

Sistema de Gestión de Transmontevideo

Descripción de la Arquitectura de Software

*Mauricio Farah
Felipe Troncoso
Julio Ariel Hurtado*

Bajo la dirección de:
*Cecilia Bastarrica
Daniel Perovich*

*Curso Arquitecturas de Software
Departamento de Ciencias de la Computación
Universidad de Chile
2007*

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Historial de Cambios

Versión	Autores	Fecha
<i>V0.1</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Ariel Hurtado</i>	<i>15 de abril de 2007</i>
<i>V0.2</i>	<i>Felipe Troncoso, Mauricio Farah Julio Ariel Hurtado</i>	<i>18 de abril de 2007</i>
<i>V0.25</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Hurtado, Daniel Perovich</i>	<i>3 de mayo de 2007</i>
<i>V0.3</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Hurtado</i>	<i>4 de mayo de 2007</i>
<i>V0.35</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Hurtado</i>	<i>5 de mayo de 2007</i>
<i>V0.4</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Hurtado</i>	<i>20 de mayo de 2007</i>
<i>V0.45</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Hurtado</i>	<i>22 de mayo de 2007</i>
<i>V0.5</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Hurtado</i>	<i>23 de mayo de 2007</i>
<i>V0.55</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Hurtado</i>	<i>24 de mayo de 2007</i>
<i>V0.6</i>	<i>Felipe Troncoso, Mauricio Farah, Julio Hurtado</i>	<i>24 de mayo de 2007</i>

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Tabla de Contenido

Página

1. INTRODUCCIÓN.....	5
1.1 CONTEXTO DEL PROBLEMA	5
1.2 PROPÓSITO	6
1.2. ALCANCE	6
1.4. ACRÓNIMOS	7
1.3. REFERENCIAS.....	7
1.4. ORGANIZACIÓN DEL DOCUMENTO	8
2. REPRESENTACIÓN DE LA ARQUITECTURA	8
2.1 REPRESENTACIÓN	8
3. METAS Y RESTRICCIONES DE LA ARQUITECTURA:.....	9
4. VISTA DE CASOS DE USO Y ESCENARIOS DE CALIDAD	11
4.1 MODELO DE CASOS DE USO	11
4.2 ESPECIFICACIÓN DE LOS CASOS DE USO RELEVANTES.....	13
4.3 ESPECIFICACIÓN DE LOS ESCENARIOS DE CALIDAD RELEVANTES	16
5. VISTA LÓGICA	18
5.1 PARTE ESTRUCTURAL: DESCOMPOSICIÓN MODULAR.....	18
5.1.1 <i>Primer Nivel de Refinamiento</i>	18
5.1.2 <i>Segundo Nivel de Refinamiento</i>	20
5.2 PARTE DINÁMICA (COMPORTAMIENTO).....	26
5.2.1 <i>View packet Unidades de Negocio Principales</i>	28
5.2.2 <i>View-packet Sistema Financiero</i>	31
5.2.3 <i>View packet Unidad de Transporte (Del segundo nivel de refinamiento)</i>	33
5.3 RELACIÓN ENTRE COMPONENTES DEL VIEW TYPE C&C Y LOS MÓDULOS DEL VIEW TYPE MÓDULOS	35
6. VISTA DE PROCESOS.....	36
6.1 MODELO DE PROCESOS PARA EL SGT	36
6.2 RELACIÓN CON LA VISTA LÓGICA.....	37
7. VISTA DE IMPLEMENTACIÓN.....	38
7.1 RELACIÓN CON LA VISTA LÓGICA.....	39
7.1.1 <i>Relación con el view type de módulos</i>	39
7.1.2 <i>Relación con la view Type C&C: Tecnología de Implementación de los conectores de la Arquitectura</i>	41
7.2 DIRECTRICES GENERALES PARA LA IMPLEMENTACIÓN DE LOS COMPONENTES.....	41

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

8. VISTA DE DESPLIEGUE (DEPLOYMENT) O FÍSICA.....	41
8.1 ARQUITECTURA TÉCNICA	42
8.2 TECNOLOGÍA REQUERIDA.....	44
9. RATIONALE.....	45
ANEXO 1. DESCRIPTOR DEL MÉTODO QAW ADAPTADO A UP	48
ANEXO 2. DESCRIPTOR DEL MÉTODO ADD ADAPTADO A UP	49

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Sistema de Gestión del TransMontevideo - SGT Software Architecture Document – SAD

1. Introducción

El sistema de gestión del Transmontevideo – SGT [Pero07] - ha sido tomado como el caso de estudio del curso de Arquitecturas de Software del Departamento de Ciencias de la Computación de la Universidad de Chile. El contexto de aplicación ha sido el sistema de transportes para la Ciudad de Montevideo – República del Uruguay, pero tomando información de sistemas de transportes similares como el de las ciudades de Bogotá en Colombia y Santiago en Chile.

