



DEPARTAMENTO
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico I - Primera Entrega

Ingeniería del Software II (2do cuatrimestre de 2009)

Integrante	LU	Correo electrónico
Castillo, Gonzalo	164/06	gonzalocastillo_086@hotmail.com
Elizalde, Victoria	452/06	kivielizalde@gmail.com
Gonzalez, Sergio	481/06	gonzalezsergio2003@yahoo.com.ar
Page Saal, Martín	315/06	martinpage2001@yahoo.com.ar

En el siguiente documento, se arribará la fase de elaboración del proyecto relacionado con la implementación de un sistema de mediciones telemétricas, que unirá puntos estratégicos del país con el fin de capturar la información necesaria y procesarla, para anticiparse a los diferentes fenómenos meteorológicos. En dicha fase se identificarán y detallarán las funcionalidades principales del sistema y su arquitectura básica. Además se definirá un plan detallado para esta primera iteración del proyecto sin tener en cuenta la iteración de de la fase de inepción ya dada por la cátedra.



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2160 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (54 11) 4576-3359

<http://www.fcen.uba.ar>

Índice general

1. Introducción	1
1.1. Contexto de este documento: Fases de RUP	1
1.1.1. Incepción	1
1.1.2. Elaboración	1
2. Requerimientos funcionales y atributos de calidad	3
2.1. Propósito del sistema	3
2.2. Análisis de entorno y enunciado	4
2.3. Generación y descripción de Casos de Uso	6
2.3.1. Índice de casos de Uso	6
2.3.2. Casos de uso dentro de las TRs:	7
2.3.3. Casos de uso dentro de la EC:	8
2.3.4. Casos de uso dentro del sistema de Monitoreo	10
2.4. Escenarios	12
3. Plan del Proyecto	15
3.1. Estimaciones a los casos de uso del proyecto(Asignación de pesos)	15
3.1.1. Planificación y reunión de lanzamiento	15
3.1.2. Preparación individual y reunión de estimación	15
3.1.3. Ensamblado de tareas y revisión de resultados	19
3.1.4. Armado de planificación de iteraciones	20
3.2. Planificación de la primera iteración	21
3.2.1. División en tareas de los casos de uso para la primera iteración	21
3.2.2. Estimación, duración de tareas asociadas a la primera iteración	23
3.2.3. Generación de dependencias	23

3.2.4.	Definición y asignación de recursos	23
3.2.5.	Estimación y precedencias en el resto de las tareas del proyecto y generación de diagrama Gantt . . .	23
3.2.6.	Diagrama de Gantt	24
3.3.	Evaluación, seguimiento y ajustes a la preentrega	28
4.	Gestión de riesgos	29
4.1.	Identificación de riesgos	29
4.2.	Selección de principales riesgos	32
4.3.	Análisis de los riesgos obtenidos	33
5.	Arquitectura	34
5.1.	Módulos	34
5.2.	Componentes y conectores	36
5.2.1.	Funcionamiento de las Terminales Remotas	36
5.2.2.	Funcionamiento de la Estación Central	37
5.2.3.	Funcionamiento del Sistema de Monitoreo	40
5.2.4.	Comunicación entre la Estación Central y el Sistema de Monitoreo	42
5.2.5.	Comunicación entre la Estación Central y las Terminales Remotas	42
6.	Prueba de Concepto	47
6.1.	Implementación de la prueba de concepto	47
6.1.1.	Descripción del funcionamiento	47
6.1.2.	Archivos fuentes utilizados	48
6.2.	Test de Integridad	48
6.2.1.	Test 01	48
6.2.2.	Test 02	49
6.2.3.	Test 03	49
6.3.	Procesamiento y modelos	49
6.3.1.	Datos y casos de test	50
7.	Informe de avance	51
8.	Bibliografía	52
9.	Conclusión	53

9.0.2. RUP: Fase de Elaboración 53

Parte 1

Introducción

1.1. Contexto de este documento: Fases de RUP

1.1.1. Incepción

La metodología adoptada en este trabajo práctico es RUP, es decir Rational Unified Process. Se partió de la base de un enunciado entregado por los docentes, que representaría en el proceso la primera de sus 4 fases: Inception. Quedaron entonces definidas algunas cosas:

- El alcance del proyecto
- Los otros sistemas con los cuales nuestro sistema debe interactuar (Sistema eólico, Sistema de Facturación, Biggest Satellite, etc)
- La funcionalidad de nuestro sistema (si bien la lista de CU del sistema no venia dada, se hizo a partir del texto del enunciado)
- Algunos conceptos muy generales del negocio que hay detrás del proyecto (el proyecto no tiene objetivos económicos mas allá de la sustentabilidad, el propósito de este sistema es brindarle un servicio fundamental a la población)

1.1.2. Elaboración

Nosotros comenzamos nuestro trabajo en el marco de la segunda etapa de RUP: Elaboration. La fase de Elaboración es una fase crítica dentro de esta metodología ya que sienta las bases de todo el proyecto. Tiene los siguientes propósitos:

- Analizar el dominio del problema
- Establecer una base arquitectónica sólida
- Desarrollar el plan del proyecto
- Eliminar los elementos de mayor riesgo del proyecto

Estos propositos serán enfocados en esta parte del trabajo práctico. De todas maneras, hay que tener en cuenta que esta es la primera iteración de la fase de Elaboración. Esto no significa que se cumplan al 100 % los propósitos de esta fase, sino que se los empiece a atacar. A esta iteración le seguira(n) otra(s) mas de la fase de Elaboración, hasta que se cumplan los objetivos de esta fase.

En esta iteración ya tuvimos un hito que fue la entrega del primer entregable (la preentrega) que contenía la lista de Casos de Uso y El plan del proyecto. Este entregable se corrigió y pasó a formar parte del nuevo entregable, que se entregará al final de esta iteración.

En cuanto al análisis del dominio del problema, la parte funcional está parcialmente cubierta por la lista de Casos de Uso. No se hará más énfasis en ellos durante esta iteración, pero si debería desarrollarse en detalle un porcentaje importante

de ellos en las próximas iteraciones de la fase de Elaboración. Si se dará énfasis en esta iteración a los atributos de calidad: a partir del QAW (Quality Attribute Workshop) obtuvimos los principales atributos de calidad del sistema y las prioridades de cada uno. Serán descriptos con la modalidad de Escenarios de Atributos de Calidad, tal como propone el SEI.

El cronograma del proyecto y la asignación de casos de uso a iteraciones serán ajustados y corregidos, según el feedback obtenido en la preentrega y los cambios que surjan en esta parte de la iteración. Junto con ellos se escribirá un documento de análisis de riesgos y plan de contingencia, formando estas 4 cosas el Plan del Proyecto.

Se sentarán las bases para el diseño de la arquitectura. Se diseñará una arquitectura que se documentará con diferentes vistas (como Componentes y Conectores) siguiendo la notación UML y con lenguaje natural. Para probar esta arquitectura se hará una prueba de concepto (de tamaño reducido) que servirá para verificar que la arquitectura permite soportar ciertas funcionalidades.

En cuanto a eliminar los mayores riesgos del problema, la prueba de concepto nos ayudará a disminuir la probabilidad de que la arquitectura diseñada no cumpla los requerimientos funcionales del proyecto. Sin embargo, probablemente sea necesario tomar acción de mitigación y/o de disminución de la probabilidad de los riesgos en las próximas iteraciones de la etapa. Hay que recordar que no se debe pasar a la siguiente fase de RUP (Construcción) hasta que no estemos totalmente seguros de que la Arquitectura y los requerimientos son estables, los principales riesgos fueron mitigados o desaparecieron y que mi visión del producto es estable.

Parte 2

Requerimientos funcionales y atributos de calidad

2.1. Propósito del sistema

La idea del trabajo práctico consiste básicamente en poder desarrollar un software consistente, que pueda satisfacer no sólo las necesidades básicas de nuestro cliente plasmadas en requerimientos funcionales, sino que también pueda garantizar el cumplimiento de atributos de calidad que constituyen un factor fundamental en el correcto desempeño de nuestro sistema al momento de salir a producción.

El objetivo de la realización de dicho software es en principio, poder predecir, mediante la existencia de terminales remotas con sensores de medición y modelos matemáticos de procesamiento, el estado del tiempo en la totalidad del país, poniendo principal énfasis en generar alertas en caso de tormentas eléctricas y huracanes a tiempo para poder evacuar las zonas afectadas por dichos fenómenos. Sin embargo es muy fuerte la correspondencia entre las funcionalidades a desarrollar para satisfacer dicho objetivo y atributos de calidad críticos en este contexto. Por ejemplo la disponibilidad es un factor fundamental para el cumplimiento del propósito.

En esta parte del informe intentaremos destacar, aquellas funcionalidades necesarias en el sistema, así como los atributos de calidad instanciados en escenarios que se pudieron obtener del relevamiento del contexto.

2.2. Análisis de entorno y enunciado

Al leer el enunciado, una de las primeras veces, surgían nuevas terminologías y se mencionaban tecnologías no conocidas, que en este proyecto tenían gran importancia. Por este motivo fue necesario investigar ciertos aspectos, para introducirse en el tema y poder comprender mejor el dominio del problema al que nos estábamos enfrentando.

Una vez hecho esto, se pudo identificar con más facilidad los requerimientos primarios del sistema y pudimos observar además del gran tamaño del proyecto, que existían ciertos factores de complejidad y disponibilidad en cuanto a los datos y a los modelos que se necesitaban manejar. Por esta razón, pensamos que no era una buena idea, tener todo el sistema centralizado y se comenzaron a identificar partes que realicen diferentes tareas de forma tal de distribuir y organizar mejor la funcionalidades. Por esta razón se decidió modularizar la Estación Central y las Terminales Remotas. Dentro de la Estación Central se pudieron identificar varias partes que se desprendían de la descripción del proyecto: Pudimos identificar una parte encargada del procesamiento de los datos que llegaban de las TRs, así también del uso de los mismos para realizar las predicciones; Otra de las partes era la destinada al intercambio de información con clientes externos o usuarios que monitoreaban el sistema, estos servicios tienen en común que pueden ser realizados mediante web services, uno de los requerimientos para realizar la comunicación con clientes externos; Y un sistema de almacenamiento, con un backup de la información, para evitar la pérdida de datos. En cuanto a la Terminal Remota, se pudo identificar una parte central, encargada de la recepción de los datos enviados por los sensores, y de el envío de estos en el momento oportuno, todo esto con su correspondiente almacenamiento para aquellos datos que fueron enviados, pero no fueron confirmados. Las demás partes son componentes físicos que se agregan a la TR, pero que es bueno identificarlos para tener una mejor visión de la TR, estos son las baterías, los paneles solares, el componente muy usado del mercado, y los sensores. Para tener entonces una idea de las diferentes partes del sistema, se realizó un diagrama contextual mostrando y relacionando las diferentes partes mencionadas. El mismo de alguna forma sirve también como un primer diagrama de arquitectura, ya que, en él se identifican los diferentes componentes físicos y la comunicación entre ellos.

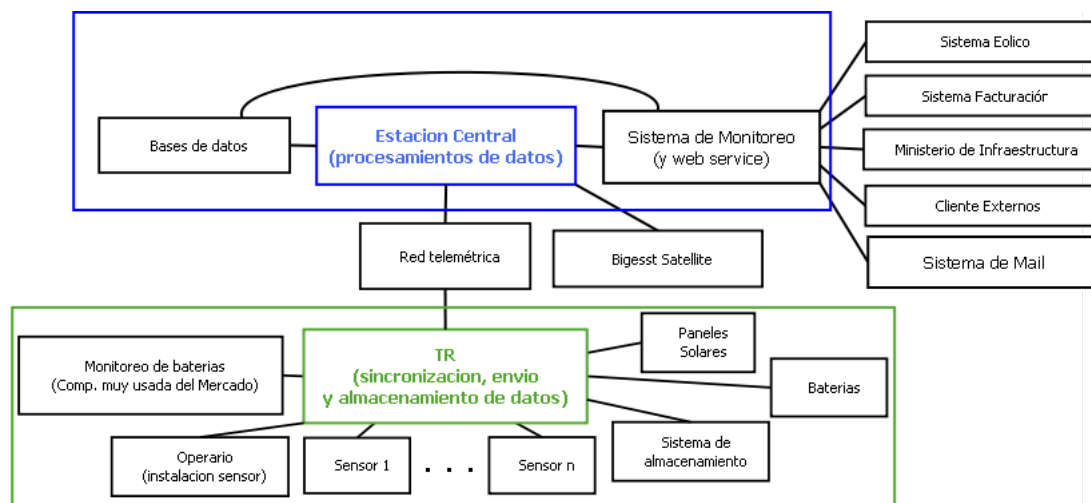


Figura 2.1: Diagrama conceptual del sistema.

Como se puede observar, el sector de la Estación Central, esta compuesta de una parte central, que es la encargada de procesar todos los datos que se reciban de las TRs, verificarlos, y según el caso, identificar aquellos que no sean correctos, o correspondan a un outlier. Como indica la descripción del proyecto, la parte central es la encargada de comunicarse con el sistema Biggest Satellite, cuando se considere necesario obtener información de una zona cuando una TRs se cae y no es posible triangularla con sus TRs vecinas, por este motivo, la estación central esta relacionada directamente con Biggest Satellite. Además de esta tarea, se realizan los cálculos utilizando los modelos matemáticos y los datos almacenados para realizar las predicciones sobre los desastres meteorológicos.

El módulo de Bases de Datos, corresponde al sector encargado de guardar todos los datos que las TRs envíen (previamente, procesados por la parte central) y otros como los resultados de los modelos matemáticos que se computen. Este sistema cuenta con redundancia activa para permitir una rápida respuesta ante las peticiones que se le hagan y además para garantizar la disponibilidad de los datos almacenados.

En el Sistema de Monitoreo, se encuentran las interfaces con todos los clientes y usuarios que utilicen el sistema, es decir, que en este módulo se implementarán las funcionalidades que corresponden con la comunicación de sistemas externos que estén interesados obtener información del sistema central, además de los usuarios que tienen acceso al estado en tiempo real del sistema y puedan realizar cambios, modificando el comportamiento del sistema (por ejemplo, cambiando la agenda

de un sensor).

Con respecto al sector que agrupa los módulos de la Terminal Remota, se encuentra la parte central, encargada de recibir, almacenar y enviar los datos que cada uno de los sensores brinda. Como el protocolo de envío tiene que ser confiable, cada dato enviado es guardado en el Sistema de Almacenamiento hasta que se confirme su llegada, de esta forma se esta contemplando uno de los requerimientos de proyecto, que es el de no tener perdida de datos.

Otra parte importante dentro del sector de la Terminal Remota, es el monitoreo de baterías, que esta representado físicamente por un componente muy usado en el mercado. Este es el encargado de indicarle a la parte central de la TR cuando debe cambiar y utilizar la energía solar, por medio de los paneles solares, y cuando utilizar la energía almacenada dentro de las baterías.

2.3. Generación y descripción de Casos de Uso

En el siguiente texto se mostrarán los casos de uso principales identificados en el sistema, junto con su descripción. En algunos casos, se incluirán también los diferentes actores, que participan en el mismo, éstos se corresponden con los actores que se muestran en el diagrama de contexto del sistema.

2.3.1. Índice de casos de Uso

Terminal Remota - TR

- TR01. Configurando agenda en sensor
- TR02. Obteniendo datos censados
- TR03. Monitoreando/configurando consumo de energía
- TR04. Detectando caída del sensor
- TR05. Detectando recuperación del sensor
- TR06. ABM de sensor
- TR07. Detectando posible caída por falta de energía
- TR08. Recuperando luego de caída por falta de energía
- TR09. Sincronizando datos de censados
- TR10. Encriptando, comprimiendo y enviando mensaje según protocolo
- TR11. Repitiendo mensaje al no tener confirmación
- TR12. Borrando mensaje al llegar confirmación

Estación Central - EC

- EC01. Marcando punto de restauración de datos correctos del sistema
- EC02. Procesando modelos matemáticos
- EC03. Notificando resultado del modelo
- EC04. Guardando datos del modelo
- EC05. Triangulando o reemplazando por BS TR/sensor
- EC06. Detectando alta de TR/Sensor
- EC07. Anunciando remplazo o triangulación de TR/sensor
- EC08. Normalizando TR/sensor y anunciándolo
- EC09. Detectando red congestionada y anunciándolo
- EC10. Detectando outliers y anunciándolo
- EC11. Descartando ó incluyendo posibles outliers
- EC12. Ordenando y descartando mensajes repetidos
- EC13. Confirmando llegada de mensaje
- EC14. Guardando información de uso de Biggest Satellite
- EC15. Definiendo intervalos de confianza para outliers
- EC16. Obteniendo información eólica del sistema del SM
- EC17. Actualizando pautas en los modelos matemáticos
- EC18. Recibiendo, descomprimiendo y desenscriptando mensaje de TR
- EC19. Recibiendo información de Biggest Satellite
- EC20. Actualizando modelos para clientes externos

Sistema de monitoreo - SM

- SM01. Pidiendo datos del uso del servicio de Biggest Satellite
- SM02. Notificando caída de TR
- SM03. Notificando red saturada
- SM04. Enviando datos al Min. de Infraestructura
- SM05. Enviando datos a cliente externo
- SM06. Recibiendo información del sistema eólico
- SM07. Configurando agenda dentro del sensor de una TR
- SM08. Configurando envío de alertas
- SM09. ABM de cliente externo
- SM10. ABM de TR
- SM11. ABM de sensor
- SM12. Actualizando pantallas del monitoreo en tiempo real

2.3.2. Casos de uso dentro de las TRs:

1. TR01:
 - Agentes: Usuario u Operador, Sensor.
 - Descripción: El Usuario u Operador indica que se tiene que modificar la agenda de uno de los sensores de la TR, esta recibe las instrucciones, y modifica dicha agenda utilizando el protocolo de conexión con el sensor.
2. TR02:
 - Agentes: Sensor.
 - Descripción: La TR recibe un dato de un sensor y lo almacena para luego cuando llegue el tiempo correcto lo envíe a la estación central.
3. TR03:
 - Agentes: Componente muy usada del mercado, Batería, Panel solar)
 - Descripción: La TR recibe información periódica del estado de la Batería y del Panel Solar y en base a esa información administra el consumo de energía.
4. TR04:
 - Agentes: Estación Central, Sensor, clock interno.
 - Descripción: Se detecta la caída de un sensor cuando la TR no recibe datos del mismo dentro del tiempo que tiene configurado para recibirlos. Se notifica esto a la EC para que tome las medidas que considere apropiadas.
5. TR05:
 - Agentes: Estación Central, Sensor.
 - Descripción: Se reciben datos de un sensor que no estaba emitiendo (estaba roto). Se lo considera reactivado y se empiezan a almacenar normalmente los datos que este produce. Se notifica a la EC.
6. TR06:
 - Agentes: Sensor, Operario, Estación Central
 - Descripción: Un operario instala un nuevo sensor en la TR y lo configura para enviar datos a la TR. Se notifica a la EC de la presencia de este nuevo sensor. En caso de modificación o baja sucede algo análogo: el operario hace las modificaciones físicas y de configuración pertinentes en la TR y esta comunica el cambio a la EC.
7. TR07:
 - Agentes: Batería y Panel Solar (Componente muy usada del mercado), Estación Central.
 - Descripción: La TR recibe una alerta de batería en nivel mínimo del sensor de batería y una alerta de radiación solar muy baja (no utilizable) del panel solar. Como no puede utilizar ninguna de las dos fuentes de energía que posee, notifica a la EC que va a quedarse sin energía.
8. TR08:

- Agentes: Batería y Panel Solar (Componente muy usada del mercado), Estación Central.
- Descripción: La TR reanuda sus funciones y comienza a almacenar nuevamente los datos que los sensores le envían, para luego hacerlos llegar a la estación central.

