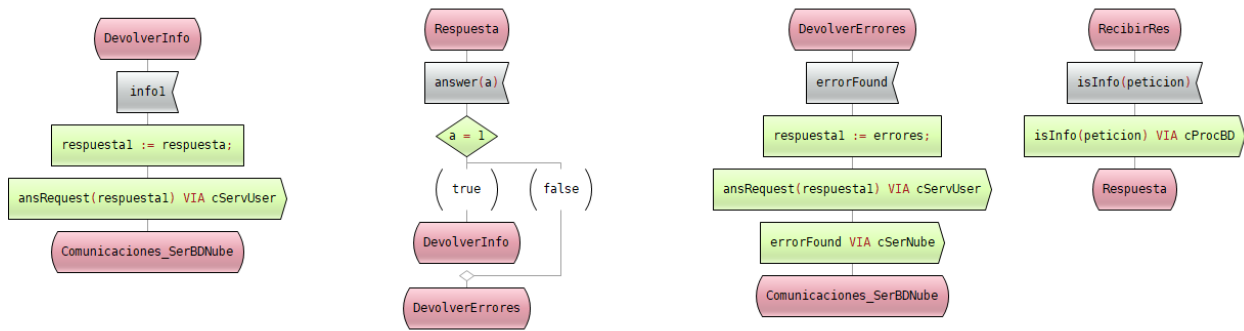


Figura 4.16: Diagrama SDL del proceso que conecta la nube con el servidor (2/2).

Por último, el proceso que maneja la base de datos se encarga de procesar el tiempo que se inicia desde el servidor y este lo suma y lo vuelve a enviar, esto con el propósito de revisar el tiempo en el servidor, realmente el manejo del tiempo se puede hacer en otro proceso, pero la decisión de elegir el proceso que hiciera esto fue arbitraria y sin preferencia, por decisión de los involucrados en el proyecto se escogió el proceso que maneja la base de datos, de hecho, una posible mejor es hacer un proceso que maneje la parte del tiempo internamente en el servidor. El proceso que gestiona la base de datos se puede ver en la figura 4.17.

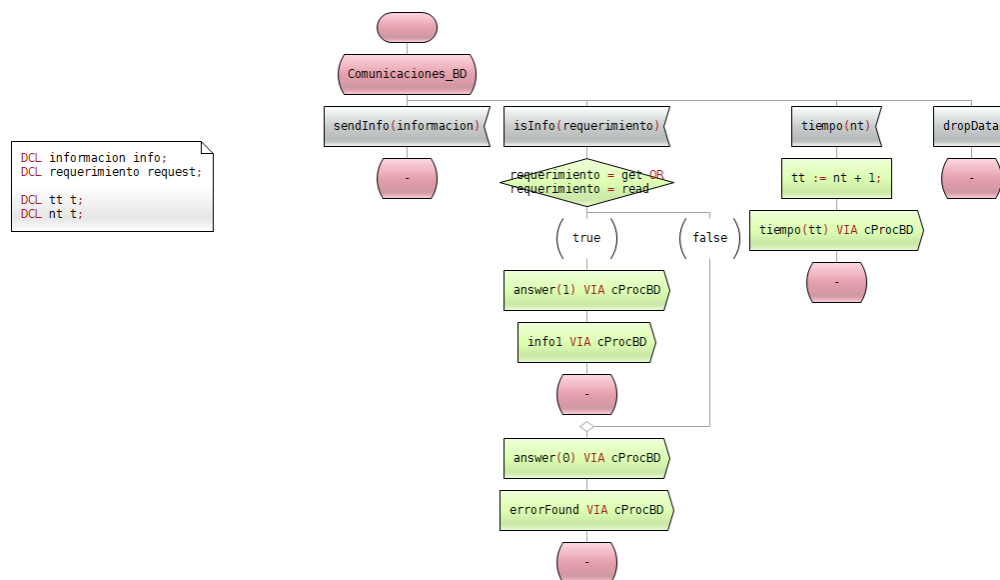


Figura 4.17: Diagrama SDL del proceso que gestiona la base de datos.