La arquitectura de software será representada por medio del framework 4+1 [Kruc95] Vistas y a través de UML2.0 [OMG03]. La metodología guía del trabajo es el UP [Jaco99] acompañada de los métodos QAW [Barb03], ADD [Woj06] y ATAM [Kazm00] desarrollados por el SEI. Además se ha tomado como referencia los casos de estudio presentados en el Curso de Arquitecturas de Software [Ucur07]. El presente documento corresponde a la descripción de la arquitectura del sistema de gestión de sistema de transporte Transmontevideo SGT.

1.1 Contexto Del Problema

TransMontevideo es el plan de modernización del transporte público que modificará completamente la forma de viajar por la ciudad de Montevideo, mejorando en forma significativa la calidad de vida de sus usuarios y habitantes. El plan está impulsado por la Intendencia Municipal de Montevideo (IMM) y el Ministerio de Transporte y Obras Públicas (MTOP), con apoyo del Gobierno de la República. Tiene como misión implementar un nuevo sistema de transporte público, moderno, eficiente, integrado y con alto nivel de servicio para los usuarios [Pero07]. El sistema informático deberá dar soporte completo al plan, lo que incluye: tarjetas bip!, unidades de transporte (buses y taxis), central de operaciones, portal web y gestor económico de acuerdo al proceso de negocio presentado en [Pero07]. La figura 1 presenta las principales unidades de negocio del contexto sobre el cual SGT deberá funcionar, incluyendo los principales actores y trabajadores del negocio, así como su relación con las principales unidades de negocio identificadas.

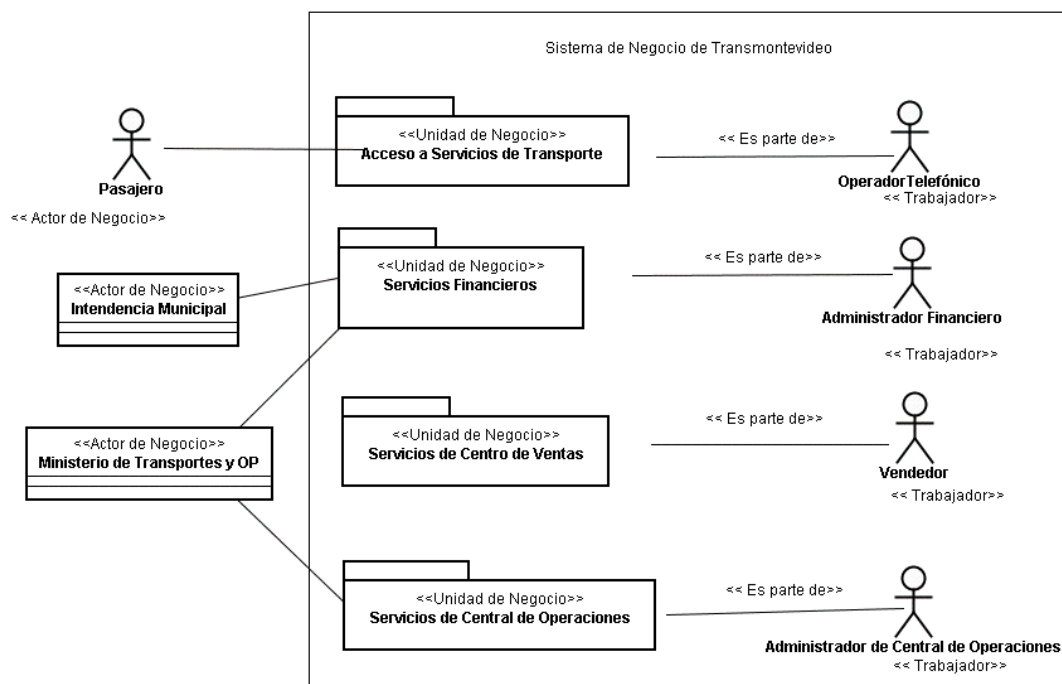


Figura 1. Modelo de Contexto de SGT

1.2 Propósito

El propósito de este documento es expresar de manera temprana y estable una descripción del SGT comprensible, coherente y de alto nivel de abstracción a los diferentes interesados en el sistema con el fin de que sirva de línea base para el desarrollo del producto. No es un documento de descripción de requerimientos. Sin embargo, este documento incluye los participantes, sus necesidades y los requerimientos clave para su construcción y expresa la arquitectura del sistema a través de un conjunto de vistas de acuerdo a las necesidades de los interesados y a las decisiones de diseño tomadas a partir de los compromisos establecidos por los requerimientos derivados. En esta primera versión cumplirá parcialmente este propósito al no haber una completa descripción de la arquitectura.