9. TR09:

- Agentes: Clock interno
- Descripción: El clock interno de la TR notifica que ha pasado un minuto desde el último envío de datos a la Estación Central. La TR recolecta los últimos datos obtenidos de los sensores para ser enviados.

10. TR10:

- Agentes: Sistema Interno
- Descripción: Los datos son comprimidos y encriptados para luego poder ser enviados a la EC según indicación de un protocolo pertinente.

11. TR11:

- Agentes: Clock interno
- Descripción: Dado que en la red GSM hay riesgo de pérdidas y retrasos, la TR tiene configurado un tiempo de espera para recibir la confirmación de que un mensaje enviado fue recibido. Al haber transcurrido este tiempo y no haber recibido esta confirmación, se da el mensaje por perdido y se lo envía nuevamente.

12. TR12:

- Agentes: EC
- Descripción: Se recibe la confirmación de recepción correcta de datos por parte de la EC. Se borran estos datos de la TR (porque ya están en la EC).

2.3.3. Casos de uso dentro de la EC:

1. EC1:

- Agentes: Sistema Interno
- Descripción: En esta funcionalidad la estación central determina un punto de restauración, donde todos los datos hasta dicho punto son correctos.

2. EC2:

- Agentes: Sistema Interno
- Descripción: Con un punto de restauración bien definido y con todos los datos necesarios el sistema procesa los modelos pertinentes.

3. EC3:

- Agentes: Sistema Interno
- Descripción: Una vez que se terminaron de calcular los modelos, y se guardan los resultados, se les notifica a los interesados.

4. EC4:

- Agentes: Sistema Interno
- Descripción: Una vez finalizados los cálculos, y obtenidos los resultados de los modelos, la estación central los guarda en la base de datos.

5. EC05:

- Agentes: Clock interno
- Descripción: Cuando se detecta la caída de una TR o de un sensor de una TR, la central trata de subsanar el problema. Si la caída fue de una TR, entonces trata de triangularla utilizando la información de las TRs vecinas. De no ser posible, se utiliza el servicio de Biggest Satelite para obtener los datos de la zona donde se encuentra la TR caída, guardando el tiempo de uso del sistema. Si la caída fue de un sensor dentro de una TR, entonces, se trata de triangular el sensor, con la información de los sensores del mismo tipo, de las TRs vecinas.

6. EC6:

- Agentes: TR / Biggest Satellite

- Descripción: La estación central detecta la caída de una TR cuando pasado el minuto del ultimo dato recibido, no se reciben mas datos. Así también de la misma forma, cuando se detecta la llegada de datos de una TR, la central detecta que la misma se reactivó. Desactiva una triangulación de esa TR si la hubiere o suspende los servicios de Biggest Sattelite para reemplazar esa TR si estuvieran activos. Se toman normalmente los datos provenientes de esa TR.

7. EC07:

- Agentes: Sistema de Monitoreo
- Descripción: Una vez que se realizó la triangulación o se comenzó a utilizar el servicio de Biggest Satellite, se anuncia al sistema de monitoreo.

8. EC08:

- Agentes: Sistema de monitoreo
- Descripción: Cuando se detecte la reactivación de un sensor, se anuncia al sistema de monitoreo.

9. EC09:

- Agentes: Sistema de monitoreo
- Descripción: Se detecta congestión en la red, cuando los datos arriban con retraso a la estación central, en este caso se informa al sistema de monitoreo para que este avise a quien corresponda.

10. EC10:

- Agentes: Sistema interno, operativo
- Descripción: Dependiendo de la configuración establecida para detectar outliers en los datos que se reciben, el sistema central, al detectar uno, se encarga de notificar dicho suceso a los responsables, para que lo examinen, y determinen si es un dato correcto o no.

11. EC11:

- Agentes: Experto en meteorología
- Descripción: Luego de la notificación de outliers, el encargado correspondiente de analizarlos, le indica al sistema que se descarten o se incluyan para procesar los modelos.

12. EC12:

- Agentes: TR
- Descripción: Ante el arribo de mensajes repetidos o en desorden, la EC se encarga de ordenarlos y descartar los repetidos.

13. EC13:

- Agentes: TR
- Descripción: Al recibir un mensaje de una TR, se le envía un mensaje de confirmación para que esta sepa que el mensaje llego correctamente.

14. EC14:

- Agentes: Sistema interno
- Descripción: Cada vez que se termina de utilizar el servicio de Biggest Satellite, la estación central, guarda el tiempo utilizado, para que luego se realice la liquidación.

15. EC15:

- Agentes: Experto en modelos matemáticos, Sistema de monitoreo
- Descripción: El experto configura los intervalos de confianza para outliers que él considera adecuados. El sistema guarda esta configuración.

16. EC16:

- Agentes: Sistema de monitoreo.
- Descripción: El sistema de monitoreo le envía a la EC los datos provenientes del sistema eólico para que esta los pueda usar según corresponda.

17. EC17:

- Agentes: Experto en modelos matemáticos, Sistema de monitoreo

- Descripción: Se ingresan o se eliminan reglas al modelo matemático, agregando de esta forma la posibilidad de realizar todo tipo de cálculos.

18. EC18:

- Agentes: Sistema interno
- Descripción: Se reciben los mensajes de las distintas TRs y pasan por las capas de transporte correspondientes para descomprimir y descryptar.

19. EC19:

- Agentes: Biggest Satellite
- Descripción: El sistema BS le envía a la EC los datos pedidos luego del reemplazo de una o varias TRS o sensores.

20. EC20:

- Agentes: Cliente Externo, experto en modelos matemáticos
- Descripción: Mediante un mecanismo fijado se agregan nuevos modelos matemáticos para ser procesados luego para un cliente externo.

2.3.4. Casos de uso dentro del sistema de Monitoreo

1. SM01:

- Agentes: Clock interno (tarea programada)
- Descripción: El último día hábil del mes el sistema envía al sistema de facturación, la cantidad de tiempo de uso del servicio de Biggest Satellite.

2. SM02:

- Agentes: Ec, Sistema de mail
- Descripción: Cuando la estación central detecta la caída de una TR, el sistema de monitoreo es notificado para que envíe un mail a los responsables avisando que una TR no está respondiendo.

3. SM03:

- Agentes: EC, Sistema de mail
- Descripción: Cuando la estación central detecta que la red se encuentra saturada, esta le notifica al sistema de monitoreo, para que envíe un mail a los responsables avisando de la eventualidad.

4. SM04:

- Agentes: Ministerio de Infraestructura, Base de datos
- Descripción: Mediante el servicio de Web Service, se le envían los datos que el sistema del Ministerio de Infraestructura pide para mostrar en su página.

5. SM05:

- Agentes: Cliente Externo, Base de Datos
- Descripción: Mediante el sistema de Web Service, se envían los datos solicitados al cliente externo, para que los utilice.

6. SM06:

- Agentes: Sistema eólico, Base de datos, EC
- Descripción: Se recibe la información del sistema eólico, y se almacena en la base de datos, para que luego la estación central los procese. Se notifica a la EC para que luego esta compare los datos con los obtenidos de las TRs.

7. SM07:

- Agentes: Base de Datos, EC, TR, operario
- Descripción: Desde el sistema de monitoreo y a través de la Estación Central, un operario puede modificar la agenda de un sensor que elija dentro de una TR. Cuando se realiza el cambio, esto es reflejado en el Servidor de Base de datos, para que quede registrado el cambio.

8. SM08:

- Agentes: Operario, Base de Datos
- Descripción: Un operario puede configurar el envío de alertas para cuando la Estación Central detecta la caída de una TR o la saturación de la red, agregando o quitando responsables en la lista de mails de notificación. Este cambio se refleja en el Servidor de Base de datos.

9. SM09:

- Agentes: Operario, Base de Datos
- Descripción: Un operario puede ingresar o eliminar a un cliente externo, el cual podra recibir o no la información que el sistema procese. Esto se refleja en el Servidor de Base de datos.

10. SM10:

- Agentes: Base de Datos, operario, EC
- Descripción: Un operario puede ingresar o quitar una TR, con toda su información relacionada (ubicación, identificación, etc.).Este cambio se refleja en el Servidor de Base de datos y en la Estación Central, la cual comenzará o no a recibir y procesar los datos que lleguen o no de dicha TR.

11. SM11:

- Agentes: Base de Datos, operario, EC
- Descripción: Un operario puede agregar o quitar un nuevo tipo de sensor. Dicho tipo es reflejado en el Servidor de Base de datos y en la Estación Central.

12. SM12:

- Agentes: Operario, Base de Datos, Sistema de Pantallas
- Descripción: Se muestra en pantallas según sea pedido las distintas características que el usuario desea ver.
 - Vista global del estado sistema
 - Vista por segmento de red
 - Vista por TR o sensor

2.4. Escenarios

En esta sección se mostrarán y detallarán los diferentes escenarios para los atributos de calidad identificados en el sistema a desarrollar. Estos, consideramos que son los principales, y que definirán la arquitectura. Mas adelante se mostrará el diagrama de arquitectura pensada para este sistema y las tácticas usadas para cubrir los principales atributos de calidad encontrados.

Los atributos de calidad se pudieron identificar gracias al QAW presentado por los stakeholders, en el cual se discutieron las necesidades de cada uno de ellos y se pudo llegar a un común acuerdo de los 5 atributos más relevantes en el sistema. Sin embargo en análisis del entorno y enunciado nos permitieron identificar algunos atributos adicionales que consideramos importantes para desarrollar un producto completo y consistente.

Se identificaron 7 atributos de calidad en total. Estos son: performace, disponibilidad, confiabilidad, modificabilidad, seguridad, interoperabilidad, usabilidad. Cada uno tiene 1 o mas escenarios en los cuales cada atributo aplica.

■ Atributos de Performace:

1. La información crítica de los resultados del modelo matemático se envía lo mas rápido posible a la pagina del ministerio.

Fuente: Interna (alarma).

Estimulo: El modelo matemático genera una predicción de uno o más fenómenos meteorológicos de importancia crítica.

Artefacto: Procesamiento (Estación central), Datos(sistema de monitoreo).

Entorno: Normal/Sobrecarga.

Respuesta: Se envia la información crítica al ministerio.

Medición de la respuesta: La información relacionada con la predicción se envia en a lo sumo 30 segundos.

2. El procesamiento de los datos obtenidos para generar una predicción se realiza lo mas rápido posible.

Fuente: Interna (alarma).

Estimulo: Pedido de procesamiento de los datos almacenados.

Artefacto: Procesamiento (Estación central).

Entorno: Sobrecarga.

Respuesta: Los datos son procesados y se devuelve un resultado.

Medición de la respuesta: El resultado se procesa en menos de 1 minuto.

■ Atributos de disponibilidad:

1. Los mensajes que se reciben en forma correcta son procesados.

Fuente: TR, Biggest Satellite, Sistema eólico.

Estimulo: Se recibe un mensaje con datos capturados.

Artefacto: Data manager (Estación central).

Entorno: Normal.

Respuesta: El mensaje es procesado.

Medición de la respuesta: El mensaje es procesado y almacenado en el 99,99 % de las veces.

2. La información del sistema está disponible para los clientes externos o usuarios.

Fuente: Cliente externo, usuario.

Estimulo: Pedido de datos almacenados.

Artefacto: Datos (Sistema de monitoreo).

Entorno: Normal.

Respuesta: Los datos son enviados a los que lo solicitaron.

Medición de la respuesta: Los datos se encuentran disponibles en el 99,999 % de los casos.

■ Atributos de confiabilidad:

1. Los resultados obtenidos son correctos.

Fuente: Componente interna.

Estimulo: Se realiza un pedido de la información meteorológica.

Artefacto: Procesamiento (Estación central).

Entorno: Normal.

Respuesta: Los datos almacenados son procesados y se devuelve un resultado(pronostico).

Medición de la respuesta: El pronóstico se confirma en un 99,9 % de los casos.

- Los resultados de los modelos son consistentes.

Fuente: Componente de procesamiento.

Estimulo: Resultado de modelo inválido.

Artefacto: Sistema corroborador de resultados(Estación central).

Entorno: Normal.

Respuesta: Se detectan los errores y no se modifica el estado actual consistente.

Medición de la respuesta: El sistema tiene resultados consistentes en el 99,999 % de los casos.

- Los mensajes desordenados son ordenados correctamente.

Fuente: Terminal Remota.

Estimulo: Llegan datos desordenados.

Artefacto: Recepcion segura (Estación central).

Entorno: Normal.

Respuesta: Los mensajes son ordenados correctamente.

Medición de la respuesta: Los mensajes desordenados son ordenados correctamente en el 99,999 % de los casos.

- Atributos de seguridad:

1. Los mensajes que llegan a la Estación central corruptos o erroneos son descartados.

Fuente: Intruso, usuario, TR.

Estimulo: Arriba un mensaje alterado, corrupto o erroneo.

Artefacto: Recepción segura (Estación central).

Entorno: Online.

Respuesta: Se detecta el mensaje alterado, corrupto o erroneo y se descarta.

Medición de la respuesta: Los mensajes alterados, corruptos o erroneos, son descartados en el 99,99 % de los casos.

2. La información recibida por el sistema en general desde sistemas externos es confiable.

Fuente: Intruso.

Estimulo: Arriba un mensaje alterado, corrupto o erroneo.

Artefacto: Interfaz SE (Sistema de monitoreo), BS Interfaz(Estación central).

Entorno: Online.

Respuesta: Se detecta el mensaje alterado, corrupto o erroneo y se descarta.

Medición de la respuesta: Los mensajes alterados, corruptos o erroneos, son descartados en el 99,99 % de los casos.

- Atributos de modificabilidad:

1. El modelo se puede modificar en un lapso de tiempo aceptable.

Fuente: Usuario, experto, desarrollador.

Estimulo: Se quiere modificar el modelo matemático.

Artefacto: Procesamiento (Estación central).

Entorno: Ejecución

Respuesta: El modelo matemático es modificado.

Medición de la respuesta: La modificación se realiza en a lo sumo 72 horas.

2. Se pueden instalar TRs o sensores en menos de 10 días.

Fuente: Empresa.

Estimulo: Instalación de nuevas TRs o sensores.

Artefacto: Controlador Estado (Estación central).

Entorno: Normal.

Respuesta: Se instalan las componentes solicitadas.

Medición de la respuesta: La instalación se realiza en menos de 10 días.

3. La agenda de una TR se configura rá

Fuente: Usuario u Operario.

Estimulo: Cambio de la agenda de un sensor.

Artefacto: Configuración agenda (Terminal Remota).

Entorno: Ejecución.

Respuesta: La agenda se configura adecuadamente.

Medición de la respuesta: La configuración se realiza en menos de 60 segundos.

■ Atributos de usabilidad:

1. El acceso a la información es satisfactorio para el usuario.

Fuente: Usuario final / empresa.

Estímulo: Acceso a la información mediante Google Maps, estado del tiempo, etc.

Artefacto: Servicios página (Sistema de monitoreo).

Entorno: Ejecución.

Respuesta: Se brinda el servicio solicitado.

Medición de la respuesta: El usuario obtiene un 90 % de satisfacción.

2. El acceso a la información del sistema se hace fácilmente.

Fuente: Empleado.

Estímulo: Acceso a la información de estado fácilmente.

Artefacto: Visualización (Sistema de monitoreo).

Entorno: Ejecución.

Respuesta: La información es brindada correctamente.

Medición de la respuesta: El usuario obtiene un 90 % de satisfacción.

■ Atributos de interoperabilidad:

1. El acceso a la información del sistema se hace fácilmente.

Fuente: Sistema eólico.

Estímulo: Llegan datos al sistema.

Artefacto: Arbitro 1 (Estación central).

Entorno: Normal.

Respuesta: Se procesan los datos y se seleccionan los mas precisos.

Medición de la respuesta: Los datos se reciben en el 99,99 % de los casos.

Parte 3

Plan del Proyecto

3.1. Estimaciones a los casos de uso del proyecto(Asignación de pesos)

El método que se utilizó para estimar las tareas fue Wideband Delphi. A continuación se detallarán las principales etapas del proceso, sus complicaciones, resultados en común y las conclusiones finales.

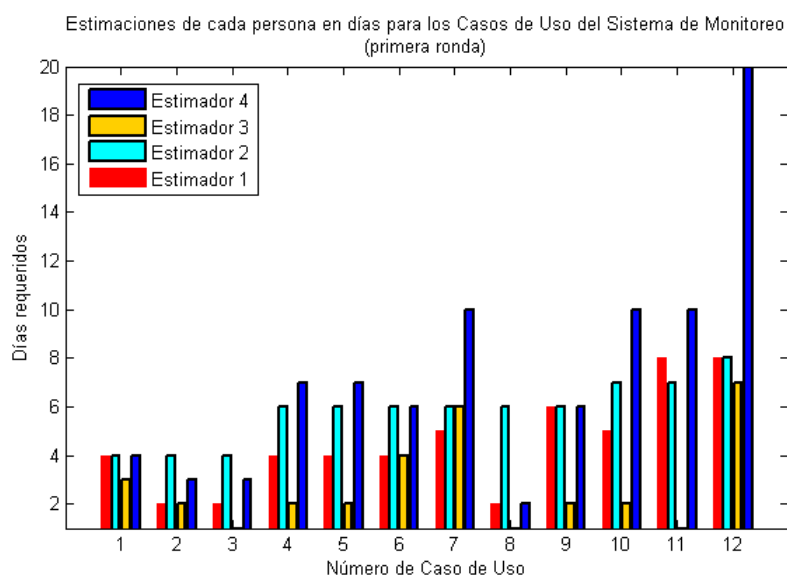
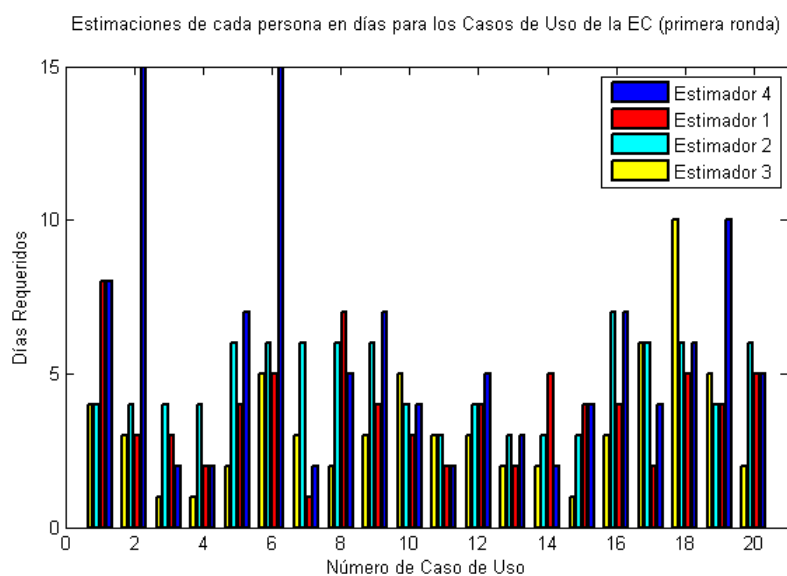
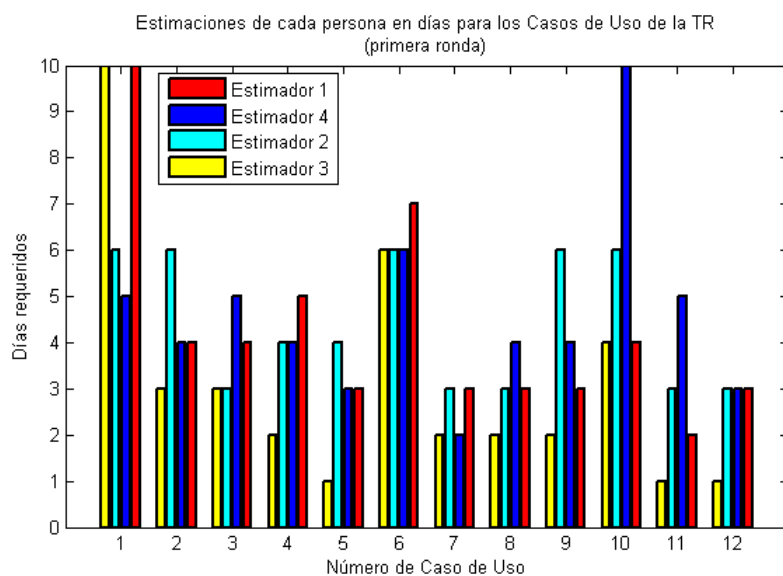
3.1.1. Planificación y reunión de lanzamiento

Estas dos etapas del proceso fueron básicamente, para poner en común que se iban a estimar los casos de uso de la sección anterior, el método utilizado, y la medida que se iba a utilizar para la estimación, en nuestro caso llegamos a que la medida para estimar viene dada por horas/persona. Esta decisión fue tomada más que nada para simplificar la estimación y no tener que trabajar en horas en esta parte del proyecto. El objetivo básico de éstas estimaciones fue para poder luego determinar cantidad de iteraciones, duración y fase a la que pertenecen de manera más o menos objetiva. Los encargados de llevar a cabo las estimaciones fuimos los 4 integrantes del grupo.