Adicionalmente, se establecieron las declaraciones de las señales y los tipos de datos generales de la arquitectura en la figura 4.18.

```
/* Signals of the system */  
SIGNAL move(boolean), warning(device, senal);  
SIGNAL sendInfo(info);  
SIGNAL requestInfo(request), isInfo(request);  
SIGNAL busy(integer), tiempo(t);  
SIGNAL ansRequest(ans);  
SIGNAL errorFound;  
SIGNAL turnCamera(boolean), turnMic(boolean);
```

```
/* Definition of the types */  
NEWTYPE info  
  LITERALS video, audio;  
ENDNEWTYPE;  
  
NEWTYPE device  
  LITERALS camera, microphone, sensor;  
ENDNEWTYPE;  
  
NEWTYPE senal  
  LITERALS batery, externo, almacenamiento;  
ENDNEWTYPE;  
  
NEWTYPE request  
  LITERALS get, read;  
ENDNEWTYPE;  
  
NEWTYPE ans  
  LITERALS errores, respuesta;  
ENDNEWTYPE;  
  
SYNTYPE t = INTEGER  
  DEFAULT 1  
  CONSTANTS 1:10  
ENDSYNTYPE;
```

Figura 4.18: Diagrama SDL de las declaraciones generales de la arquitectura.

Capítulo 5

Reflexión del proceso

Conclusión Dilan: Parcialmente, pese a las complicaciones se evidencia un avance en el desarrollo del proyecto. Ha sido importante comprender que la implementación de un sistema no solamente se puede ver desde la observación computacional o el funcionamiento interno del sistema sino que debe ser detallado desde cada aspecto por más mínimo que sea. Como por ejemplo, en su momento no se había tenido en cuenta el funcionamiento básico de los dispositivos, y aunque es cierto que estos pueden presentar fallas y batería baja en algún momento del transcurso del sistema, es por eso que es necesario ahorrar batería en su mayor posibilidad, por diferentes motivos, más allá de los económicos, que también juegan un papel importante. Aún quedan cosas por refinar y mejorar para la presentación final de sistema, espero y confío en nosotros como equipo que vamos a poder sacar adelante lo que resta y poder entregar un sistema funcional y eficiente.

Conclusión Luis: Dentro de toda la realización del proyecto, a medida que se desglosa en partes más específicas, comienzan a surgir más complicaciones, ya que se tienen que definir más cosas y hay que tener en cuenta más cosas a la hora de hacer los diagramas. A comparación de los diagramas MSC, el diagrama SDL representa la arquitectura del sistema más específica sobre como se comunicaría todo el sistema, sin embargo, por cuestiones de tiempo no se pudo realizar las ejecuciones del plan de pruebas y no se pudo implementar de manera correcta el diagrama SDL, ya que tiene errores al momento de ejecutarlo. Por lo cual, estos problemas serán solucionados para la siguientes entregas y se presentarán como un escalamiento de los diagramas, ya que serían la versión completa y terminada de toda la idea del sistema.

Conclusión Guido: El proyecto ha tenido un desarrollo relativamente lento, debido a las complicaciones presentadas a lo largo del desarrollo del mismo. Sin embargo, cabe destacar que el valor que tiene el proyecto para el entendimiento en los patrones de los animales dentro de su propio hábitat hacen que los esfuerzos valgan la pena. Debido a esto, se ha tenido que hacer un análisis extenso sobre la arquitectura del sistema para poder garantizar un funcionamiento óptimo y sobre todo garantizar un sistema libre de fallos que permita en todo momento su uso de manera eficiente en temas de costos y mantenimiento. Es en esta misma fase de análisis, donde todavía quedan muchas

cosas para mejorar para lograr los objetivos propuestos y entregar un sistema funcional que permita generar valor dentro de los esfuerzos de la preservación animal.

Conclusión General: El proyecto de sensar la población de animales es prometedor y tiene un gran potencial para mejorar el entendimiento de la vida salvaje en temas de comportamiento y sobre todo los esfuerzos de conservación. El proyecto sigue en desarrollo pero contempla un futuro prometedor, que junto al uso de tecnología de IoT permite la captura de datos en tiempo real para ofrecer un alto valor a la hora de entender los hábitos, patrones de migración y uso del hábitat de los animales. Dicha información podrá ser usado para informar y ayudar en los esfuerzos de conservación animal con el objetivo de la protección de los mismo. Mientras hay todavía mucho trabajo por realizar, el potencial del proyecto hace que el esfuerzo sea valioso y enriquecedor.