1.2. Alcance

Este documento es un documento de tipo académico y por tanto realmente no corresponde a una base sólida para la construcción real del sistema real. Sin embargo, el esfuerzo dedicado al caso de estudio ameritará el refinamiento para ser convertido en un documento de referencia para el desarrollo de la arquitectura en sistemas intensivos en software siguiendo el UP y algunos elementos aportados por métodos de requisitos, análisis y evaluación de arquitecturas de software.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

1.4. Acrónimos

ADD	Attribute-Driven Design Method
ATAM	Architecture Tradeoff Analysis Method
GPSN	GPS Network
J2ME	Java 2 Micro Edition
J2EE	Java 2 Enterprise Edition
J2SE	Java 2 Standard Edition
JCE	Java Cryptography Extension
JDBC	Java Data Base Connectivity
JMS	Java Message Service
JTS	Java Transaction Service
LAN	Local Area Network
MAN	Metropolitan Area Network
QAW	Quality Attribute Workshop
SAD	Software Architecture Document
SEI	Software Engineering Institute
SGT	Sistema de Gestión del Sistema de Transportes TransMontevideo
UML	Unified Modeling Language
UP	Unified Process

1.3. Referencias

[Barb03] Barbacci M. et Al. Quality Attributes Workshops. Technical Report CMU/SEI-2003-TR-016. SEI, 2003.

[Iver04] Ivers J. et. Al. Documenting Component and Connector View with UML 2.0. Technical Report CMU/SEI-2004-TR-008. SEI 2004.

[Jaco99] Jacobson I., Booch G., and Rumbaugh, J. The Unified Software Development Process. Addison-Wesley, 1999. ISBN 0-201-57169-2. 1999.

[Kazm00] Kazman, R. et. Al. Method for Architecture Evaluation. CMU/SEI-2000-TR-004. 2000.

[Kruc95] Kruchten, P. The 4+1 View Model of Architecture. IEEE Software. Volume 12 , Issue 6, November. 1995.

[OMG03] Object Management Group, OMG Unified Modeling Language Specification, Version 1.5 (OMG, 2003-03-01). 2005.

[Pero07] Perovich D. y Bastarrica C. Descripción del Problema: Sistema de gestión de Transmontevideo. Descripción del Problema. Departamento de Ciencias de la Computación, Universidad de Chile. 2007.

[Roh04] Roh, S. et Al. Architecture Modeling Language based on UML2.0. Proceedings of the 11th Asia-Pacific Software Engineering Conference (APSEC'04).

[Ucur07] Material del Curso CC61H. Disponible en <http://ucursos.ing.uchile.cl/> para usuarios registrados en el curso. Último acceso abril de 2007.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

[Woj06] Wojcik, Rob et. Al. Attribute-Driven Design (ADD), Version 2.0 (CMU/SEI-2006-TR-023).

1.4. Organización del Documento

El presente documento está organizado por un conjunto de capítulos, aún sin determinar completamente en esta versión inicial. En el capítulo 2 se presentan las metas y la restricción de la arquitectura. En el capítulo 3 se presenta la vista de casos de uso, en el capítulo 4 se presenta la vista lógica, el capítulo 5 presenta la vista de implementación. El capítulo 6 presenta la vista de procesos y el capítulo 7 la vista física (deployment) del SAD del SGT. Finalmente, el capítulo 8 presenta el razonamiento general seguido (Rationale, justificación) para la obtención del diseño arquitectural.

2. Representación de la Arquitectura

La arquitectura está representada por diferentes vistas utilizando la notación UML 2.0 con el fin de visualizar, entender y razonar sobre los elementos significativos de la arquitectura así como identificar y localizar los riesgos a ser mitigados en la fase de elaboración y en las primeras iteraciones de la construcción.

2.1 Representación

La arquitectura del SGT está representada siguiendo el enfoque de del framework 4+1 y las recomendaciones del proceso unificado y del método ADD. Las vistas incluidas en esta versión del documento son:

- **Vista de Casos de Uso y Escenarios de Calidad:** Describe los casos de uso más significativos, presenta los actores y una descripción de sus casos de uso asociados. De igual forma describe los escenarios de calidad más relevantes para la arquitectura.
- **Vista de Metas y Restricciones:** Describe restricciones tecnológicas, normativas, estándares, etc., los cuales influyen sobre las decisiones arquitectónicas, del producto y del proceso de desarrollo.
- **Vista Lógica:** Describe la arquitectura del sistema presentando varios niveles de refinamiento. Indica los módulos lógicos principales, sus responsabilidades y dependencias. Usa el view type Módulos para representar la estructura lógica y el view type Componentes y Conectores para representar el comportamiento.
- **Vista de Procesos:** Describe los procesos involucrados para darle sentido a la ejecución del sistema, así como sus relaciones de comunicación y sincronización.
- **Vista de Implementación:** Describe los componentes de deployment contruidos y sus dependencias.