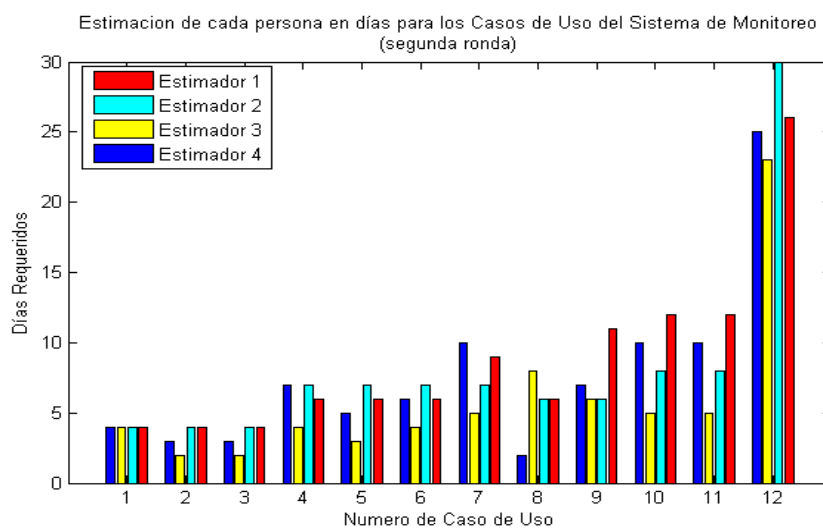
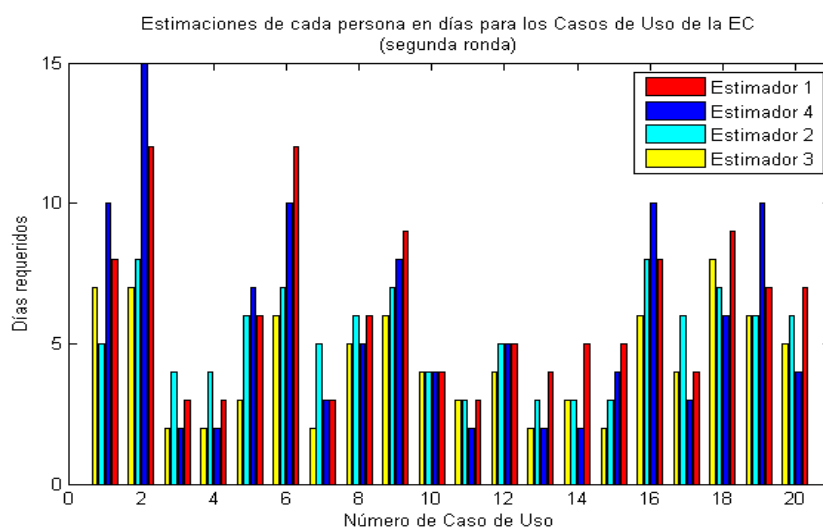
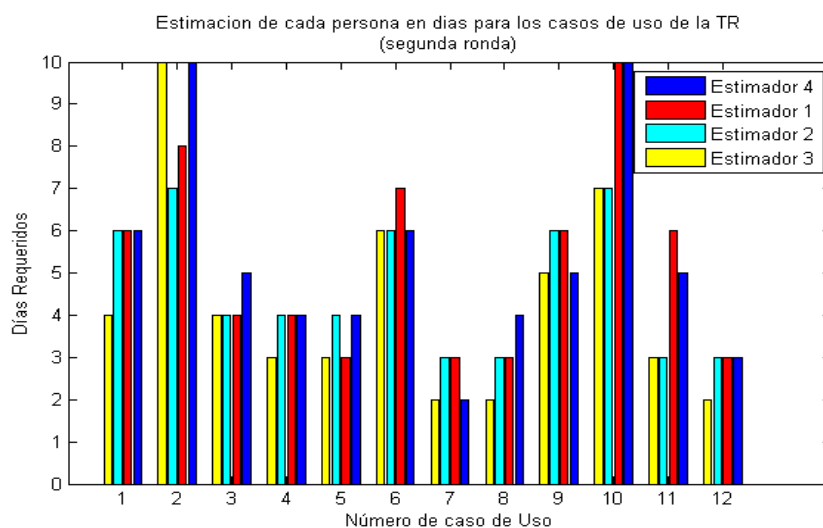
3.1.2. Preparación individual y reunión de estimación

En estas dos etapas del proceso llevamos a cabo cada una de las estimaciones que consideramos necesarias para cada uno de los casos de uso. Luego analizamos diferencias realizando gráficos que reflejaran la información de las estimaciones como se detalla a continuación.

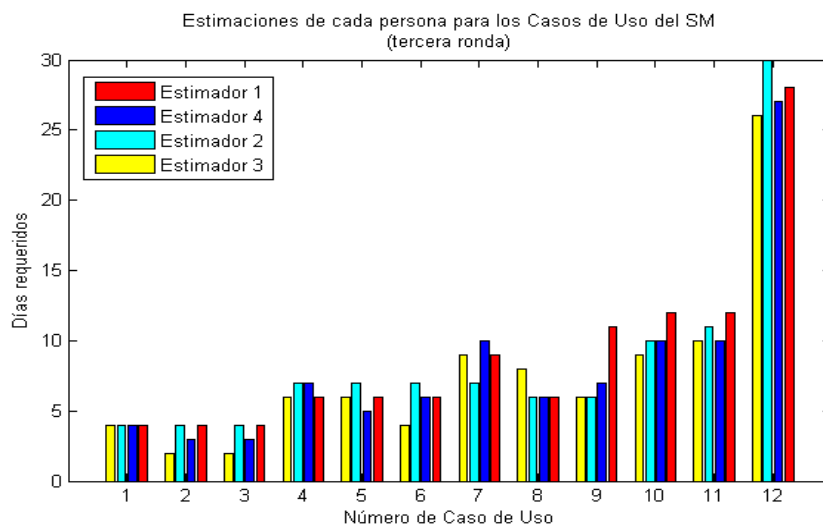
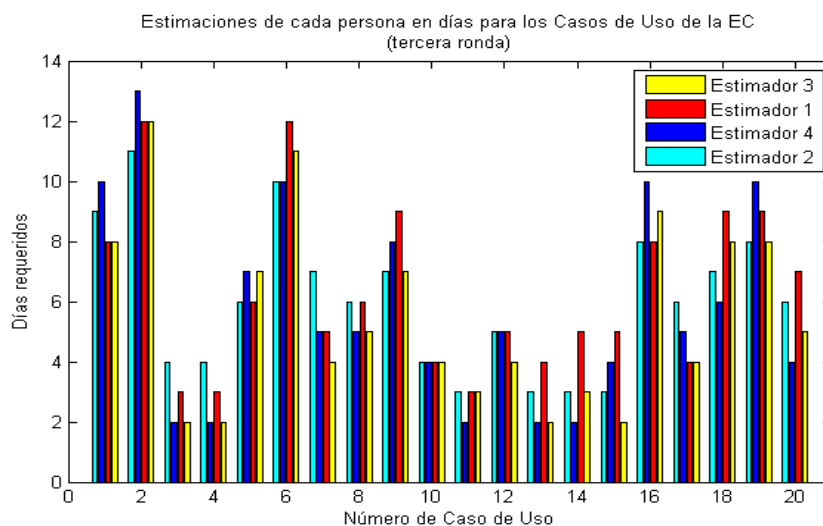
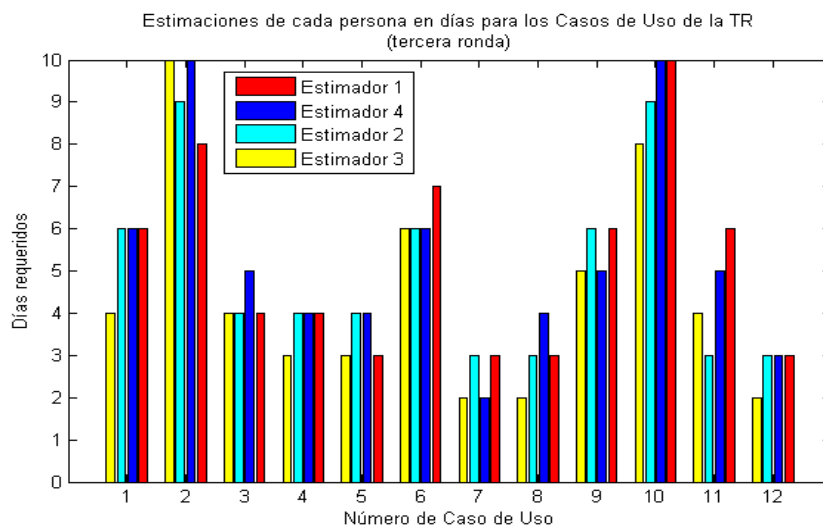
La primera ronda de estimaciones generó los siguientes resultados:



Había muchas tareas que diferían bastante en la estimación por lo cual en la primera reunión de estimación se intentó nuevamente ver el alcance de cada caso de uso, lo cual sirvió para que algunos estimadores pudieran tener en cuenta tareas y funcionalidades que en principio no habían considerado. Luego se procedió a realizar una segunda ronda de estimaciones y los resultados se detallan a continuación en los siguientes gráficos:



Si bien hubo mejorías en la segunda ronda, consideramos que existían desvíos no aceptables, por lo cual nuevamente se revisaron los casos de uso y se llegó a la conclusión que una tercera ronda de estimaciones reduciría éstos desvíos. Luego de la tercera ronda consideramos detener las estimaciones ya que el desvío fue aceptable (alrededor de dos días)y las estimaciones, calculando un promedio entre ellas podían llegar a reflejar valores adecuados. Los resultados son los siguientes:



3.1.3. Ensamblado de tareas y revisión de resultados

Finalmente en ésta sección se calculó un promedio de las estimaciones realizadas para cada tarea por cada estimador y se pudo concluir en resultados que a todos los integrantes le parecieron adecuados. A continuación se detallan los pesos asignados a cada caso de uso que resultaron del proceso de estimación.

■ Casos de uso de la TR.

Número CU	Duración en hs. (días)
CU 1	120 (5)
CU 2	216 (9)
CU 3	96 (4)
CU 4	72 (3)
CU 5	72 (3)
CU 6	144 (6)
CU 7	48 (2)
CU 8	72 (3)
CU 9	120 (5)
CU 10	216 (9)
CU 11	96 (4)
CU 12	48 (2)

■ Casos de uso de la EC.

Número CU	Duración en hs.
CU 1	192 (8)
CU 2	288 (12)
CU 3	48 (2)
CU 4	48 (2)
CU 5	144 (6)
CU 6	240 (10)
CU 7	120 (5)
CU 8	120 (5)
CU 9	168 (7)
CU 10	96 (4)
CU 11	48 (2)
CU 12	96 (4)
CU 13	48 (2)
CU 14	72 (3)
CU 15	72 (3)
CU 16	192 (8)
CU 17	96 (4)
CU 18	168 (7)
CU 19	192 (8)
CU 20	120 (5)

■ Casos de uso del SM.

Número CU	Duración en hs.
CU 1	96 (4)
CU 2	72 (3)
CU 3	72 (3)
CU 4	144 (6)
CU 5	144 (6)
CU 6	120 (5)
CU 7	192 (8)
CU 8	144 (6)
CU 9	168 (7)
CU 10	240 (10)
CU 11	240 (10)
CU 12	648 (27)

3.1.4. Armado de planificación de iteraciones

A partir de los resultados anteriormente obtenidos se realizó una estimación global del proyecto, conjuntamente con la división en iteraciones y la asignación de los casos de uso a las distintas iteraciones

- Inception 01 - Resuelta por los docentes de ISII
- Elaboration 01 - Iteración actual - Casos de uso asociados a la primera iteración, documento con las secciones correspondientes detalladas en los entregables de la primera iteración (Riesgos, Arquitectura, Atributos de Calidad, Prueba de concepto, etc)
- Elaboration 02 - Refinamiento final de casos de uso y definición de algunos protocolos críticos
- Construction 01 - TR10, TR11, TR12, EC12, EC13, EC18 (lo que no se llegó a hacer en Elab. 1 y 2)
- Construction 02 - TR01, TR02, TR04, TR05, TR06
- Construction 03 - TR03, TR07, TR08, TR09, EC01, EC10
- Construction 04 - EC05, EC06, EC07, EC08
- Construction 05 - EC02, EC03, EC04, EC11, EC15, EC17
- Construction 06 - EC09, EC14, EC20, SM01, SM02, SM03
- Construction 07 - EC16, EC17, SM06, SM08
- Construction 08 - SM04, SM05, SM07, SM09
- Construction 09 - SM10, SM11
- Construction 10 - SM12
- Transition 01 - Puesta en producción y test integral
- Transition 02 - Puesta en producción y test integral
- Transition 03 - Puesta en producción y test integral

Se debe destacar que el orden elegido para realizar los casos de uso se debió a partes críticas del sistema y se priorizó entonces en este orden: protocolo, sensores, datos, caídas, procesos matemáticos, energía de TR, y luego el resto de las tareas.

Es también pertinente que en cada etapa de construcción se realicen los test de integración posibles con los casos de usos ya realizados en las iteraciones anteriores.

El tiempo de las iteraciones de construcción es de un mes mientras que las de elaboración son de 5 semanas y las de transición de 3 lo que da un costo total del proyecto en tiempo de aproximadamente 60 semanas.

3.2. Planificación de la primera iteración

En esta parte del informe se mostrarán todos los procedimientos llevados a cabo para obtener una completa y consistente planificación de la primera iteración. Finalizadas dichas etapas se presentará el diagrama Gantt resultante de la recopilación de todos los datos de la planificación.

3.2.1. División en tareas de los casos de uso para la primera iteración

En esta sección consideraremos las tareas asociadas a los casos de uso que fueron determinados para la primera iteración por la cátedra. Cabe destacar que sólo se van a considerar un subconjunto de las tareas para cada caso de uso, ya que constituyen una prueba de concepto y la funcionalidad completa se podrá completar en el resto de las iteraciones.

Casos de uso asociados a la terminal remota

1. Sincronizando datos de censados. (TR09)

- Investigación
 - Posibles formas de completar datos según métodos numéricos.
 - Librerías de alarmas y precisión de las mismas (deben tener en cuenta el tiempo REAL que paso y no solo el que tardo la aplicación en ejecutar).
- Implementación
 - Verificador de datos completos.
 - Completador de datos según datos históricos.
 - Armador de paquete de datos.
- Testing
 - Armado de datos falsos con posibles datos faltantes.
 - Testing de unidad.
- Otros
 - Armado de sandboxes para realizar implementación.
 - Generación de datos falsos.

2. Encriptando/comprimiendo datos y enviado según protocolo. (TR10)

- Investigación
 - Protocolo seguro y confiable para la TR con la EC.
 - Modelos de encriptación/compresión de datos (NO para esta iteración)
- Implementación
 - Interfaz de capa de encriptación/compresión de datos.
 - Capa de envío de datos segura y confiable.
- Testing
- Otros
 - Definir forma de generar datos falsos que luego puedan ser levantados por la EC.

3. Repitiendo mensaje si no llega confirmación. (TR11)

- Investigación
 - Formas de protocolos que permiten reenvío de mensajes.
 - Definir el tiempo esperado para el ACK.
- Implementación
 - Obtención de dato anteriormente enviado sin ACK recibido.
 - Protocolo de no reenvío dato si llega algún ACK de todos los enviados.
- Testing
- Otros

- Generación de ACK falsos con probabilidad de arribo.
- 4. Borrando mensaje al llegar confirmación. (TR12)
 - Implementación
 - Buffer de datos históricos ya enviados.
 - Verificador de existencia de datos y sacarlo.
 - Testing
 - Generación de datos falsos los cual serán los borrados luego.

Casos de uso asociados a la Estación central

1. Procesando modelos matemáticos. (EC02)
 - Investigación
 - Modelos matemáticos más simples para primera etapa.
 - Implementación
 - Modelos Matemáticos simples.
 - Simulador de pedido de procesamiento de datos (Ya que no se hace CU1).
 - Testing
 - Otros
 - Armado del entorno que permita procesamiento de modelo.
2. Detectando caída/alta de TR. (EC05)
 - Investigación
 - Motivos de caída de una TR y formas de detección.
 - Implementación
 - Capa de detección de caída por los casos antes mencionados.
 - Verificador real de que la TR de que la esta caída.
 - Regenerador de TR original al detectar alta.
 - Testing
 - Armado de escenario falsos simulando caída de TR.
3. Ordenando/descartando mensajes repetidos. (EC12)
 - Investigación
 - Protocolos de descarte de SMS y formas de darse cuenta de dicho caso.
 - Implementación
 - Almacenamiento de mensajes históricos para verificar si es repetido.
 - Etiquetador único para mensaje.
 - Testing
 - Armado de ambiente para simular mensajes repetidos
4. Confirmando llegada de mensaje. (EC13)
 - Investigación
 - Formato de ACK e información extra que pueda ser útil enviar (como un identificador único de error).
 - Implementación
 - Recibidor de mensaje.
 - Verificador de mensaje correcto.
 - Capa (simple) de descriptación / descompresión.
 - Envío de mensaje con ACK.
 - Testing
 - Generación de mensajes falsos para poder simular recibirlos y confirmar.
5. Recibiendo, descomprimiendo y descriptando mensaje de TR. (EC18)

- Investigación
 - Formato de datos a recibir.
- Implementación
 - Capa de recibo de mensajes (ver CU anterior).
- Testing
 - Generar datos falsos para poder recibir.