Capítulo 6

Costos y presupuesto asociados al proyecto

Hasta el momento, lo que se tiene estimado para la ejecución del proyecto, solamente incluyendo los productos y materiales son los siguientes:

- Cámara inteligente con luz infrarroja:
 - Marca: Victure HC300
 - Resolución de video: 1080p
 - Precio: \$47.99 USD
 - Tienda: Amazon
- Micrófono inteligente:
 - Marca: Rode VideoMicro
 - Patrón Polar: Cardioide
 - Precio: \$59.00 USD
 - Tienda: B&H
- Sensor de movimiento:
 - Marca: SANSI
 - Ángulo de detección: 180 grados
 - Precio: \$17.99 USD
 - Tienda: Amazon
- Baterías para los dispositivos:
 - Marca: Energizer AA
 - Tipo: Alcalina
 - Precio: \$8.99 USD (paquete de 24 unidades)
 - Tienda: Target
- Almacenamiento en una base de datos y nube:
 - Marca: Amazon Web Services (AWS) S3

- Precio: \$0.023 por GB/mes
- Tienda: AWS Detalake
- Computadores para el registro y/o auditoría del sistema:
 - Marca: HP EliteDesk 800 G5
 - Procesador: Intel Core i5-9500
 - Precio: \$1,053.00 USD /Por computador
 - Tienda: CDW
- Servidor:
 - Marca: Dell PowerEdge T140
 - Procesador: Intel Xeon E-2224
 - Precio: \$1,043.49 USD
 - Tienda: Dell
- Lente Sigma 150-600mm f/5-6.3 DG OS HSM:
 - Marca: Canon EF
 - Precio: \$1,399.00 USD
 - Tienda: Adorama
- Conexión a internet de alta velocidad:
 - Marca: Movistar Fibra Óptica
 - Precio: 70 USD/mes
 - Se tiene en cuenta al mejor proveedor que brinde mayor disponibilidad de red y una excelente estabilidad de red.
- Sistema de alimentación ininterrumpida (UPS):
 - Marca: UPS APC Back-UPS 650VA
 - Precio: 85.99 USD
 - Tienda: Amazon

Tenga en cuenta que los precios son actuales, es por esto que con el transcurso del tiempo los precios pueden variar. Se buscaron los mejores dispositivos/productos, que fueran los más accesibles económicamente hablando y que puedan suplir las necesidades del proyecto.

Capítulo 7

Referencias

7.1. Personas

Las personas que fueron consultada para la elaboración de este proyecto fueron:

- El Profesor a cargo de la materia Eugenio Tamura.
- Seis estudiantes de la carrera de Biología entre 8vo y 9no semestre de la Pontificia Universidad Javeriana.

7.2. Proveedores

- Amazon, Inc. - Company Profile, Information, Business Description, History, Background Information on Amazon.com, Inc. Reference for Business. <https://www.amazon.com> (accedido el 22 de abril de 2023).
- B&H Photo Video Digital Cameras, Photography, Computers. <https://www.bhphotovideo.com> (accedido el 22 de abril de 2023).
- Target Corporation - Company Profile, Information, Business Description, History, Background Information on Target Corporation. Reference for Business. <https://www.target.com/c/electronics/-/N-5xtg6> (accedido el 22 de abril de 2023).
- AWS Lake Formation - Build a Secure Data Lake in Days. Amazon Web Services. <https://aws.amazon.com/lake-formation/> (accedido el 22 de abril de 2023).
- CDW Corporation - Company Profile, Information, Business Description, History, Background Information on CDW Corporation. Reference for Business. <https://www.cdw.com> (accedido el 22 de abril de 2023).
- Dell Technologies - Company Profile, Information, Business Description, History, Background Information on Dell Technologies. Reference for Business. <https://www.dell.com/es-co/lp/homepage> (accedido el 22 de abril de 2023).