Las vistas presentadas forman en su conjunto una especificación aún incompleta del

sistema (en desarrollo). La relación entre estas vistas se describe en la figura 2.

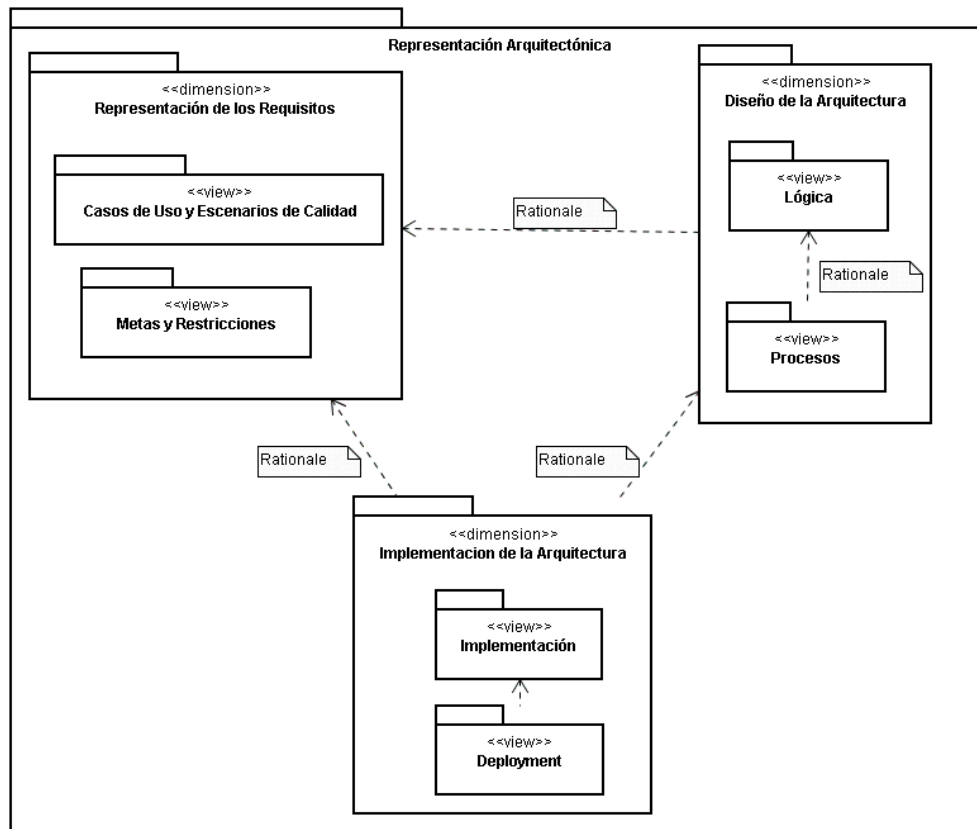


Figura 2. Representación de la Arquitectura

3. Metas y Restricciones de la Arquitectura:

Para definir las metas y restricciones se desarrolló una reunión donde se aplicó el QAW [Barb03]. Los participantes interesados fueron tres:

- **Representante de gerencia:** este rol lo representó Mauricio Farah como representante de los intereses de la alta gerencia a quienes les interesa generar valor a su negocio, así como imponer restricciones de tiempo y costos al proyecto.
- **Representante de los usuarios:** este rol lo representó Felipe Troncoso como representante de los usuarios de las interfaces que ofrece el sistema: pasajeros y administradores, así como todo lo referente a la calidad del servicio esperado.
- **Representante de la empresa desarrolladora:** este rol lo representó Julio Ariel Hurtado como representante de los intereses de una empresa desarrolladora que busca comercializar el producto en otras ciudades y que desea asegurar el mantenimiento de software con pocos traumatismos.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

De esta reunión surgieron los principales controladores iniciales de la arquitectura los cuales corresponden a las metas arquitectónicas iniciales:

- **Desempeño:** corto tiempo de respuesta en la validación de las tarjetas de los pasajeros al momento de subirse al bus o descontar el monto del taxi.
- **Escalabilidad:** el sistema puede crecer de acuerdo a las proyecciones estimadas para el uso del sistema de transporte, así como puede requerirse nuevas dimensiones de usuarios, unidades de transporte y datos si este es instalado en una ciudad más grande en otro país o si es integrado con otros servicios de transporte.
- **Tolerancia a fallos:** el sistema deberá ser capaz de tolerar problemas de conectividad, procesamiento con grandes cargas y bajas del sistema sin daños que perjudiquen la cadena de valor del Transmontevideo.
- **Modificabilidad/Reuso:** el sistema deberá ser capaz de soportar cambios debido a la evolución del sistema de transporte y de futuros y muy probables cambios en la lógica del negocio (en especial la forma de pago). Igualmente, se espera poderlo modificar/reusar en otros sistemas de transporte.