3.2.2. Estimación, duración de tareas asociadas a la primera iteración

En principio, lo que se hizo en ésta etapa es, como resultado de la división en subtareas para llevar a cabo la prueba de concepto, reducir considerablemente los tiempos estimados de dichos casos de uso, ya que la funcionalidad completa no era necesaria a esta altura del proyecto, sino que las funcionalidades a implementar se corresponden con un conjunto minimal de dichos casos de uso. Entonces luego de una evaluación y puesta en común de todos los integrantes del grupo se llegó a la conclusión que el tiempo que se necesita para implementar ese conjunto de funcionalidades para cada caso de uso es el siguiente:

Número CU	Duración(En días/persona de 6 horas)
TR09	2
TR10	3
TR11	2
TR12	2
EC02	2
EC05	3
EC12	2
EC13	1
EC18	3

Estos tiempos corresponden a la implementación de todas las subtareas mencionadas en la etapa anterior para los casos de uso abarcados en ésta iteración. También se consideraron en las estimaciones presentadas en la tabla anterior los tiempos asociados a la instalación de entornos de desarrollo para dichos casos de uso.

3.2.3. Generación de dependencias

Luego lo que se hizo a partir de éstos casos de uso fue analizar dependencias entre sí. Se tuvieron en cuenta dependencias en cuanto a implementaciones necesarias para llevar a cabo otras, es decir del estilo: No voy a poder implementar este caso de uso a menos que tenga éstos otros dos implementados. Sin embargo como se verá mas adelante en el Gantt realizado nos surgieron dependencias por falta de recursos.

3.2.4. Definición y asignación de recursos

Los 4 recursos que se utilizarán son los cuatro integrantes del grupo cumpliendo cualquiera de los posibles roles como ser programador, analista de requerimientos, testers, etc. En general hubo muchas tareas en las cuales decidimos participar íntegramente todos y otras en las cuales hacerlas en paralelo dividiendo recursos nos era más ventajoso ya que los tiempos se acortaban considerablemente.

3.2.5. Estimación y precedencias en el resto de las tareas del proyecto y generación de diagrama Gantt

En esta sección vamos a describir brevemente el armado del diagrama Gantt correspondiente a la planificación de la primera iteración. Además de las tareas asociadas a los casos de uso mencionadas anteriormente (para la prueba de concepto), en el proyecto se tuvieron en cuenta todas las tareas asociadas a la primera iteración tales como análisis de requerimientos, análisis de riesgo, definición de arquitectura e incluso las tareas que conllevaron a la planificación y el armado del Gantt propiamente dicho.

Muchas de éstas tareas se estimaron de manera sencilla, o bien porque ya las habíamos hecho antes de diseñar el Gantt o bien porque en trabajos prácticos de otras materias habíamos realizado tareas similares. Sin embargo tuvimos que hacer estimaciones sin muchos argumentos para aquellas tareas en las que en principio no conocemos un tiempo para realizarlas, tales como el armado de la arquitectura. De todos modos intentamos ser lo más realistas posibles en las estimaciones y dentro de lo posible cumplir con el cronograma y las fechas de entrega de cada parte del proyecto.

En el Gantt que se verá en la siguiente sección, se podrán observar tareas generales, precedencias por recursos o funcionalidad, tareas críticas e incluso tres hitos que propusimos: fin de la preentrega, fin de la arquitectura y fin de la primera iteración. Cabe destacar que las tareas asociadas a generación de documentos de preentrega o entrega de primera iteración corresponden a armar finalmente dicho documento para entregarlo. Sin embargo no corresponde a realizarlo en su totalidad, sino que el documento se realizará en paralelo a medida que se avanza en el proyecto y lo que ésta tarea refleja es el armado, juntada de partes y revisiones finales.

Es importante destacar que el calendario que utilizamos son los días de la semana incluyendo los sábados y domingo ya que en realidad utilizaremos estos días, por lo cual aproximamos que desarrollaremos tareas relacionadas con el proyecto alrededor de 6 horas por cada uno de los días mencionados anteriormente.

3.2.6. Diagrama de Gantt

A continuación se presentará el diagrama de Gantt finalmente concluido.

	Nombre de tarea	Duración	Comienzo	Fin	Predec
1	Análisis de Entorno y Financiado	3 días	mar 08/09/09	vie 11/09/09	
2	Entendimiento enunciado	1 día	mar 08/09/09	mié 09/09/09	
3	Análisis principales requerimientos	1 día	mié 09/09/09	jue 10/09/09 2	
4	Elaboración modelos conceptuales	1 día	jue 10/09/09	vie 11/09/09 3	
5	Consultas Stakeholders (Docentes)	1 día	jue 10/09/09	vie 11/09/09 3	
6	Investigación/Comprensión métodos estimación, planificación	2 días	sáb 12/09/09	lun 14/09/09 1	
7	Planificación del Proyecto	5 días	lun 14/09/09	lun 21/09/09 3,6	
8	Generación casos de uso	2 días	lun 14/09/09	jue 17/09/09	
9	Asignación de pesos a CU	2 días	jue 17/09/09	sáb 19/09/09 8	
10	Describir tipo/cantidad/duración iteraciones	1 día	dom 20/09/09	lun 21/09/09 9	
11	Asignación CU a iteraciones	1 día	dom 20/09/09	lun 21/09/09 9	
12	Planificación de la primera iteración	4 días	lun 21/09/09	sáb 26/09/09 10,11	
13	División en tareas de CU para primera iteración	6 horas	lun 21/09/09	mar 22/09/09	
14	Estimación/Duración tareas asociadas a 1 iteración	1 día	mar 22/09/09	mié 23/09/09 13	
15	Generación de dependencias	6 horas	mié 23/09/09	jue 24/09/09 14	
16	Definición de Recursos	6 horas	mié 23/09/09	jue 24/09/09 14	
17	Asignación de Recursos	6 horas	jue 24/09/09	vie 25/09/09 15,16	
18	Generación Diagrama Gantt	6 horas	vie 25/09/09	sáb 26/09/09 17	
19	Generar Documento Preeentrega	1 día	sáb 26/09/09	dom 27/09/09 12,7,1	
20	Fin Preeentrega	0 días	dom 27/09/09	dom 27/09/09 19	
21	Análisis Avance y correcciones preeentrega	1 día	lun 28/09/09	mar 29/09/09 20	
22	General OAW	3 días	mar 29/09/09	sáb 03/10/09 21	
23	Definir atributos de calidad	1 día	mar 29/09/09	mié 30/09/09	
24	Definir Escenarios	4 horas	mié 30/09/09	jue 01/10/09 23	
25	Usar sistema SEI	1 día	jue 01/10/09	vie 02/10/09 24	
26	Definir Documentación	4 horas	vie 02/10/09	sáb 03/10/09 25	
27	Análisis de Retorno	3 días	mar 29/09/09	sáb 03/10/09 26	

	Nombre de tarea	Duración	Comienzo	Fin	Predec	
28	Investigar riesgos posibles	1 día	mar 29/09/09	mié 30/09/09		
29	Detectar riesgos más peligrosos	4 horas	mié 30/09/09	jue 01/10/09	28	
30	Generar planes de contingencia	4 horas	jue 01/10/09	jue 01/10/09	29	
31	Generar lista de riesgos principales de primera iteraci	4 horas	vie 02/10/09	vie 02/10/09	30	
32	Definir Documentación	4 horas	vie 02/10/09	sáb 03/10/09	31	
33	Arquitectura	4 días	sáb 03/10/09	jue 08/10/09	27,22	
34	Estudiar modelos arquitectonicos	1 día	sáb 03/10/09	dom 04/10/09		
35	Analizar protocolo sensores	4 horas	dom 04/10/09	lun 05/10/09	34	
36	Analizar protocolo TR-EC seguro y contable	4 horas	dom 04/10/09	lun 05/10/09	34	
37	Analizar protocolo Biggest Satelite	4 horas	lun 05/10/09	lun 05/10/09	35,36	
38	Generar interfaces TR-EC TR-Sensor	1 día	mar 06/10/09	mié 07/10/09	37	
39	Generar arquitectura general	4 horas	mié 07/10/09	mié 07/10/09	38	
40	Definir documentación Arquitectura	4 horas	jue 08/10/09	jue 08/10/09	39	
41	Fin Arquitectura	0 días	jue 08/10/09	jue 08/10/09	33	

sep '09

L | M | X | J | V | S | D | L | M | X | J | V | S | D | L | M | X | J | V | S | D | L | M | X | J | V

04 oct '09

11 oct '09

18 oct '09

25 oct '09

01 nov '09

Gonzalo,Martin

Gonzalo,Martin

Gonzalo,Martin

Gonzalo,Martin

Gonzalo,Martin

Gonzalo,Martin

Gonzalo,Martin,Sergio,Victoria

Martin,Sergio

Gonzalo,Victoria

Gonzalo,Martin,Sergio,Victoria

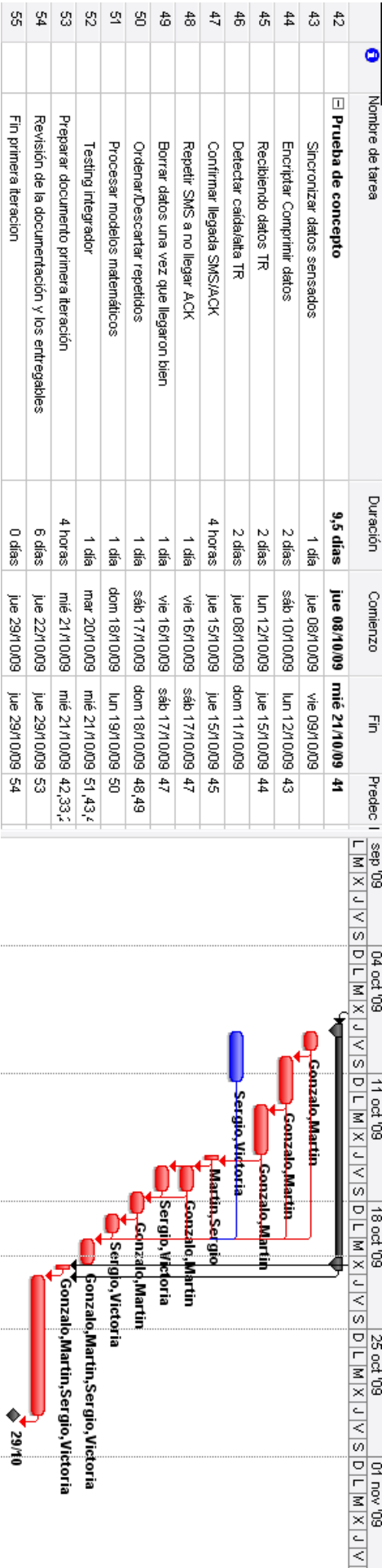
Gonzalo,Martin,Sergio,Victoria

Gonzalo,Martin,Sergio,Victoria

Gonzalo,Martin,Sergio,Victoria

Gonzalo,Martin,Sergio,Victoria

08/10



3.3. Evaluación, seguimiento y ajustes a la preentrega

Una vez finalizada la planificación detallada de la primera iteración y el plan general del proyecto respresentados en lo que fue la preentrega del proyecto y llevada a cabo una primera validación con stakeholder dada por la corrección de la preentrega por nuestro corrector, se pudieron sacar en claro algunas conclusiones.

- Estimaciones: Pese a que era nuestra primera experiencia en estimaciones de tiempo y esfuerzo la utilización del método Wideban Delphi no orientó bastante a la hora de estimar. Las consultas post-entrega con los docentes y otros grupos rectificaron que la duración total del proyecto entaba dentro de los límites pensados por la cátedra. Además se pudo seguir el cronograma detallado en el Gantt sin mayores complicaciones y finalmente entregar a tiempo la preentrega.
- Métricas de estimación: En esta sección tuvimos que corregir algunas cuestiones que hacía que el entendimiento de las estimaciones fuese complicado para nuestro corrector, es decir, si bien la información era consistente (el mismo corrector probó un par de equivalencias) la información no era del todo clara y las métricas utilizadas eran diferentes de acuerdo a la parte del informe que llevábamos a cabo. Por lo cual se ajustaron las métricas normalizándolas para un mejor entendimiento de las mismas.
- Gantt: El cronograma detallado en el Gantt había quedado en días de 8 horas para compensar la no utilización de sábados y domingos. Cambiamos ésto y preferimos utilizar los sábados y domingo y días de 6 horas ya que se adapta mucho mejor a lo que se hará en realidad. Además cada una de las funcionalidades que se muestran en el Gantt en la sección prueba de concepto se corresponden con la planificación, diseño, implementación, armado de entornos de desarrollo y testing. Se agregó además la semana adicional dada por la cátedra para completar el trabajo práctico.
- Eliminación de un caso de uso: En la primera iteración contemplabamos para la prueba de concepto el caso de uso asociado a la triangulación, sin embargo posteriormente se determinó que dicha funcionalidad no era necesaria, por lo cual la quitamos de dicha iteración.
- Otros: Se corrigió información mal redactada y otros detalles menores mencionados por el corrector.

Parte 4

Gestión de riesgos

En esta sección se discutirán los riesgos más relevantes en el contexto de nuestro problema. Para cada uno de ellos definiremos las características que los definen incluyendo la probabilidad de que ocurran, el impacto (negativo) si ocurren (en una escala del 1 al 10), la exposición al riesgo (medida como probabilidad * impacto). Además incluiremos en los casos que sea posible, formas de mitigarlo y también posibles planes de contingencia en caso de que ocurriesen.

4.1. Identificación de riesgos

Para llevar a cabo la identificación de riesgos tuvimos en cuenta en mayor o menor medida los siguientes métodos:

- Brainstorm.
- Cuestionario de identificación taxonómica.
- Lista de riesgos comunes.

Para determinar la probabilidad de impacto y exposición al riesgo, primero determinamos los riesgos y luego cada uno estimó por separado estos valores (para no influirnos entre nosotros) y en base a esto discutimos después y consensuamos los valores finales.

Finalmente los riesgos determinados fueron:

1. - R1 - Incorporar modelos matemáticos erróneos.

- Probabilidad: 0.2
- Impacto: 10
- Exposición al riesgo: 2.0
- Mitigación: Hacer pruebas con los modelos antes de ponerlos en funcionamiento. Probarlos con datos históricos reales y ver si acierta las predicciones. Estas pruebas deberían hacerse en conjunto con el meteorólogo que desarrollo el modelo para que pueda corregir eventuales errores.
- Plan de Contingencia: Volver al último modelo matemático válido.

En caso de: incorporar modelos matemáticos inexactos posiblemente se hagan predicciones incorrectas, lo cual es gravísimo en el contexto de nuestro sistema.

2. - R2 - Mala interpretación de requerimientos no esenciales.

- Probabilidad: 0.3
- Impacto: 6
- Exposición al riesgo: 1.8
- Mitigación: Mayor tiempo de validación con los stakeholders.

- Plan de Contingencia: Solucionar de alguna manera, en tiempo de ejecución los problemas generados.

En caso de: una mala interpretación de los requerimientos no esenciales (aquellos no necesariamente críticos) los usuarios no estarán conformes y el posiblemente haya que corregir dichos errores después de haber puesto el proyecto en producción.

3. - R3 - Infiltración en modelo eólico.

- Probabilidad: 0.1
- Impacto: 6
- Exposición al riesgo: 0.6
- Mitigación: –
- Plan de Contingencia: Utilizar los datos generados por nuestro sistema aunque no sean más precisos que los del sistema eólico.

En caso de: infiltración en el modelo eólico llegarán datos incorrectos lo cual producirá, en caso de ser considerados válidos, resultados y reportes incorrectos de nuestro sistema.

4. - R4 - Quiebra de empresas que previamente hayan contratado nuestros servicios.

- Probabilidad: 0.09
- Impacto: 7
- Exposición al riesgo: 0.63
- Mitigación: Cobrar por adelantado los costos de la instalación física.

En caso de: Quiebre de empresas que previamente nos hayan contratado y se hayan instalado las componentes solicitadas los gastos de instalación perjudicarían la economía de la empresa. Para evitar esto, se debería cobrar por adelantado a la empresa siempre que sea posible.

5. - R5 - Caída página de la infraestructura.

- Probabilidad: 0.4
- Impacto: 4
- Exposición al riesgo: 1.6
- Mitigación: Analizar con la infraestructura la posibilidad de que el mantenimiento del sitio web sea por parte de la empresa.
- Plan de Contingencia: En caso de recuperación tardía crear una página simple y temporal que provea nuestros servicios.

En caso de: Caída de la página de la infraestructura los usuarios no podrán acceder a la información provista por el sistema.

6. - R6 - Caída del servidor web (Alarmas).

- Probabilidad: 0.3
- Impacto: 5
- Exposición al riesgo: 1.5
- Mitigación: Estudiar, analizar e investigar el servidor que garantice mayor disponibilidad.
- Plan de Contingencia: Utilizar algún tipo de sistema vía sms para informar a los usuarios.

En caso de: caída del servidor web no podrá alertarse a los empleados de problemas como congestión o caídas de TRs con lo cual habría que implementar algún tipo de interfaz alternativa que utilice mensajes de texto.

7. - R7 - Desarrollo de proyecto centralizado.

- Probabilidad: 0.1
- Impacto: 6
- Exposición al riesgo: 0.6
- Mitigación: Imponer pautas a cumplir por cada uno de los encargados del proyecto para evitar centralización.
- Plan de Contingencia: Incrementar tiempo de proyecto.

En caso de: centralización del proyecto el mismo tenderá a desarrollarse en más tiempo.

8. - R8 - Incremento de requerimientos en tiempo de construcción.

- Probabilidad: 0.5
- Impacto: 5
- Exposición al riesgo: 2.5
- Mitigación: Planificar lo mejor que se pueda de manera pesimista.
- Plan de Contingencia: Negarlos, en caso de no ser posible llevarlos a cabo pero intentando convencer a los stakeholders de un cierto incremento en el tiempo de entrega.

En caso de: incremento de los requerimientos el tiempo de desarrollo del proyecto se verá necesariamente en aumento.

9. - R9 - Baja en los recursos.

- Probabilidad: 0.1
- Impacto: 9
- Exposición al riesgo: 0.9
- Mitigación: –
- Plan de Contingencia: Incrementar tiempo de proyecto / incorporar nuevo recurso.

En caso de: una posible baja de alguno de los recursos que llevan a cabo el proyecto, el tiempo de desarrollo en el mismo tenderá a incrementarse.

10. - R10 - Mal funcionamiento de la CMUM (Componente muy usada en el mercado).

- Probabilidad: 0.1
- Impacto: 5
- Exposición al riesgo: 0.5
- Mitigación: –
- Plan de Contingencia: Cambiar la componente.