Desde la perspectiva del negocio existen restricciones de tiempo y costos que no están definidos y que seguramente afectará las decisiones arquitectónicas. Para efectos de este proyecto, se presentan las siguientes restricciones (bajo el supuesto del equipo):

El proyecto (software) ha sido adjudicado a nuestra empresa de desarrollo de software y la gerencia le ha asignado un presupuesto al proyecto. Dicho presupuesto logra cubrir \$900 millones de pesos chilenos y está estimado para ser terminado en un año y medio (incluyendo la implantación). Para el proyecto se han asignado montos para los siguientes rubros:

- Personal: 24 personas (administradores, ingenieros, arquitectos, analistas, testers, etc.).
- Capacitación: 50 millones de pesos
- Software (componentes software si se requiriesen): 50 millones – compensables con personal (como colchón presupuestal).

Debido a que el plan está impulsado por la Intendencia Municipal de Montevideo (IMM) y el Ministerio de Transporte y Obras Públicas (MTO), con apoyo del Gobierno de la República, en el presente o en un futuro cercano, SGT deberá interactuar con los sistemas de estos organismos para propósitos regulatorios y de supervisión, para lo cual el SGT debe suministrar una o varias interfaces. Para garantizar la compatibilidad dichas interfaces podrán ser manejadas a nivel de datos con un formato definido o a través de un web service.

La empresa desarrolladora no cuenta con proyectos similares en el mismo dominio, y dado que TransMontevideo es un sistema completamente nuevo podemos hacer las

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

siguientes consideraciones:

- No existen componentes reutilizables (candidatos) para ser incluidos en el proyecto, tampoco una Arquitectura de Referencia para el proyecto (al interior de la organización).
- No existen sistemas legados (en el contexto del negocio objetivo) a tener en cuenta ya que el sistema software empieza desde cero con el sistema de transporte.

Además del monto contratado al desarrollo del proyecto software (el cuál es diferente al costo del proyecto definido en la empresa desarrolladora), está el monto que TransMontevideo le asignó a la infraestructura tecnológica, para esto TransMontevideo cuenta con el presupuesto para:

- Compra de 20 Servidores de alto desempeño con altas capacidades (acordes a la información estimada a ser manejada) de almacenamiento.
- 100 terminales de venta.
- 50 terminales de operaciones y administración.
- 10000 Validadores (Buses y taxis) y GPS
- Configuración de una red de área metropolitana (MAN) de alto (de acuerdo a las velocidades estimadas requeridas) ancho de banda para las comunicaciones internas a la red TransMontevideo, así como de tres redes LAN para algunas dependencias.
- Licenciamiento del software respectivo (aún por aclarar)

4. Vista de Casos de Uso y Escenarios de Calidad

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad al aplicar QAW. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

4.1 Modelo de Casos de Uso

El modelo de casos de uso de SGT es presentado en la Figura 3. En él se pueden visualizar los actores principales del sistema, así como los principales usos que estos le darán al sistema.

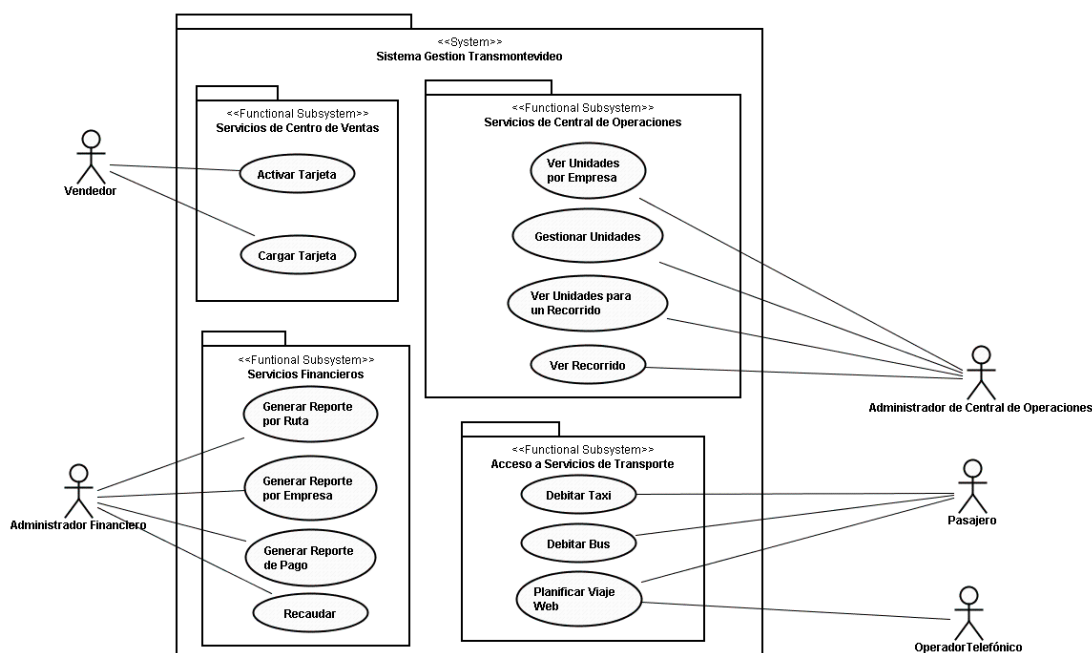


Figura 3. Modelo de Casos de Uso

Los actores principales del sistema se describen a continuación:

- **Pasajero:** persona que actúa de cliente de los servicios de TransMontevideo.
- **Vendedor:** es la persona encargada de atender a los usuarios del TransMontevideo en los centros de venta. Su función en el negocio es a la llegada de un usuario vender una tarjeta o cargar con dinero una previamente vendida.
- **Operador Telefónico:** es la persona encargada de atender a los pasajeros cuando estos requieren ayuda vía teléfono para la planificación de sus viajes o presencialmente solicitan una información cuando el operador se encuentra en una estación de información.
- **Administrador de Central de Operaciones:** es la persona encargada de soportar los procesos de monitoreo y control del sistema de buses y taxis del TransMontevideo.
- **Administrador Financiero:** es la persona encargada de las actividades administrativas del área financiera de TransMontevideo. Este actor deberá registrar las recaudaciones diarias por cada empresa, administrar los pagos a los contratos de las empresas de acuerdo a las recaudaciones logradas en los períodos de tiempo respectivo y realizar las transacciones financieras correspondientes a dichos pagos.

El modelo de casos de uso se describe de manera general partiendo de sus procesos de negocio de la siguiente forma:

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Los *pasajeros* de TransMontevideo usan los servicios del TransMontevideo. Cuando abordan la unidad de transporte el pasajero ***debita su pasaje*** pasando su tarjeta por el verificador. Este débito depende si el viaje lo realiza en taxi (***debitar pasaje taxi***) o si se realiza en un bus (***debitar pasaje bus***). Cuando su tarjeta se queda sin saldo, el *pasajero* debe recargar su tarjeta con algún valor que él lo decida, en este caso, el pasajero lo solicita al *vendedor* y el *vendedor* le solicita al sistema ***cargar tarjeta***. Si el *pasajero* no cuenta con una tarjeta deberá comprarla, en este caso el *vendedor* debe solicitarle al sistema ***activar tarjeta*** con el fin de entregarle una tarjeta activada. En algunas ocasiones el *pasajero* puede requerir planificar su viaje y para ello usará el servicio ***planificar viaje web*** ofrecido por el sistema a través de la web. Si el *pasajero* no tiene acceso a la web, este puede realizar una llamada y un *operador telefónico* le guiará en el proceso, en este caso el *operador telefónico* requerirá iniciar el caso de uso ***planificar viaje web***.

El *administrador de operaciones* para soportar los servicios de monitoreo y gestión puede requerir observar información del estado de las unidades de transportes en un instante determinado. Para ello requiere de los casos de uso: ***ver recorrido*** para visualizar la información asociada a un recorrido (paraderos y puntos de congestión), ***ver las unidades para un recorrido*** (las unidades de transporte, sus distancias, pasajeros), ***ver las unidades por empresa*** (para verificar el cumplimiento por parte de las empresas proveedoras de servicio de transporte) y ***gestionar unidades*** para gestionar las unidades de transporte (crear, modificar, activar/desactivar, agregar nuevas variables de monitoreo, monto cobrado por el servicio). A excepción de la gestión de unidades, los casos de uso descritos en este último párrafo toman como información de monitoreo, la última información enviada por las unidades, las cuales de manera periódica están notificando su estado.

El administrador financiero por su parte debe configurar el proceso de recaudo a través del caso de uso recaudar, así como actualizar contratos, y generar los reportes financieros correspondientes a rutas, empresas y el pago las mismas, para esto ofrece los casos de uso: ***recaudar***, ***administrar contrato***, ***generar reporte por ruta***, ***generar reporte por empresa*** y ***generar reporte de pago*** respectivamente.

4.2 Especificación de los Casos de Uso Relevantes

A través de la reunión donde se aplicó QAW, los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados. Los criterios usados para dicha determinación fueron:

- Su implementación implica varios nodos de la vista de deployment (previamente bosquejada).
- Su implementación es de alto riesgo.
- Incluye muchos conceptos y relaciones del dominio.
- Incluye posibles escenarios críticos de calidad.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

A continuación se describen los casos de uso seleccionados:

ID	CU1
Nombre	Acceso a Servicios de Transporte :: Debitar Bus
Actores	<u>Pasajero</u>
Sinopsis	Se inicia cuando el pasajero sube a un bus y debe cancelar su pasaje. O al hacerlo en un terminal de acceso al bus.
Curso Típico de Eventos	
	1. El pasajero acerca su tarjeta BIP! al validador de tarjetas del bus. 2. El sistema comprueba validez de la tarjeta 3. El sistema descuenta del saldo de la tarjeta BIP! del pasajero, el valor del pasaje, aún cuando el saldo de la tarjeta quede negativo (siendo positivo previamente al pago) y da acceso al uso del bus.
Extensiones	
	3a. Si la tarjeta es inválida, el sistema no da acceso al uso del bus o paso por el terminal de acceso.
Prioridad AQ	Alta

ID	CU2
Nombre	Servicios de Central de Operaciones :: Ver Unidades para un recorrido
Actores	<u>Administrador de la Central de Operaciones</u>
Sinopsis	El Administrador de la central de operaciones desea conocer la localización actual de cada una de las unidades para un recorrido (ruta).
Curso Típico de Eventos	
	1. El administrador selecciona el recorrido del mapa de recorridos. 2. El sistema recupera el último refresco de las posiciones de los buses activos en el recorrido seleccionado. 3. El sistema despliega posiciones y dirección de los buses.
Extensiones	
	Ninguna.
Prioridad AQ	Alta

ID	CU3
Nombre	Información al usuario :: Planificar Viaje
Actores	<u>Pasajero</u> , <u>Operador Telefónico</u>
Sinopsis	Se inicia cuando el pasajero desea planificar su viaje, para lo cual dispone de acceso por Internet o por teléfono.
Curso Típico de Eventos	
	1. El pasajero entra a la página Web de TransMontevideo.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

	2. El pasajero se dirige a la sección de origen y destino. 3. El pasajero ingresa la dirección de origen y de destino. 4. El sistema procesa los datos y le muestra las rutas disponibles. 5. El pasajero solicita visualizar ruta 6. El pasajero selecciona ver información de un paradero 7. El sistema le muestra la posición del bus más próximo
Extensiones	
	1a. El pasajero no tiene acceso a Internet: 1. El pasajero debe llamar a la Central de Información Telefónica. 2. Un operador telefónico recibe su llamada. 3. El pasajero le indica su dirección de origen y de destino. 4. El operador telefónico consulta en la página Web, ingresando los datos indicados por el pasajero. 5. El operador indica las rutas disponibles al pasajero. 5.a El pasajero no sigue planificando el viaje en más detalle.
Prioridad AQ	Alta

ID	CU4
Nombre	Servicios Financieros::Recaudar
Actores	<u>Administrador financiero</u>
Sinopsis	El administrador financiero ingresará al sistema para definir la forma en que se hará el recaudo de acuerdo a las políticas definidas para el TransMontevideo. El sistema registra la información de recaudo (empresas activas y frecuencia de recaudo) y de acuerdo a esta información realiza de manera periódica el recaudo. El recaudo corresponde al dinero descontado de las tarjetas en el período especificado y para cada empresa prestadora de servicio de transporte (obtenido de sus unidades al hacer los pagos).
Curso Típico de Eventos	
	1. El administrador financiero solicita la definición de la forma de recaudo. 2. El sistema muestra la información de recaudo (periodicidad, empresas de servicio). 3. El administrador financiero actualiza la información de recaudo. 4. El sistema valida y acepta los nuevos datos. 5. El sistema periódicamente hace el recaudo de acuerdo a los parámetros establecidos.
Extensiones	
	4a. El sistema valida y rechaza la solicitud y vuelve al paso 2.

Prioridad AQ	Alta
ID	CU5
Nombre	Gestión Financiera::Generar Informe Diario por Recorrido
Actores	Administrador Financiero
Sinopsis	El administrador financiero solicita generar el informe diario para todos los recorridos.
Curso Típico de Eventos	
	1. El administrador financiero solicita informe diario. 2. El sistema realiza la consulta y recupera la información. 3. El sistema estructura la información del uso de acuerdo al formato del informe diario de recorrido. 4. El sistema despliega el informe en pantalla. 5. El sistema despliega el informe en pantalla según el uso de cada uno de los recorridos existentes.
Extensiones	
	Ninguna.
Prioridad AQ	Baja

4.3 Especificación de los Escenarios de calidad Relevantes

QAW nos permitió a partir de la asignación de roles de los principales interesados (stakeholders influenciadores de la arquitectura), una negociación de los principales atributos de calidad, primero a través de la identificación, priorización y descripción. Determinar las mediciones en los respectivos escenarios permite visualizar que atributos pueden estar comprometidos y como pueden tenerse en cuenta como drivers arquitectónicos para definir las estrategias y estilos arquitectónicos que permitan resolverlos más adelante. Los escenarios de calidad se expresan a continuación.