En caso de: mal funcionamiento de la componente muy usada en el mercado para alguna TR, se puede producir la situación de que la misma deje de funcionar, por lo cual hay que detectar la situación rápidamente y solucionar el problema.

11. - R11 - La arquitectura no satisface los requerimientos.

- Probabilidad: 0.4
- Impacto: 9
- Exposición al riesgo: 3.6
- Mitigación: Incrementar el tiempo dedicado a la arquitectura lo más que se pueda.
- Plan de Contingencia: Solucionar los posibles problemas incrementalmente en tiempo de ejecución.

En caso de: generar una mala arquitectura, se pueden producir muchos problemas posibles tales como lentitud en las conexiones, datos erróneos o inconsistentes, falta de disponibilidad, lo que haría que nuestro software sea de mala calidad, con lo cual la idea es dedicarle el mayor tiempo posible validando en todo momento lo que se haya realizado. En todo caso se intentará realizar una arquitectura flexible para que cualquier posible cambio se aplicable.

12. - R12 - Caída del servicio Biggest Satellite.

- Probabilidad: 0.2
- Impacto: 8
- Exposición al riesgo: 1.6
- Mitigación: –
- Plan de Contingencia: cambiar el radio de triangulación para generar TRs virtuales y calcular promedios con datos previos a la caída.

En caso de: caída del servicio Biggest Satellite en caso de necesidad de acceder a sus servicios por la caída de una TR y no poder triangular estaríamos dejando de proveer datos lo cual no puede suceder en nuestro sistema.

13. - R13 - Caída red GSM.

- Probabilidad: 0.05
- Impacto: 10
- Exposición al riesgo: 0.5

- Mitigación: Ninguna, la red GSM no depende de nosotros.
- Plan de Contingencia: Utilizar el servicio Biggest Satellite a pesar de sus costos en conjunto con el sistema eólico.

En caso de: caída de la red GSM nuestro proyecto es obsoleto.

14. - R15 - Uso de sensores que no soporten cifrado de la información enviada a las TRs.

- Probabilidad: 0.4
- Impacto: 5
- Exposición al riesgo: 2.0
- Mitigación: Informar a los stakeholders que tomen decisiones de que productos comprar, la necesidad de comprar sensores seguros.
- Plan de Contingencia: Evaluar en el seguimiento del proyecto la idea de incrementar el tiempo de alguna de las iteraciones.

En caso de: utilizar sensores que se comunican en medios inseguros (Comunicaciones wireless, bluetooth, etc) y no cifren los datos que envían, podría suceder que un intruso alterara los datos en vuelo (mediciones) antes de que lleguen a la TR y podamos detectarlo, tendríamos datos incorrectos en el sistema y podríamos arrojar pronósticos (parcialmente) incorrectos.

4.2. Selección de principales riesgos

En esta sección usaremos una técnica provista por el SEI la cual consiste en armar una tabla, que nos permite, una vez armada nuestra lista de riesgos, clasificarlos según 2 criterios: probabilidad y severidad de impacto. De esa forma, podremos determinar cuales son los riesgos mas severos que podrían hacer fracasar nuestro proyecto. En dicha tabla se ubicará cada riesgo en su correspondiente celda de acuerdo al siguiente criterio:

En cuanto a probabilidad se clasificarán en:

- Baja : menos de 0.3
- Media : entre 0.3 y 0.6
- Alta : mayor a 0.6

En cuanto a severidad se clasificarán en:

- Marginal : menos de 3
- Normal : entre 3 y 6
- Crítica : mayor a 6

A continuación se podrán observar cada uno de los riesgos mencionados en la sección anterior en la tabla.

Severidad/Probabilidad	Alta	Media	Baja
Crítica		R11, R14	R1,R4,R9,R12,R13
Normal		R8,R6,R2,R5	R7,R10,R3,R15
Marginal			

Finalmente seleccionaremos los riesgos mas críticos para nuestro proyecto. En base a la tabla, decidimos que los 2 riesgos más críticos son R11 y R14, ya que son altamente probables y el riesgo de impacto es crítico. Luego, decidimos que el resto de los riesgos críticos eran los que tenían alta probabilidad o un impacto muy grande, es decir, dejamos de lado los riesgos R3, R7, R10 y R15.

Finalmente, nos pareció que a los 2 riesgos mencionados antes, deberíamos agregar como críticos los riesgos: R1, R5, R6 y R12 por un lado, y por otro R2, R8, R11 y R14. Los primeros 4 riesgos son riesgos que tendrían como consecuencia no poder cumplir con el objetivo del sistema, ya se no poder realizar pronósticos adecuados (1 y 5) o no poder alertar a la población / los stakeholders de las amenazas (6 y 12). Por otro lado, están los riesgos que harían fracasar nuestro proyecto de software desde el punto de vista del Plan de Proyecto (riesgos no específicos al tema del proyecto, riesgos estándar que

enfrenta un proyecto de software): no comprender los requerimientos correctamente, excederse en los plazos, no cumplir los atributos de calidad, es decir riesgos que hacen a la satisfacción del cliente.

Esta selección de riesgos se hizo para reducir la lista y poder monitorear durante el resto del proyecto los riesgos críticos para el proyecto, y concentrarse en ellos, restándole algo de atención a los menos críticos (que sin embargo, tampoco serán dejados de lado por completo).

4.3. Análisis de los riesgos obtenidos

Los riesgos encontrados pueden dividirse en dos categorías: riesgos técnicos y riesgos de negocio. En su mayoría son riesgos generales y no tan específicos: esto responde a la etapa del proyecto en la que nos encontramos (es una etapa temprana, en la que se tiene una visión mas general o “macro” del proyecto).

Si bien en la etapa de elaboración suelen aparecer más que nada riesgos técnicos, es entendible que también hayan aparecido riesgos de negocio (que suelen aparecer en la fase de inception) ya que por un lado las disciplinas se superponen (que en la fase de elaboración aparezcan riesgos técnicos no significa que sean los únicos, ya que a veces aparecen espontáneamente algunos riesgos) y por otro porque el documento generado en la fase de Inception (el enunciado) no incluía un documento de gestión de riesgos.

Estamos contentos con los haber utilizado mas de una técnica de identificación de riesgos ya que se complementaron muy bien: debido a nuestra formación académica y profesional tenemos un sesgo técnico muy grande y es difícil que se nos ocurran riesgos de negocio / gestión, con lo cual leer cuestionarios y listas de riesgos comunes nos ayudó a tener una visión más completa.

Finalmente, hay que entender que este es solo un análisis de riesgos inicial, y que la gestión de riesgos tiene que seguir durante todo el proyecto. A medida que se avance en las diferentes etapas del proyecto, algunos riesgos desaparecerán, y aparecerán otros nuevos.

Parte 5

Arquitectura

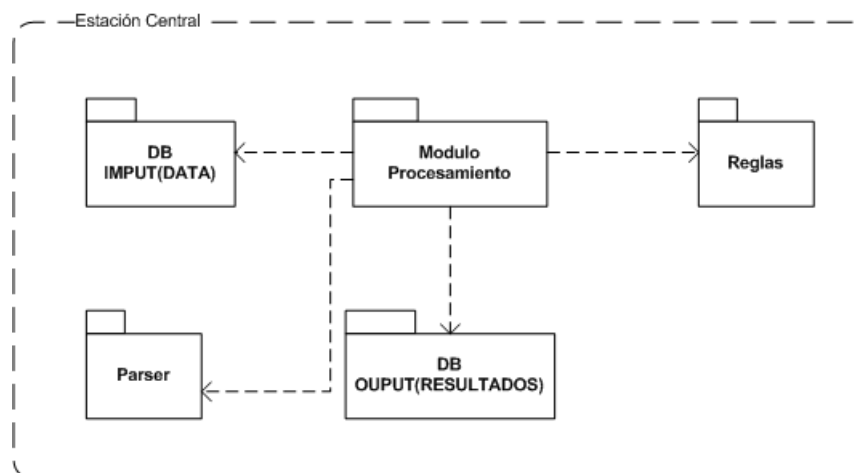
En esta parte se desarrollará la arquitectura general del producto. Analicemos estilos arquitectónicos, las tácticas correspondientes para poder contemplar los atributos de calidad y finalmente diferentes estructuras que nos conllevarán a realizar las vistas arquitectónicas apropiadas para lograr un buen entendimiento de las diferentes partes del sistema.

5.1. Módulos

Aquí vamos a mostrar una vista de módulos que nos permitirá analizar básicamente la sección de procesamiento de datos mediante modelos matemáticos en la Estación Central. Vamos a tener una unidad de código encargada de dicho procesamiento de modelos, pero para los cuales van a ser necesarios una serie de módulos que contribuyan a dicho procesamiento.

Entonces ésta vista estará compuesta por los siguientes módulos:

- **Modulo procesamiento:** Este módulo es básicamente un manejador del proceso de obtención parseo y procesamiento de los datos por modelos matemáticos. La idea es que una vez obtenidos los datos los mismos se puedan configurar o adaptar de tal manera que la inclusión de distintos modelos matemáticos no afecte el sistema en general y la modificación sea sencilla.
- **DB INPUT(DATA):** Este módulo comprenderá la operatoria asociada a la obtención de los datos a procesar de la base de datos que guarda los mismos previos al procesamiento.
- **DB OUPUT(RESULTADOS):** Este módulo comprenderá la operatoria asociada al depósito de los datos una vez procesados a la base de datos que guarda los resultados obtenidos.



- Parser: El Parser básicamente como mencionamos anteriormente se encarga de configurar la información en datos de modo tal que el intercambio entre distintos modelos matemáticos sea fácilmente configurable
- Reglas: Finalmente éste modulo es el encargado de tener las reglas de los modelos matemáticos. El módulo de procesamiento hará uso de éste modulo para procesar los datos obtenidos y posteriormente guardar los resultados.

La idea de utilizar ésta forma de procesar fue responder fácilmente al intercambio de modelos matemáticos. Al lograr bajo acoplamiento en los módulos y separando el módulo de reglas independientemente del resto de los módulos logramos que intercambiar este módulo de reglas no sea una tarea complicada y poder seguir usando el sistema rápidamente.

Como se mencionó anteriormente, las reglas estarán traducidas al lenguaje que estemos utilizando y además utilizarán estructuras de datos comunes a todas ellas para referirse a los diferentes datos. De éste modo éstas estructuras estarán fijadas y las reglas operarán sobre ellas sin necesidad de tener sus propias estructuras. Sólo constarán de definiciones de funciones y reglas aplicadas a los datos.

En la prueba de concepto daremos una breve explicación adaptada a dicha prueba para mostrar cómo podría simularse éste módulo.

5.2. Componentes y conectores

La primera de las estructuras que vamos a analizar es la asociada a entidades run-time. Para ello utilizaremos vistas de componentes y conectores que nos permitan identificar éstas entidades y poder instanciarlas como componentes o conectores. Esto nos dará una idea de como los diferentes componentes contribuyen a la ejecución de las funcionalidades solicitadas y como las tácticas aplicadas se corresponden al control de los diferentes atributos de calidad. Presentaremos tres vistas de acuerdo a los componentes que funcionan en cada uno de las tres grandes partes que componen nuestro sistema(Terminal Remota, Estación central y Sistema de Monitoreo) y también explicaremos la forma de comunicación y flujo de datos entre dichas partes, por lo cual también puede verse como una vista de Deployment combinada con componentes y conectores. Cada una de las tres vistas se encuentran al final de ésta sección por lo que es aconsejable analizarla en paralelo a medida que se lee la explicación del funcionamiento de las componentes internas.

5.2.1. Funcionamiento de las Terminales Remotas

Las Terminales Remotas representan los nodos o unidades que sensan y proveen información a la red telemétrica. Tienen a su cargo tareas como:

- Recopilación de información de sensores.
- Envío de datos a la EC.
- Configuración en la Agenda de los Sensores.
- Minimización del consumo de energía de las baterías.
- Mantenimiento de Estado de sensores.

Descripción de las componentes de la Terminal Remota

- **Procesador Estado:** Esta componente se encarga de la interacción con la componente muy utilizada en el mercado. Dicha componente proveerá la información acerca del monitoreo de paneles solares y baterías. Una vez obtenida dicha información ésta componente la procesará y determinará las acciones a llevar a cabo, enviando dichas ordenes al Switch Estado.
- **Switch Estado:** A partir de las órdenes generadas por el Procesador Estado, ésta componente se encargará de cambiar la forma de obtención de energía por parte de la TR ordenando apagar las baterías y habilitar los paneles solares o viceversa. Finalmente cada vez que se produce un cambio en la fuente de alimentación
- **ABM Sensor:** Esta componente es la encargada de llevar a cabo la alta, modificación y baja de sensores en la terminal remota.
- **Datos TR:** Esta base de datos tiene la información de los datos que van llegando por parte de los sensores y además el estado general de los mismos. Por otro lado también existe información asociada al consumo de energía de la Terminal Remota.
- **Adaptador Sensor:** Los adaptadores de sensores, permiten como su nombre lo indica normalizar los datos provenientes de los distintos sensores. Además es importante su existencia debido a las distintas formas de conexión que tienen cada uno de los sensores con la Terminal Remota. Está asociado a dos conectores de input y dos de output por los cuales se llevan a cabo las funcionalidades de configuración de las agendas de los sensores y la llegada de datos por parte de los mismos. Hay un adaptador por cada sensor comunicado con la Terminal Remota.
- **Configuración Agenda:** Esta componente se encarga de recibir las órdenes en el cambio de una agenda de un determinado sensor proveniente de la Estación Central y envía dicha información al adaptador del sensor correspondiente.
- **Receptor Datos:** Esta componente se encarga de ir almacenando los datos obtenidos por los diferentes sensores para su posterior envío. Además de acuerdo a la frecuencia de envío de los sensores puede determinar si existe alguno de ellos se cayó y avisar de dicha información a la componente de Inf Estado Tr/Sensor.
- **Inf Estado Tr/Sensor:** Esta componente se encarga de manejar el estado general de la TR(consumo de energía) y el funcionamiento de los sensores de la misma. Además se encarga de proveer ésta información almacenada en la base de datos Datos TR, a la Estación Central cuando la misma así lo requiera.
- **Sincronizador:** El sincronizador se encargará del envío de datos cada 1 minuto a la Estación Central. Es decir, cada un minuto envía la información obtenida por los sensores almacenada en la base de datos aplicando algún algoritmo para solucionar el problema de las distintas frecuencias de envío de los sensores. Una vez armada la transmisión envía dicha información a la componente Comunicación EC.
- **Comunicación EC:** Esta componente se encarga del protocolo de comunicación segura y confiable con la Estación central. Dicho protocolo se detallará con mayor detalle posteriormente en éste informe.

Tácticas elegidas en las Terminales Remotas

En las terminales remotas se utilizaron tácticas que garanticen de disponibilidad de los datos que se recolectan de los diferentes sensores, antes de ser enviados, y también luego de enviarse, ya que permanecen almacenados hasta que los mismos lleguen a la Estación Central de forma correcta. Se tuvo en cuenta la disponibilidad de toda la TR en si, tratando de evitar que por la caída de algún componente, la TR deje de funcionar, es decir, evitando que deje de censar los datos de los sensores o que deje de enviar los mismos a la Estación Central.

Uno de los componentes mas importantes, el componente relacionado con la comunicación y del envío de datos (*Comunicacion EC*), dispone de redundancia pasiva, tratando de disminuir el tiempo en el que la TR se encuentra no disponible para la Estación Central.

Los componentes mas críticos, y que disponen de redundancia activa, son los relacionados con el tratamiento y almacenamiento de los datos obtenidos de los sensores, esto ayuda a lograr los atributos relacionados con la disponibilidad de los datos, de la TR. Estos componentes son: *Receptor Datos*, donde confluyen todos los valores que los sensores envían. El mismo, los toma y los guarda en la base de datos *Datos TR*. Cabe aclarar que este componente, conoce cuantos sensores tiene la TR, y además de esto, también conoce el intervalo de tiempo que cada uno de los sensores tiene para enviar los datos, por lo tanto es el que detecta la posible caída de alguno de ellos, notificándolo al componente *Inf Estado Tr/Sensor*, por eso es importante que no esté fuera de servicio mucho tiempo, con lo cual, la redundancia activa nos permite disminuir este tiempo al mínimo; *Sincronizador* es el encargado de llevar el estado de los datos en vuelo entre la TR y la Estación Central. Como los datos que se envían, no son eliminados de *Datos TR* sino hasta que llega la confirmación de su llegada, el *Sincronizador* tiene como función, limpiar *Datos TR*, dejando solo los datos que van a ser enviados y los que se encuentran en camino y no fueron confirmados. La redundancia activa en este caso permite bajar al mínimo el tiempo en que este componente se encuentra fuera de servicio, ya que los mensajes de confirmación de llegada son considerados críticos para la comunicación confiable; *Datos TR*, lugar donde se guardan todos los valores que se obtienen de los sensores, y como esta información se va modificando cada momento, no es posible implementar una táctica con redundancia pasiva, ya que algunos datos se pueden perder si la componente principal se cae y no llega a actualizar la componente secundaria. por lo tanto es necesario que la información no se pierda, es por esto que se decidió implementar redundancia activa, y donde todos los datos estarán duplicados.

Los componentes que no son demasiado críticos, son aquellos que se encargan de enviar mensajes de un lado a otro y de recibir órdenes para obtener ciertos datos. En estos casos la elección de redundancia pasiva, resultó ser la mejor, ya que cuando uno de los componentes se cae, el secundario toma su lugar y continúa con las tareas sin tener un tiempo muy elevado fuera de servicio. Se considero que este tiempo debía ser pequeño, pero el hecho de utilizar redundancia activa en estos componentes no se justificaba, ya que el margen de este tiempo es aceptable.

5.2.2. Funcionamiento de la Estación Central

La Estación Central representa el gran cerebro del sistema. Tiene a su cargo funcionalidades tales como:

- Comunicación con las terminales remotas ya sea para conocer su estado, para recibir datos o bien para configurar agendas.
- Comunicación con el Sistema de Monitoreo para proveer la información necesaria a agentes externos e internos del sistema.
- Procesamiento de datos.
- Estado general del sistema.
- Estrategias alternativas como triangulaciones, Biggest Satellite, etc.
- Mantener consistencia, completitud y coherencia en la información.

Descripción de las componentes de la Estación Central

- **Históricos:** Este base de datos contempla la información histórica recopilada por la red telemétrica, es decir todos los datos desde el comienzo de la ejecución del sistema hasta el día actual. Esto sirve para aplicar algoritmos para la detección de posibles outliers o mismo para recuperar y comparar datos históricos con actuales. La implementación de la misma no es necesaria a esta altura del diseño pero según nuestros conocimientos podría llevarse a cabo algún tipo de base de datos similar a un Data Warehouse, lo cual resulta muy cómodo a la hora de tomar decisiones y comparar con datos históricos.

- **Resultados:** Esta base de datos va almacenando los resultados del procesamiento de los datos provenientes de las terminales remotas. Esta información es la que necesita verdaderamente ser confiable, precisa y en tiempo real para poder dar a conocer las predicciones correctas acerca del sistema meteorológico.
- **Data:** Esta base de datos almacena temporalmente (Suponemos alrededor de un día) los datos provenientes de las distintas terminales remotas a ser procesados, es necesario almacenar ésta información para triangulación y procesamiento correcto de los datos completos.
- **Estado:** Esta base de datos contempla el estado de las terminales remotas en tiempo real, ésta información es necesaria para la triangulación y o utilización del sistema Biggest Satellite y para informar en la página web y operarios acerca del estado real del sistema.
- **Información/Rollback:** Es necesario llevar a cabo algún tipo de Log o seguimiento de las acciones (llegada de datos, etc) para poder tomar decisiones acerca de la vuelta a estados consistentes en caso de fallas. Esta base de datos lleva a cabo dicho seguimiento.
- **Recepción Segura:** Esta componente representa el otro extremo de la comunicación con las terminales remotas. Vamos a suponer que existen muchos de estos componentes asignados estáticamente a cada una de las terminales remotas para evitar cuellos de botella. La componente se encarga del protocolo de comunicación y la seguridad en el envío/recepción de mensajes con las Terminales Remotas asignadas y luego envía ésta información a un divisor de mensajes asignado también estáticamente.
- **Divisor mensajes:** Esta componente se encarga de transferir los distintos tipos de mensajes a cada componente asociada a ellos. Los de estado a la componente Estado y los de datos a la componente Data Manager. Es importante reflejar que las transmisiones a este punto ya están completas porque el protocolo de comunicación es llevado a cabo por la componente Recepción Segura.
- **Info Tráfico:** Esta componente se encarga de enviar información de tráfico al sistema de monitoreo cuando éste así lo requiera, para ellos en caso de recibir una petición de tráfico particular, consulta a la componente recepción segura que es la que tiene dicha información por la funcionalidad que emplea
- **Interfaz SE:** Esta componente se encarga de recibir la información del sistema eólico capturada por el sistema de monitoreo, y transformarla a un formato útil para compararla, luego la envía al Arbitro 1.
- **Data Manager:** Esta componente centraliza todos los datos obtenidos por cada una de las Tr, y los va almacenando en Data. Cuando los datos están completos se los envía al Arbitro 1. El data manager también se encarga de ir llevando un Log para en caso de fallas poder regresar a un estado consistente.
- **Arbitro 1:** El árbitro 1 compara los datos que son comparables con los del sistema eólico y selecciona los más precisos de acuerdo a la zona de la cual provengan los datos. Por ejemplo se dice que los datos del sistema eólico suelen ser más precisos en el sur del país. Una vez normalizados los datos se los envía al Arbitro 2.
- **Arbitro 2:** El árbitro 2 compara los datos con los de la base de datos histórica, los almacena pero además analiza la posible existencia de outliers. En caso de existir envía dicha información al sistema de monitoreo para generar warnings. Un vez analizada ésta información se comunica nuevamente con el Data Manager quien actualiza los datos en la base de datos si así fuese necesario.
- **Controlador Estado:** El controlador de estado es la componente encargada de llevar a cabo el estado del sistema y sus posibles transiciones. Es quién cada cierto tiempo consulta el estado de las TR y sus sensores y actualiza la base de datos de estado. Si le llega información en caso de caída de una Terminal Remota se comunica con el Administrador Fallas TR y luego actualiza el estado de dicha TR de acuerdo a la solución obtenida. Es consultado por el Monitor de Estado del Sistema de Monitoreo y se encarga de recibir la información de alta, baja, modificación y de TRs o sensores para actualizar el estado. Cuando se detecta el alta de una Terminal Remota también se comunica con el administrador de Fallas De Para que todo vuelva a funcionar normalmente.
- **Administrador fallas Tr:** Esta componente lleva a cabo la solución cuando se detecta la caída de una terminal remota. El primer paso es comunicarse con un triangulador, que en caso de poder triangular lo hace y si no devuelve algún mensaje que no pudo. Si ésto llegase a suceder el administrador de fallas se comunica con el BS Proveedor para que adquiera los servicios correspondientes a Biggest Satellite para la zona en donde la TR se ha caído. Una vez llevado a cabo éste procedimiento se comunica nuevamente con el controlador de estado informando qué solución se llevó a cabo. Finalmente cuando se detecta el alta de la TR nueva mente el administrador de fallas interrumpe la triangulación o la adquisición del sistema Biggest Satellite.
- **Triangulador:** Ante la caída de una terminal remota accede a la base de datos de estado para determinar si es posible triangular dicha terminal. En caso de que sea posible accede a la base de datos Data para obtener las mediciones de terminales utilizadas en la triangulación, calcula promedios y luego pasa los datos triangulados a la base de datos Data nuevamente. Si no pudiese triangular, ya sea porque no hay terminales remotas próximas o están caídas simplemente le contesta a Administrador de Fallas que no se pudo triangular.
- **BS Proveedor:** En caso de que se necesite usar el servicio BS se comunica con la BS interfaz para que acceda a ellos y luego determina que parte de los datos necesita para esa terminal remota caída. Una vez hecho ésto envía los datos a la base de datos Data y comienza a contar el tiempo transcurrido hasta que se recupere la Tr y se deje de proveer el servicio, enviando el consumo realizado a la Interfaz Pagos.

- **BS Interfaz:** Representa la interfaz de acceso al servicio Biggest Satellite mediante un protocolo Cliente-Servidor.
- **Interfaz Pagos:** La interfaz de pago almacena los minutos consumidos en el uso de Biggest Satellite, cada cierto tiempo, envía dicha información al Sistema de Monitoreo para transferirla al sistema de pagos.
- **Modificación Agenda:** Esta componente recibe las peticiones en la modificación de la agenda para un sensor de una TR en particular, y envía dicha información al divisor de datos asignado a la Terminal Remota cuyo sensor necesita de la configuración.
- **Procesamiento:** Cada cierto lapso de tiempo, ésta componente adquiere los datos en la base de datos Data, los últimos válidos, y procesa mediante la aplicación de modelos matemáticos dichos datos. Una vez llevada a cabo ésta tarea notifica al sistema de monitoreo la finalización del procesamiento y guarda los resultados en la base de datos resultados.

Tácticas elegidas en la Estación Central

En las vistas de arquitectura intentamos mostrar los principales componentes y las tácticas más fáciles de reflejar en un diagrama. De todas formas, hay algunas tácticas que no se aprecian en estas vistas (no las incluimos porque en caso de hacerlo dificultarían la comprensión de las vistas). En ésta sección comentaremos estas tácticas y además ampliaremos sobre las otras tácticas utilizadas, incluyendo una justificación de porque las elegimos.

Como señalamos anteriormente existirán varias componentes Recepción segura, asignadas cada una de ellas a un grupo de terminales remotas determinado (en principio podría ser por zona) de manera estática. Con ésto lo que intentamos reflejar es un balanceo de carga estático para no generar un único cuello de botella, de modo tal de no saturar una única componente, lo que hace que la llegada de datos sea más performante debido a que cada una de ellas recibirá y procesará un subconjunto de los datos recibidos. A su vez cada una de estas componentes tendrá asociado un único divisor de mensajes. Por su parte y ya que logramos performance en la llegada de los datos, éstas componentes no pueden fallar, ya que de hacerlo no estaríamos recibiendo datos de todas las terminales remotas con lo cual atenta a la completitud de la información, es por ellos que existe redundancia activa en éstas dos componentes.

Es importante también notar que todo tipo de comunicación entre partes es segura, ya sea Sistema de Monitoreo-Estación Central o mismo la de la Estación Central con las terminales remotas. La componente Recepción segura, no sólo se encargará del protocolo de comunicación con las terminales remotas sino que verificará la autenticidad de los datos por alguna modo de codificación y ya que cada terminal remota tendrá un código unívocamente identificatorio se podrá determinar si un mensaje proviene de ella o algún intruso intervino en el medio.

Como se mencionó la componente recepción segura encargada del protocolo de comunicación y seguridad es la que proveerá un correcto ordenamiento de los mensajes entrantes.

Pasemos a analizar ahora algunas cuestiones relacionadas con la performance y la confiabilidad de los datos procesados lo cual constituye uno de los punto más importantes en nuestro sistema. En general cuando hablamos de redundancia activa estamos considerando una sola componente de redundancia y aclararemos en caso de necesitar más de una. Por ejemplo en el caso de la componente procesamiento vamos a utilizar tres componentes activos, y ésto no sólo nos servirá para disponibilidad y tolerancia a fallas sino también para confiabilidad. Como las tres componentes van a realizar los mismos cálculos (también se intentará lograr una concurrencia o paralelismo de procesos dentro de cada una de las componentes en lo que sea posible) utilizaremos un voter (o Sistema de corroboración) de modo tal de lograr un sistema dúplex en la cual, las tres componentes procesan datos y se evalúa la mayor coincidencia de los mismos (es por ello la elección de un número impar de componentes). Esto nos proporcionará performance y confiabilidad en el procesamiento de los mismos y garantizará por otro lado un buen nivel de disponibilidad de ésta componente.

La existencia de componentes bien diferenciadas para el ABM de diferentes componentes hace que sea menos acoplado el sistema y las modificaciones puedan llevarse a cabo a partir de éstas componentes.

Como se definió en la arquitectura cada vez que se procesa información se avisa inmediatamente al sistema de monitoreo, por lo cual en caso de existir algún fenómeno meteorológico crítico el mismo será informado en tiempo y forma.

La interfaz con el sistema eólico en la Estación Central permite que los datos le lleguen al arbitro 1 en un formato comparable con los datos capturados por las terminales remotas, por lo cual la interoperabilidad es garantizada.

Al separar las interfaces con los sistemas externos como usuarios, operarios, otras empresas, etc, en el sistema de monitoreo con el procesamiento de los datos, la información almacenada, etc en la Estación Central probablemente sea más fácil cuestiones relacionadas con la facilidad en los cambios.

Una vez aclaradas éstas cuestiones adicionales al diagrama podemos destacar que utilizamos redundancia activa en todas

aquellas componentes que consideramos que su caída es crítica para nuestro sistema y atenta fuertemente a la disponibilidad del mismo. Sin embargo consideramos redundancia pasiva en componentes tales como las interfaces con Biggest Satellite o el triangulador ya que consideramos que como no siempre están en uso exclusivo tenemos un margen de tiempo para corregirlo en caso de fallas.

Finalmente, para asegurar la disponibilidad de mediciones de cada TR (un punto fundamental del sistema), implementamos la táctica de heartbeat, para poder determinar si una TR está caída. El motivo es muy simple, las TRs ya envían mensajes periódicamente: los datos que envían cada 1 minuto a la estación central (más alguna eventual retransmisión de mensajes). Entonces, lo único que necesitábamos era chequear si hace un cierto tiempo que no se reciben mensajes de cierta TR; con lo cual, utilizar heartbeat tiene un costo casi nulo en este caso, y nos permite detectar una caída rápidamente (esto es fundamental ya que hay que tomar acciones apropiadas inmediatamente).

5.2.3. Funcionamiento del Sistema de Monitoreo

El Sistema de Monitoreo es la parte del sistema encargada de la interfaz con los usuarios, clientes y otros sistemas externos. En este sector se prestarán diferentes servicios, ya sea mediante una interfaz de comunicación directa o mediante Web Services. Estos servicios tienen que ver, en su mayoría, a la visualización de los datos que el sistema tenga disponible en el momento. Éstos datos pueden ser:

- Los resultados de los datos procesados por la Estación Central, que pueden estar destinados a operarios internos o usuarios, como por ejemplo la página del Ministerio de Infraestructura, o para otros clientes externos que se espera tener en un futuro, como es el caso de la empresa AgroTop.
- Los datos que se refieren al estado general del sistema en tiempo real, el estado de la conexión entre las TRs y la Estación central, el estado individual de cada TR, cuáles de éstas están funcionando correctamente, cuales fueron trianguladas, y cuales no fueron trianguladas y se está utilizando el sistema satelital Biggest Satellite para obtener los datos de su zona.
- La información que es enviada en forma de mail a los responsables por alertas que se producen.
- La recepción de datos provenientes del Sistema Eólico, que son utilizados por la Estación Central para discernir que datos son mas precisos.

Como se puede observar, el Sistema de Monitoreo es la gran interfaz que tiene el sistema y por esta razón debe cumplir con los atributos de calidad relacionados con la disponibilidad de la información y la seguridad.

Descripción de las componentes del Sistema de Monitoreo

En ésta sección se nombrarán y explicarán los componentes que forman parte del Sistema de Monitoreo, así también como su rol dentro de la arquitectura.

- **Interfaz SE:** Esta componente es la encargada de comunicarse con el Sistema Eólico, implementando su protocolo de comunicación, y a su vez se encarga de comunicarse con la Estación Central, para enviarle los datos recibidos a la componente encargado del procesamiento de los mismos. Los dos conectores que esta componente utiliza son seguros.
- **Interfaz Pagos:** Esta interfaz se encarga de la transferencia mediante un protocolo sencillo con el Sistema de Pagos de los minutos acumulados por la utilización del Sistema Biggest Satellite, el canal debe ser seguro para que se cobre adecuadamente el uso del servicio.
- **ABM Cliente-Sensor-Tr:** Esta componente se encarga básicamente de las funciones de alta, baja y modificación de algún cliente externo, terminal remota o sensor, es por ello que contempla y se comunica con una base de datos que tiene un estado general de dichos elementos. A su vez recibe éstas configuraciones de la consola de mando de los operarios mediante una comunicación segura.
- **Programación Agenda:** Esta componente se encarga de enviarle los datos de configuración recibidos por la consola de mando a la Estación Central. La comunicación entre la consola de mando es segura se podría utilizar SSL o similares, cuestiones que probablemente se puedan establecer mejor en las iteraciones de construcción asociadas a dichos casos de uso.
- **Datos:** Esta componente es avisada por la EC cada vez que un lote de datos es procesado por los modelos matemáticos, por lo cual luego de dicho aviso recupera dichos datos de los resultados almacenados en la base de datos(Resultados) y se encarga de transferirla al sistema de visualización.

- **Outliers:** Esta componente es notificada por la EC cada vez que se detecta un dato que es considerado de tipo outlier. Luego de recibir esta información, la componente envía estos al componente Visualización para que se muestre por pantalla y sea analizado por algún experto. A esta altura, tenemos conocimiento de que los datos que son outliers, solo deben ser notificados, sin esperar respuesta de la persona que los analice.
- **Monitor Tráfico:** La función que cumple esta componente como su nombre lo indica es monitorear el tráfico existente entre los canales de comunicación de la Trs y la EC, para ellos consulta regularmente a una componente en la EC que se encarga de obtener dicha información. Una vez obtenida la información de tráfico, la misma es enviada a las componentes encargadas de la visualización para los operarios como para la que se encarga de proveer dichos servicios a la página web del ministerio. Finalmente ésta componente en caso de detectar un mucho tráfico (cuyos límites serán definidos al momento de implementar ésta funcionalidad) se comunicará con el sistema de alertas (Sistema de mails) para que se informe de dicha situación a los destinatarios de la misma.
- **Monitor Estado:** Al igual que la componente anterior monitorea regularmente el estado de las Trs/Sensores, para ello le consulta a una componente en la EC (Estado) quien mantiene una base de datos con dicha información. Es importante destacar que si bien es competente más a una decisión más cercana a la implementación que a la arquitectura, los tiempos de consulta o polling serán de fracciones de segundo para poder tener un sistema lo más parecido a uno en tiempo real. La información recopilada será enviada a las mismas componentes mencionadas anteriormente.
- **Sistema de mails:** EL sistema de mails actúa como las alarmas mencionadas en casos críticos, en caso de que le llegue información para informar, se comunicará con un servidor de mails para enviar las alarmas correspondientes a los destinatarios.
- **Visualización:** Esta componente recopila toda la información en tiempo real que le va llegando y la transforma en datos útiles para la visualización de la misma en las consolas de mando de los operarios.
- **Servicios Página:** Provee los web services a la página de la infraestructura con la información en tiempo real que va adquiriendo. La comunicación es Cliente(Página infraestructura) Servido(Servicios Página).
- **Servicios Empresas:** Filtra la información requerida para cada empresa y como la componente anterior la comunicación es mediante el protocolo Cliente Servidor.
- **Consola de mando:** Es la componente que se encarga de la comunicación con los operarios, ya sea para mostrar la información(Estado de TRs en colores, Trafico, Etc), así como para cambios en las agendas de los sensores o ABM de TRs, Sensores y Clientes.

Tácticas elegidas en el Sistema de Monitoreo

Como ya se sabe, uno de los principales atributos de calidad es que el sistema este disponible en todo momento. Por este motivo, la mayor parte del sistema de monitoreo dispone de algún tipo de redundancia para cubrir este exigencia.

Los componentes de uso mas critico, y que son usados en todo momento, disponen de redundancia activa, ya que permite tener un tiempo mínimo, casi nulo de estado fuera de servicio. Estos componentes son los encargados de brindar la información en tiempo real y los resultados obtenidos luego de procesar la información que envían las TRs. Esta medida se pensó para satisfacer el atributo de calidad relacionado con la disponibilidad de los datos hacia los usuarios.

Las componentes de uso menos crítico o que no requieren un estado para trabajar (es decir, que tiene una función mas bien de interfaz), utilizan redundancia de tipo pasiva. Esta medida, si bien tiene un tiempo de fuera de servicio un poco mas alto que la redundancia activa, no es un tiempo que se considere alto a la hora de la caída de un componente. Las componentes que utilizan este tipo de redundancia son aquellas que brindan sus funciones a clientes externos, realizan la comunicación con otros sistemas, y realizan tareas que pueden ser tolerables a la caída de un componente por un tiempo un poco mayor (como es el caso de la configuración de la agenda de un sensor).

Por último, las componentes que son bastante mas tolerables a fallas, utilizan repuestos. Este es el caso de la Interfaz de Pagos y ABM Cliente-Sensor-TR, que pueden estar un tiempo considerable fuera de servicio.

Estas decisiones ayudan como ya se dijo a los atributos de disponibilidad del sistema. Pero además de esto, las comunicaciones hacia los clientes externos, los usuarios y hasta la comunicación con la Estación Central requieren ser seguras, para evitar que intrusos puedan interferir en los datos que se envían en todo momento. Es por esta razón que los conectores, hacia el exterior del Sistema de Monitoreo son seguros e implementan algún protocolo que garantice la integridad y la confidencialidad de los mismos. Con esto se cubren los atributos de calidad relacionados con la seguridad de los datos que envía y recibe el sistema a usuarios y clientes.

5.2.4. Comunicación entre la Estación Central y el Sistema de Monitoreo

Estas dos partes del sistema, están conectadas mediante varios conectores como se puede observar en los diagramas de arquitectura de cada una. Una de las principales características que se puede observar en esta conexión, es el hecho de que los conectores utilizados para el transporte de los datos, son seguros. Esto se debe a que la parte del Sistema de Monitoreo puede estar físicamente separada de la Estación Central, y por este motivo se requerido este tipo de comunicación segura, para evitar que alguien pueda alterar la información que viaja de una parte a la otra.

A continuación se describirán los conectores entre estas dos partes, nombrando las componentes que unen. Vale aclarar que cada uno de los conectores que se explicarán están identificados por una etiqueta en los correspondientes diagramas de arquitectura.

- **I1:** Este conector une las dos componentes que procesan los datos correspondientes al Sistema Eólico, Interfaz SE de la Estación Central e Interfaz SE del Sistema de monitoreo. Por último, este conector es de tipo *OneWay*.
- **T1:** Este conector es el encargado de transportar la información del tráfico en la red de comunicación de las TRs que le brinda el componente Info. Tráfico de la Estación Central al componente llamado Monitor Tráfico en el Sistema de Monitoreo. Este conector es de tipo *Client-Server*, donde el extremo cliente en este caso se encuentra en el lado del Sistema de Monitoreo.
- **E1:** Es el conector encargado de transportar la información referida al estado de las TRs entre el *Controlador de Estado* de la EC y el *Monitor de Estado* en el SM. Este conector es seguro y es de tipo *Client – Server*, donde el cliente se encuentra del lado del SM.
- **OL:** Conector encargado de transportar la información relacionada con los outliers que se detectan en el sistema. Este es seguro y es de tipo *OneWay*, y une el componente *Arbitro2* de la EC, con *Outliers* del SM.
- **D1:** Conector de tipo *OneWay*, donde se envían las notificaciones desde la componente *Procesamiento* en la EC a la componente *Datos* en el SM, indicando que se han generado nuevos resultados.
- **D2:** Es el complemento del conector D1, y es utilizado para transportar los datos referidos a los resultados de los modelos matematicos. Ya sea cuando hubo una notificación previa, enviada por D1, o cuando alguno de los clientes externos desean obtenerlos. Este conector es de tipo *Client – Server*, donde el cliente se encuentra del lado del SM. Une las componentes *Resultados* en la EC, con *Datos* en el SM.
- **A1:** Este conector se utiliza para transportar la información referida a la modificación de la agenda de una TR. Une las componentes *Programaci'on Agenda* en el SM, con *Modificaci'on Agenda* en la EC. Este conector es seguro y es de tipo *OneWay*.
- **ABM1:** Es el encargado de comunicar el componente *ABM Cliente – Sensor – TR* del SM con el *Controlador Estado* de la EC, y de transportar la información del sensor o TR que se instalo hacia el *Controlador Estado*. Es de tipo *OneWay*.
- **P1:** Es el encargado de transportar la información relacionada al pago por el uso del servicio del sistema de Biggest Satellite. Es de tipo *OneWay* y comunica las componentes *interfaz Pagos* en ambas partes.

5.2.5. Comunicación entre la Estación Central y las Terminales Remotas

La comunicación entre la Estación Central y las Terminales Remotas es una parte crítica del sistema, ya que es en este conector que se transmiten todas las mediciones que realizan las TRs y se envían a la Estación Central para su procesamiento.

Físicamente, los datos viajan en SMS, Short Text Message mediante la red GSM. Esta comunicación tiene que ser segura y confiable: se tiene que preservar la integridad y la confidencialidad de la información que se transmite, y tenemos que tener seguridad de que los datos no se pierden durante la transmisión. Como la tecnología GSM no nos provee ni seguridad ni confiabilidad, tuvimos que diseñar un protocolo que si lo haga.

Seguridad en las comunicaciones

La forma que se nos ocurrió de transmitir la información de manera segura es utilizar dos mecanismos de seguridad:

- Cifrar la información mediante un algoritmo criptográfico (puede ser simétrico o asimétrico) para mantener la confidencialidad de la información
- Aplicar al mensaje una función de hash (como SHA1 o MD5) al enviar el mensaje, y verificar que el hash sea correcto cuando se recibe el mensaje. De esta forma, aseguramos la integridad de la información.

Confiabilidad en las comunicaciones

La comunicación debe ser confiable, es decir debo obtener una confirmación de la recepción. En nuestro protocolo, cada vez que una Terminal Remota envía un mensaje a la Estación Central, ésta le envía una respuesta (ACK) a la terminal que emitió el mensaje, como confirmación de la recepción. Además, cada vez que envía un mensaje, la Terminal Remota espera la confirmación por parte de la EC. Si dentro de un lapso definido no recibe esta confirmación, asume que el mensaje se perdió y lo reenvía.

Ensamblado de mensajes

Las TR realizan mediciones de muchos datos distintos, que envían a la EC. Puede ocurrir que una transmisión de mediciones sea lo suficientemente grande como para no entrar en un mensaje SMS (que como su nombre lo indica, están previstos para mensajes cortos). Es por eso que nuestro protocolo tiene que poder soportar, que una misma transmisión lógica de datos, puede ser enviada en distintos SMS.

En cada Terminal Remota, el módulo de la comunicación con la EC se encarga de fragmentar un mensaje si es necesario y enviar cada parte a la estación Central. La respuesta que mencionamos antes, es por fragmento de mensaje, no por mensaje completo.

En la EC, se realiza el proceso inverso: en el módulo de Recepción Segura se reciben los fragmentos de los mensajes y se reensamblan. Este módulo pasa el mensaje una vez que está completo al siguiente módulo, para su posterior procesamiento. Los mensajes son transmitidos la siguiente módulo en orden: aún cuando el mensaje 2 de la TR 1 haya llegado completo, no se lo pasará al siguiente componente hasta que esté completo el mensaje 1 de la TR 1.

Protocolo

Para especificar el protocolo con mas formalidad, utilizamos máquinas de estados finitos (FSMs). Son 4:

- Una máquina que representa el canal de comunicaciones, por donde circulan los mensajes. Con esta FSM representamos las posibilidad de que los mensajes se pierdan.
- Una máquina que representa una Terminal Remota (la parte de envío y reenvío de mensajes).
- Y dos para la Estación Central: por un lado la fragmentación de mensajes y por otro el envío de la confirmación de recepción.

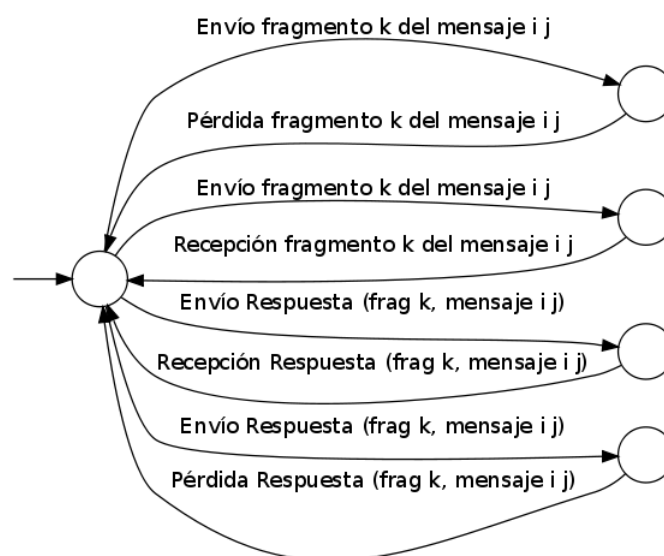


Figura 5.1: FSM del Canal de comunicaciones (el medio físico, en este caso la red GSM)

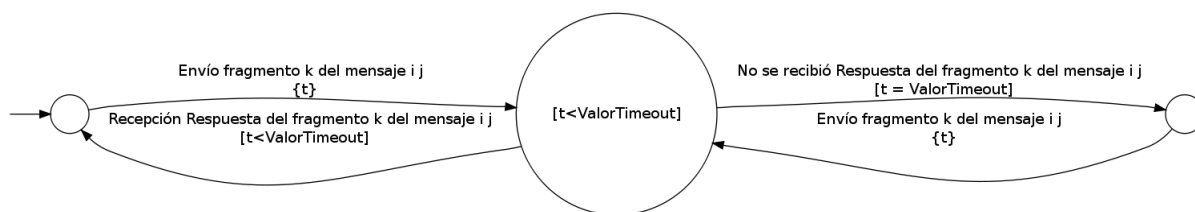


Figura 5.2: FSM de una Terminal Remota

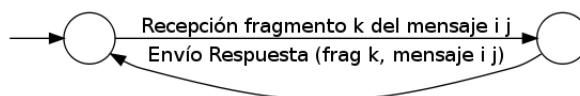


Figura 5.3: FSM que muestra el envío de respuesta de un mensaje

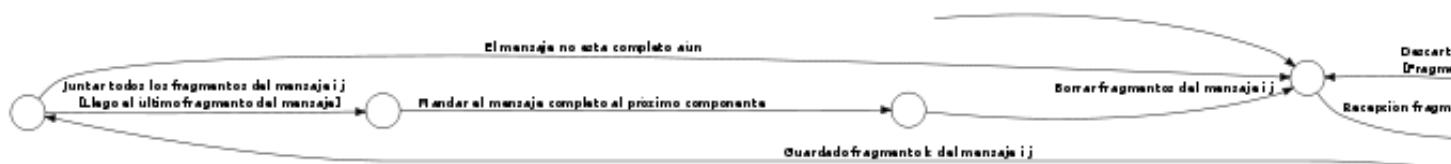
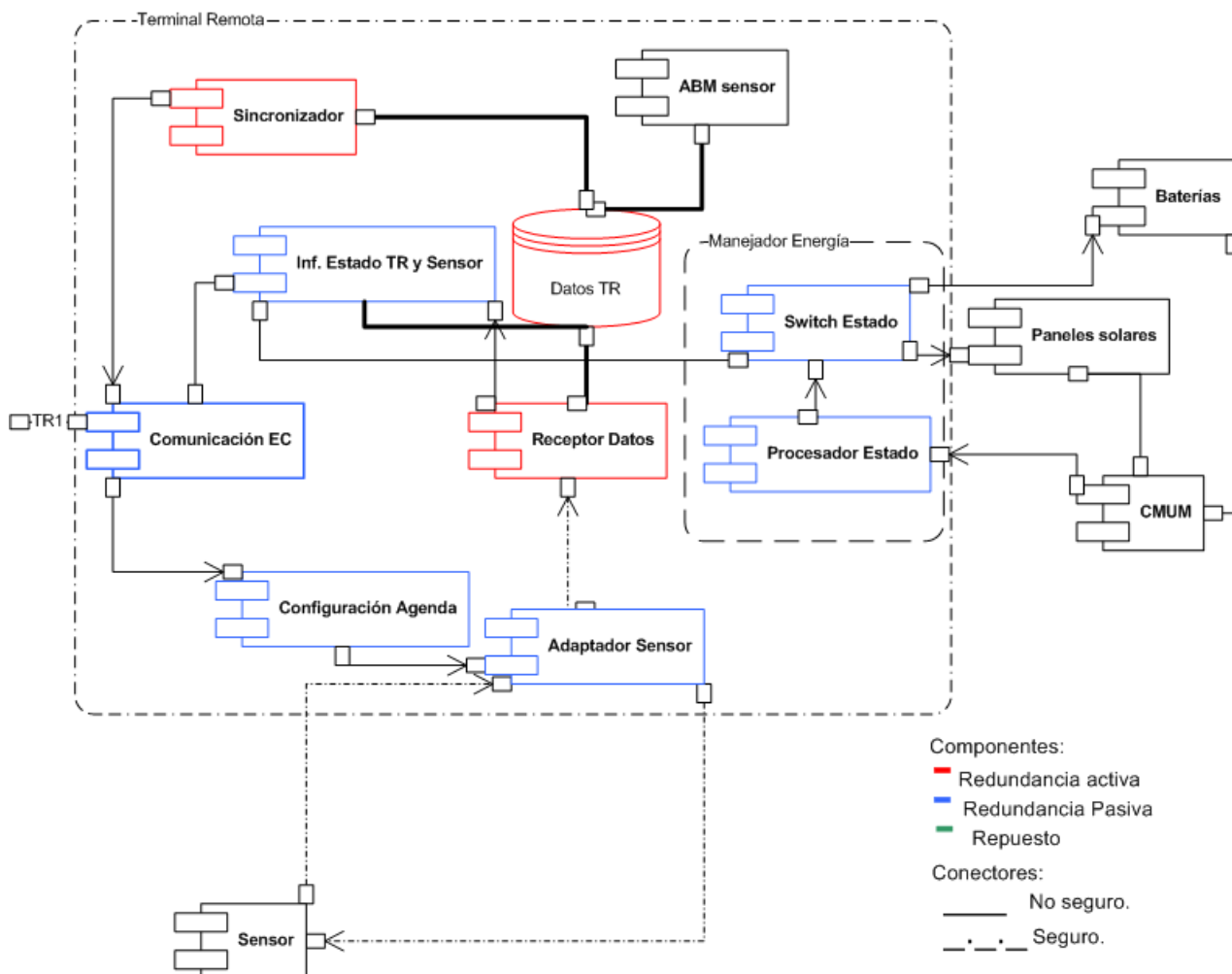
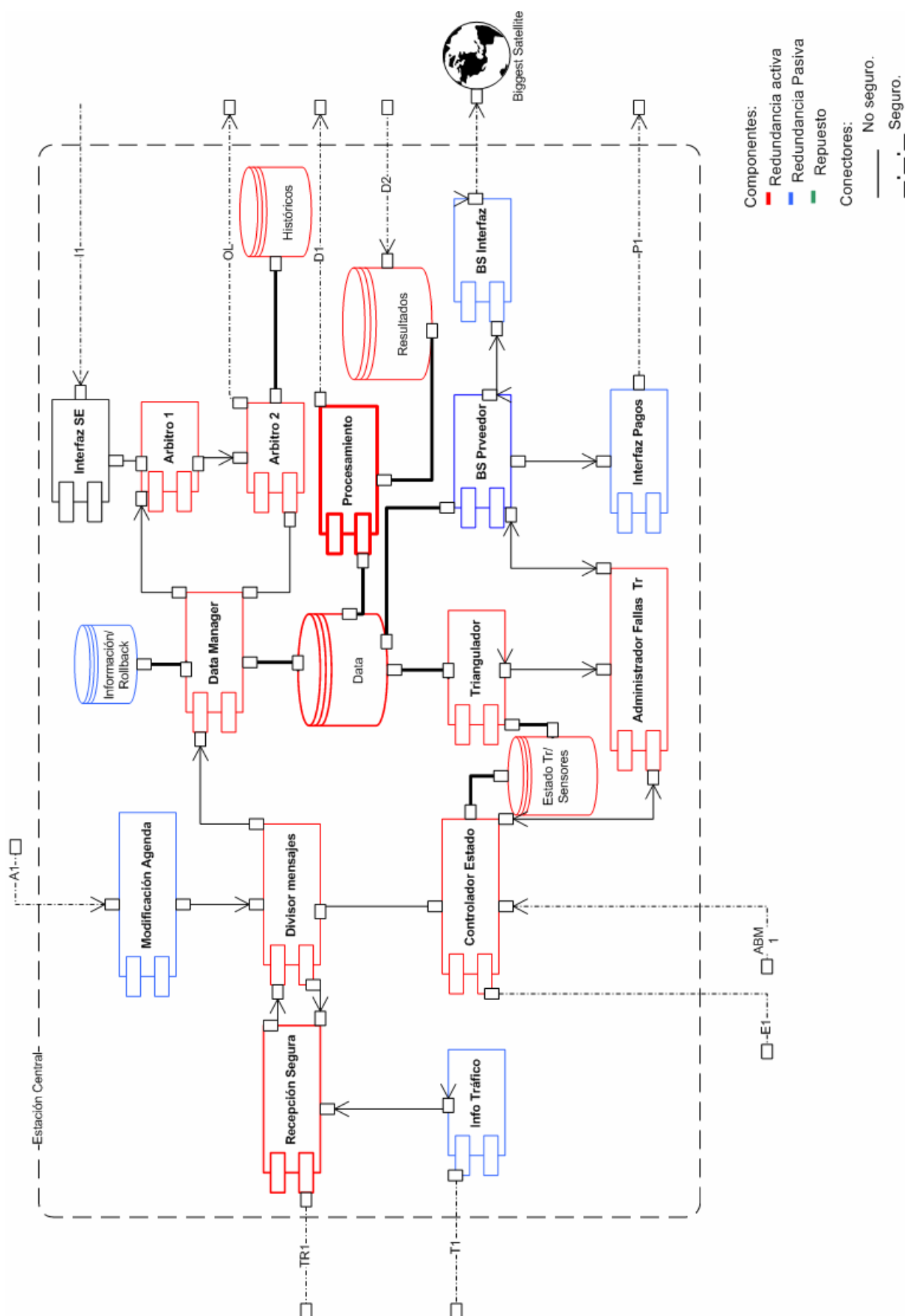
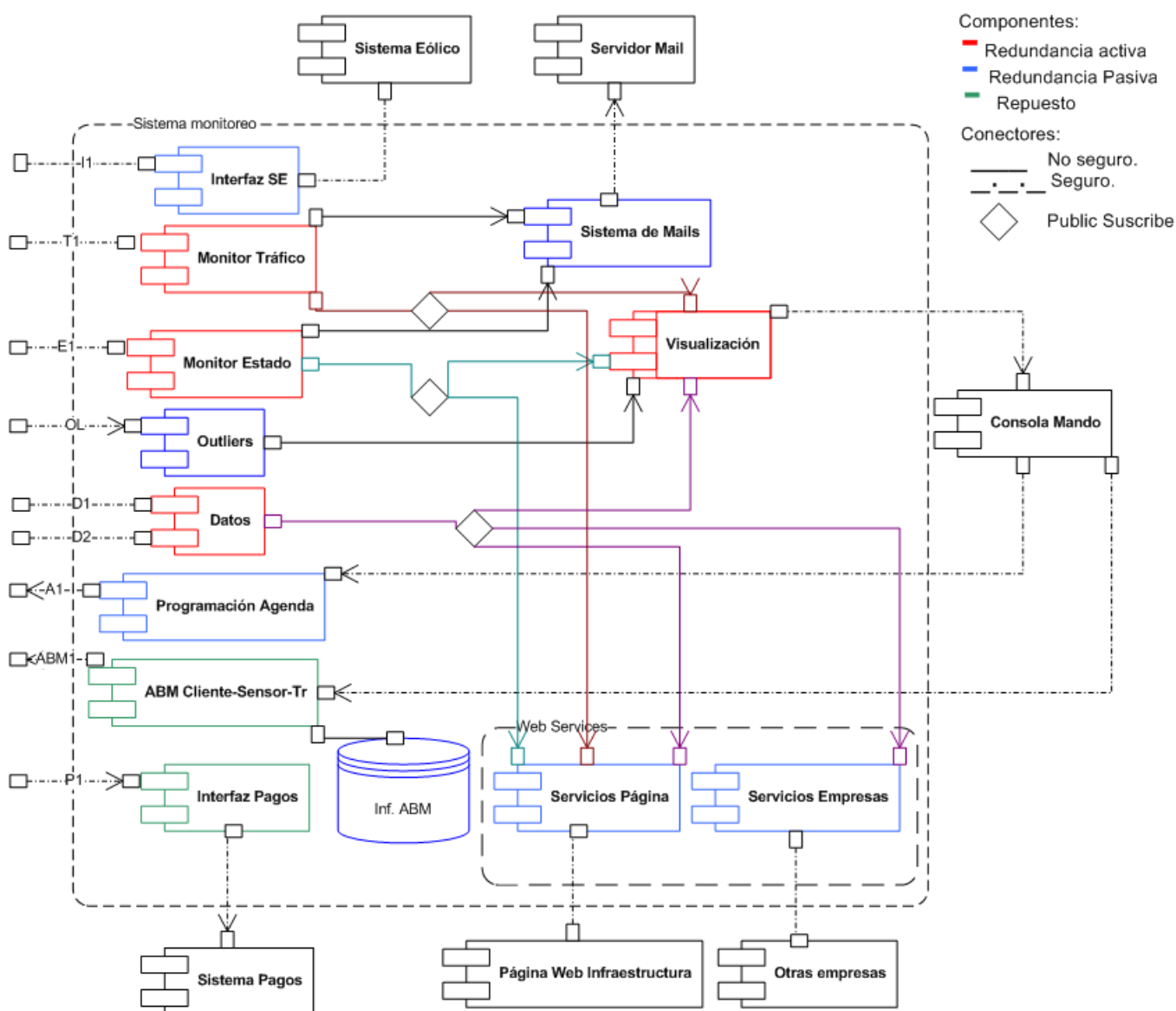


Figura 5.4: FSM del Canal de comunicaciones (FSM que muestra el ensamblado de mensajes en la EC)







Parte 6

Prueba de Concepto

6.1. Implementación de la prueba de concepto

6.1.1. Descripción del funcionamiento

Para realizar la prueba de concepto referida al protocolo de comunicación entre la Estación Central y las TRs, se tuvieron que implementar ciertos componentes descritos en la arquitectura del sistema, tanto de la Estación Central como en la TR. Además de esto se implementó un canal de comunicaciones, que simula la red *GSM* por donde los mensajes viajarán, por este motivo el mismo también simula la pérdida y el retraso de mensajes por congestión, que es lo que habitualmente ocurriría en la red *GSM* real.

El lenguaje de programación elegido para realizar la prueba de concepto fue *Python*, ya que es un lenguaje muy completo y permite realizar aplicaciones de manera rápida y fácil. Como los componentes implementados son entidades de runtime diferentes, la comunicación entre estos se realizó mediante *RPC* (Remote Procedure Call), donde cada uno de los procesos exportan funciones que se utilizan para intercambiar información y realizar todas las actividades necesarias para la prueba de concepto.

Como se mencionó anteriormente, se implementaron diferentes componentes de la arquitectura del sistema. Estos son los encargados del envío y recepción de los mensajes, que implementan además el protocolo de comunicación confiable y seguro. Para que el protocolo sea seguro, se agregó una capa de encriptación/desencriptación antes de enviar/recibir los mensajes, con eso se asegura que la información de los mensajes no sea corrompida en el camino. Para que el protocolo sea confiable se implementó un mecanismo en el cual, los mensajes enviados desde una TR y que llegan exitosamente a la Estación Central, son confirmados, mediante el envío de un mensaje de tipo ACK.

De la Estación central, se implementó el componente *Recepcion Segura*, donde se realiza el tratado de los mensajes que llegan. Los mensajes pueden ser fragmentados, ya que puede pasar que por la red *GSM* no puedan ser enviados de forma completa. Por este motivo en dicho componente se implementó la lógica necesaria para el tratamiento de las partes que forman un mensaje, y cuando se tienen todas estas partes, se forma el mensaje original y completo. Es por esto que es necesario tener un registro de los mensajes que van llegando y saber a que mensaje pertenece cada parte que arriba. Cada mensaje completo tiene un identificador y un timestamp que indica a que hora se envió, cuando el mensaje es fragmentado, existe además un identificador que indica a que parte pertenece cada fragmento. Esta es la información que utiliza *Recepcion Segura* para ir procesando la información que le llega. También en este componente se encuentra el sector de desencriptado de mensajes, ya que como el protocolo además de ser confiable, es seguro, cada fragmento que llega está encriptado y es necesario desencriptarlo para poder tratarlo.

De la Terminal Remota, se implementó el *Sincronizador y Comunicacion EC*. El *Sincronizador* para esta prueba de concepto genera datos al azar, que se presentan valores de los sensores de temperatura y humedad, y genera el mensaje con estos valores y con la información del mensaje que llamaremos encabezado. El mismo contiene un *ID* de la TR, un *Timestamp*, un *ID* del mensaje, un *ID* de la parte (que en este lugar siempre es uno), la *Cantidad de Partes* en las que está fragmentado el mensaje (que en esta parte es uno), el *Tipo de Mensaje* que puede ser *DATOS* o *CONTROL* y el *Contenido* del mensaje, donde se guardan los valores que se recibieron de los sensores.

Cuando un mensaje le llega a la componente *Comunicacion EC* desde el *Sincronizador*, ésta verifica que el mensaje

completo pueda ser enviado al *Canal*, si no lo fragmenta. Este proceso lo que hace básicamente, es obtener cada valor dentro del contenido del mensaje y generar nuevos mensajes, que tienen el mismo encabezado que el original, pero con los valores de *ID* de parte y *Cantidad de Partes* modificados según sea necesario. Una vez realizado esto, se procede a encriptar cada fragmento y enviarlo por el *Canal* hacia la Estación Central. Cabe aclarar que estos fragmentos al momento de ser enviados, son considerados como datos en vuelo, y para ellos se espera un mensaje con *Tipo de Mensaje* igual a *CONTROL* cuyo *Contenido* tenga un ACK que confirme la llegada de forma correcta a la Estación Central, momento en el cual, es considerado como fragmento entregado satisfactoriamente y es eliminado de los datos en vuelo.

Con respecto al *Canal*, este solo actúa de intermediador entre *Comunicacion EC* y *Recepcion Segura*, exportando funciones para enviar la información hacia una TR o hacia la Estación Central. Una característica que se comentó al principio es que el *Canal* además de reenviar mensajes de un lado a otro, contiene una lógica que permite simular la demora y la pérdida de los mensajes que llega. Para ello se definieron una serie de constantes y se consideró que sobre el canal existe una probabilidad de 0.05 de perder el mensaje, y el delay en él varía desde 0.05 segundos a 2 segundos.

6.1.2. Archivos fuentes utilizados

Aquí se mostrarán los archivos fuente utilizados para realizar los tests, y se explicarán que parámetros utilizan para poder correrlos en una pc.

- *Canal.py*: Simula el canal de comunicación que se usa para enviar los mensajes entre la TR y la Estación Central, Aquí se implementa el retraso y pérdida de mensajes. No requiere ningún parámetro adicional.
- *RecepcionSegura.py* (tiempo caída): Implementa la funcionalidad de componente de la arquitectura del sistema llamado del mismo nombre y explicado anteriormente. Requiere un parámetro adicional, que indica el tiempo en segundos en el cual se considera que una TR deja de funcionar desde el último mensaje recibido.
- *TR.py* (tiempo envío, cantidad de mensajes a enviar, ID de la TR, tiempo de vida de la TR): Este código en *Python* genera una instancia de *Comunicacion EC* (*comunicacionEc.py*) y una de *Sincronizador* (*sincronizador.py*), ambas son entidades de runtime distintas, y por este motivo *TR.py* hace que ejecución de estos procesos sea mas automática. Requiere 4 parámetros: El primero indica cuál será el intervalo de envío de los mensajes en segundos (en la realidad será de 60 segundos); El segundo indica la cantidad de mensajes que va a enviar la TR, esto se hizo para poder realizar pruebas y los tests; El tercero indica el *ID* que tendrá la TR; El cuarto y último indica el tiempo de vida en segundos que tendrá la TR. Esto se hizo también para realizar los tests de integridad.

6.2. Test de Integridad

Para ésta prueba de concepto se generaron tres test de integridad en forma de archivos batch. Los mismos simulan la interacción del componente *Comunicacion EC* de la Estación Central con diferentes TRs en diferentes situaciones. En cada test se puede apreciar el correcto funcionamiento del protocolo de comunicación descripto. Para realizar los test se dispone de un *canal* (*Canal.py*), que simula la pérdida y retraso de mensajes de forma aleatoria, También se dispone de un componente de la Estación Central (*RecepcionSegura.py*) que recibe los mensajes de las TRs por el canal, los ordena y los empaqueta (si estan completos), del lado de las TRs se implementaron 2 componentes (*sincronizador.py*, y *comunicacionEC.py*) que realizan las tareas descriptas en la sección de la arquitectura, cabe aclarar que estos dos componentes son creados por *TR.py*, y son entidades de runtime diferentes.

Cada uno de estos test corresponde a un archivo *.bat* que encapsula cada test y va ejecutando las distintas partes necesarias para realizar la comunicación y realizar los test. Cada uno de ellos intenta abarcar uno de los casos posibles en la comunicación entre las TR y la EC.

6.2.1. Test 01

- Archivo: Test 01.bat
- Descripción: En este caso se simulan cinco TRs de las cuales todas funcionan bien menos una, que luego de 45 segundos se cae (es decir, deja de funcionar) y 50 segundos más tarde se vuelve a levantar.
- Parámetros : Cada TR envía sus mensajes cada 20 segundos y la *RecepcionSegura* asume que si no recibe un mensaje luego de 40 segundos de una TR, la misma se cayó. La TR que deja de funcionar esta identificada con el ID 1 y envía 4 mensajes antes de dejar de funcionar, mientras que las demás TRs (que están identificadas con id's de 2 a 5) envían 9 mensajes en total.

- Resultado: Este caso de test se ejecutó, y fue verificado que se detecto la caída y la vuelta en marcha de la TR con ID 1. También se verificó que se hayan procesado todos los mensajes correctamente.

6.2.2. Test 02

- Archivo: Test 02.bat
- Descripción: Este es un test que simula el comportamiento normal, tanto de las TR como de la EC. En este caso todas las TR permanecen activas sin problemas durante todo el test.
- Parámetros : Cada TR envía 15 mensajes en total cada 20 segundos y la RecepcionSegura asume que si no recibe un mensaje luego de 40 segundos del último mensaje recibido de una TR, la misma se dejó de funcionar.
- Resultado: Este caso de test se ejecutó y fue verificado que no haya dejado de funcionar ninguna TR y que todos los mensajes que llegaron se procesaron correctamente.

6.2.3. Test 03

- Archivo: Test 03.bat
- Descripción: Este es un test que intenta simular la posibilidad de que el tiempo de llegada de los mensajes desde la TR hacia la EC sea muy alto (simulando una red congestionada por ejemplo) y así se asuma que la TR esta caída (cuando en realidad esto no es cierto), pero luego de un tiempo se recibe otro mensaje que continua con la comunicación.
- Parámetros : Cada TR envía 15 mensajes cada 20 segundos y la RecepcionSegura asume que si no recibe un mensaje luego de pasados 10 segundos del último mensaje recibido de una TR, la misma dejó de funcionar. Se agrega a este caso de test también, un delay de 3 segundos a cada TR antes de iniciarlas.
- Resultado: Este caso de test se ejecutó y se verificó que la recepción segura se de cuenta de la caída de una TR y que luego cuando lleguen los siguientes mensajes, se detecte que la TR sigue funcionando y se tome como activa nuevamente.

6.3. Procesamiento y modelos

En ésta sección vamos a mostrar una simulación de lo que respecta a la parte de procesamiento de datos por los modelos matemáticos.

La idea como se mencionó anteriormente en la arquitectura es poder incorporar distintos modelos matemáticos sin necesidad de implementar cambios globales a nivel de sistema. Para ello lo que tratamos de simular fue un módulo que se encargara básicamente de transformar los datos a procesar en estructuras que puedan ser utilizadas por cualquier modelo matemático representado por un conjunto de reglas.

En éste caso se intentó armar un diccionario de modo tal que cada Terminal Remota(cuyo id corresponde a las claves del diccionario) provea un conjunto de mediciones (valores del diccionario) representada a su vez por otro diccionario en el cual las claves sean los datos sensados tales como temperatura, presión, etc y los valores sean las mediciones de dichos datos, es mdecir supungamos que tenemos una sola terminal remota que sensa presiones, entonces el diccionario representado viene a ser el siguiente:

Clave : 1 (Terminal remota cuyo id es 1), Valor : (Clave : presion, Valor : lista de valores sensados)

De éste modo implentando ésta estructura de datos común, las reglas pertenecientes a un modelo matemático aplicarán sobre ésta estructura de datos independientemente del modelo.

Para poder probar la implementación lo que hacemos es correr los archivos de la siguiente manera:

```
python Modelos.py reglasX.py datosX.py(opcional)
```

Lo que hace la sentencia anterior es cargar el módulo de procesamiento (Modelos.py), el conjunto de reglas a aplicar(reglas1.py, reglas2.py, etc) y finalmente los datos en forma de diccionario (Asumimos que ya vienen con la estructura que deseamos, que otro módulo (Parser) se encarga de llevar a cabo dicha tarea).

Igualmente como se puede ver en la sentencia, la inclusión de datos es opcional, ya que de no proveer dicho archivo se generará un diccionario con valores aleatorios de sentido.

Hicimos estas dos posibles formas de obtener los datos para que sea sencillo el probado de las pruebas. Si generamos nuestros propios datos podremos analizar los diferentes resultados generados por las reglas, pero si por el contrario queremos valores aleatorios, también se brinda dicha posibilidad.

6.3.1. Datos y casos de test

Para probar las implementaciones, nos generamos básicamente dos archivos de datos: `datos1.py` y `datos2.py` y también dos conjuntos de reglas: `reglas1.py` (Que utiliza máximos para calcular) y `reglas2.py` (que utiliza promedios para calcular) y lo que hicimos fue crear todas las posibles ejecuciones, es decir para cada regla cualquiera de los dos archivos de datos o ninguno. De este modo obtuvimos 6 casos de test que se pueden ejecutar directamente con los archivos `Test X.bat` que se encuentran en la carpeta casos de test y dentro de ella en el directorio Test Procesamiento.

Lo que se intentó llevar a cabo con estas pruebas de test es el intercambio entre reglas de procesamiento de manera sencilla y no tanto así el resultado de los procesamientos por lo cual no explayaremos los resultados obtenidos y daremos principal interés a la correcta modificabilidad de las reglas.

Parte 7

Informe de avance

En esta parte de la documentación presentaremos una visión global del avance del proyecto, complicaciones, soluciones y otras cuestiones relacionadas con el desarrollo del proyecto.

Lo más relevante es el cumplimiento en los tiempos estimados, la preentrega se pudo realizar a tiempo y nos fue de mucha utilidad las correcciones para continuar con las etapas siguientes corrigiendo aquellas cuestiones que no eran del todo apropiadas en esa primera parte de la iteración (Ver Sección Evaluación, avance y ajuste a la preentrega).

Al tiempo en que determinábamos atributos de calidad, surgió el QAW de la cátedra que nos permitió terminar de cerrar aquellos atributos de calidad más relevantes para así poder comenzar a pensar en tácticas y finalmente establecer una arquitectura que soporte las necesidades del proyecto. Si bien es cierto que los atributos de calidad podían ser determinados por el enunciado, el QAW sirvió para definir prioridades y enfocarnos más en aquellos aspectos que a los stakeholders le resultaban más importantes, ya que los Atributos de Calidad “compiten” entre sí, no es posible satisfacer todos (por ejemplo, un típico trade-off es entre performance y modificabilidad).

La arquitectura fue uno de los momentos de mayor debate en el grupo, hubo que investigar y leer todo el material provisto por los docentes para poder seleccionar tácticas que nos permitiesen obtener las respuestas requeridas (especificadas en los escenarios) a los estímulos recibidos. Sin embargo la tarea de depuración de la arquitectura fue una de las más discutidas ya que implementamos una forma no muy óptima de llevarla a cabo y que se contrarrestaba con la idea de Brooks de centralizar el diseño de la arquitectura en lo posible en una persona o en un grupo con un marcado líder. En principio cada integrante del grupo pensó por separado y posteriormente a hablar de lo necesario a diseñar, una arquitectura propia. Luego se intentó ensamblar y hacer coincidir cada una de éstas arquitecturas lo que fue una tarea complicada. Finalmente optamos por tomar una de ellas (la de mayor completitud) e ir armándola y modificándola de acuerdo a las ideas comunes entre todos los integrantes. Así al fin, se logró establecer una arquitectura consensuada.

La existencia de un parcial en medio del desarrollo del trabajo práctico fue de cierto modo otro problema con el cual lidiar. Si bien no se dejó de lado por completo el trabajo práctico, si se bajó el ritmo de trabajo para poder estudiar para el parcial.

La implementación de la prueba de concepto no fue tan simple como esperábamos. Nos pusimos de acuerdo en el lenguaje a utilizar: Python. Creemos que fue una buena decisión, ya que pese a que solo uno de los integrantes del grupo había programado en Python previamente, todos conocíamos lenguajes del estilo Java o C#, que nos hubiera dificultado aún más la tarea por las características del lenguaje. El mayor obstáculo fue programar distintas unidades de runtime (nosotros utilizamos threads de Python) ya que, si bien cursamos la materia Sistemas Operativos, lo vimos mas que nada teóricamente.

En análisis de los riesgos fue progresando incrementalmente a medida que avanzaba el tiempo dedicado en el proyecto. Hubo una primera aproximación de riesgos iniciales y se sumaron algunos posteriormente a la preentrega, como por ejemplo no habíamos tenido en cuenta la incertidumbre en la duración de algunas tareas para la generación del diagrama Gantt. Es importante destacar que se llevaron a cabo mitigaciones a algunos de los riesgos más relevantes como las problemáticas en las estimaciones o en el armado de la arquitectura.

La semana adicional brindada por la cátedra para la entrega del trabajo práctico nos sirvió en gran medida para releer la documentación del mismo y realizar las correcciones necesarias en los documentos a entregar. Fue fundamental para terminar la prueba de concepto y diseñar casos de test para ella.

Parte 8

Bibliografía

Consultamos algunas fuentes adicionales para comprender mejor el método de desarrollo RUP y como documentar nuestra arquitectura. El material consultado fue el siguiente:

- Software Architecture in Practice, Second Edition. Len Bass, Paul Clements , Rick Kazman
- Paul Clements, Felix Bachmann y otros. Documenting Software Architectures, views and beyond.
- Davor Gornik. IBM Rational Unified Process: Best Practices for Software Development Teams. IBM White Paper
- Artículos varios de la página Web de Rational.

Parte 9

Conclusión

En este Trabajo Práctico, hicimos la prueba de hacer una parte del desarrollo de nuestro proyecto de software con RUP - Rational Unified Process. Fue un experimento interesante, ya que ninguno de nosotros había estado involucrado en un proceso de desarrollo tan grande en nuestra vida profesional. La ventaja de utilizar un método iterativo incremental es que nos permite hacer una iteración, probando así una parte completa del método (y no solo haciendo una parte incompleta de él).

En la próxima etapa del TP, seguiremos con métodos ágiles, que a priori, nos da la sospecha de que nos van a gustar más. Veremos si esto se confirma o no.

9.0.2. RUP: Fase de Elaboración

Con el fin de esta iteración, cumplimos con algunos (la mayoría) de los procesos previstos para la etapa de Elaboración[

- Analizamos el dominio del problema: tenemos la lista completa de los CU.
- Complementamos los requerimientos funcionales con la especificación de Atributos de Calidad, mediante la técnica de escenarios.
- Establecimos una base para nuestra arquitectura: que documentamos mediante vistas y la probamos con un prototipo ejecutable.
- Desarrollamos con cierto detalle el plan del proyecto.
- Armamos una lista de Riesgos del proyecto.

De todas formas, esto no significa que se haya finalizado la fase de Elaboración, hay cosas que faltaron. Por ejemplo, habría que especificar en más detalle una porción importante de los Casos de Uso del Sistema para poder pasar a la siguiente fase, Construcción.