ID: QS1

Nombre: Seguridad :: Seguridad en las tarjetas

Sinopsis: Directivas de seguridad para detectar tarjetas falsas o adulteradas.

Entorno: Las tarjetas y validadores están funcionando correctamente.

Cambio en el entorno: Una tarjeta falsa o adulterada es acercada al validador.

Comportamiento esperado: El validador debe poder detectar el intento de adulterio, registrar la fecha y el lugar en que ocurrió y notificar al chofer y a la central.

Medida: Ninguna tarjeta inválida o adulterada puede aprobar la prueba del validador.

Prioridad Arquitectónica: Alta

Aplicación: Local

ID: QS2

Nombre: Desempeño:: Tiempo de respuesta en la lectura de tarjeta

Sinopsis: Al pasar la tarjeta sobre el validador se necesita obtener la respuesta de acceso en un corto tiempo.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Entorno: La tarjeta y el validador funcionando adecuadamente.

Cambio: Paso de la tarjeta por el validador.

Comportamiento: El validador responde permitiendo el acceso y descontando el importe o indicando que no se cuenta con el cambio suficiente.

Medida: La espera del pasajero debe ser inferior a 2 segundos.

Prioridad Arquitectónica: Alta

Aplicación: Local

ID: QS3

Nombre: Tolerancia a fallos :: Recuperación en un fallo al momento de la recaudación de las ventas.

Sinopsis: En algún momento este proceso es interrumpido sin aún haber alcanzado la totalidad de la transferencia.

Entorno: El sistema trabajando en su carga normal

Cambio en el entorno: Interrupción de la transferencia de la recaudación.

Comportamiento esperado: El sistema intenta identificar el momento más cercano a la interrupción para reanudar la transferencia desde dicho punto.

Medida: Se debe poder asegurar un 95% en la recuperación de los datos transferidos, el 5% podría ser solicitado como reenvío.

Prioridad Arquitectónica: Alta

Aplicación: Parcialmente local

ID: QS4

Nombre: Seguridad :: Denegar acceso a usuarios sin los privilegios respectivos.

Sinopsis: Características que deben tener los sistemas para prohibir el acceso a usuarios ajenos al mismo.

Entorno: Los sistemas de información y de la central de operaciones están instalados y funcionando.

Cambio en el entorno: Un usuario malintencionado intenta acceder fraudulentamente al sistema.

Comportamiento esperado: El sistema debe detectar los intentos de accesos indebidos y bloquearlos de forma inmediata y notificar a los administradores que hubo un intento de entrar al sistema de mala forma.

Medida: Un acceso no permitido en el sistema en diez millones de intentos de violación.

Prioridad Arquitectónica: Media

Aplicación: Localizado

ID: QS5

Nombre: Modificabilidad(o Reuso)::Cambio en la lógica de cobro del servicio.

Sinopsis: La organización desarrolladora piensa proveer la misma solución a otras empresas municipales dentro y fuera de Uruguay. Es necesario que la misma solución, en el mismo contexto soporte cambios en su lógica del negocio (p.e. lógica de cobro).

Entorno: El sistema en Desarrollo/Evolución.

Cambio en el entorno: se requiere cambiar el modelo de cobro del servicio de prepago a pospago (consumo mensual le llega como una factura más del servicio).

Comportamiento esperado: Cambios son realizados con éxito.

Medida: Los cambios son realizados en un tiempo inferior a un mes con un esfuerzo máximo de 2 personas mes.

Prioridad Arquitectónica: Alta

Aplicación: Global

ID: QS6

Nombre: Escalabilidad:: Crecimiento del Sistema de Transportes TransMontevideo

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Descripción: Debido a que el número de clientes de TransMontevideo crecerá en el futuro, y al mismo requerimiento de negocio de ofrecer el sistema hacia otras ciudades, en especial se tiene como objetivo de negocio ciudades aún más grandes y/o con capitales financieros mayores que implique un sistema de transporte mucho más grande en buses, rutas y paraderos y por lo tanto en usuarios y en información.

Entorno: Deployment

Cambio en el entorno: Se requiere reconfigurar el sistema para soportar una cantidad adicional de pasajeros del 100%

Comportamiento esperado: El mismo sistema en una nueva configuración de deployment.

Medida: Los cambios son realizados en un tiempo inferior a una semana con un esfuerzo máximo de 4 personas mes.

Prioridad Arquitectónica: Alta

Aplicación: Global

5. Vista Lógica

Para el desarrollo de ésta vista se usó el ADD [Woj06], de acuerdo a este método los drivers de la arquitectura son identificados, así como las estrategias arquitectónicas asociadas. Esta información es presentada en el ítem de Justificación (Rationale) de la Arquitectura. De acuerdo al Rationale, para esta vista se tuvieron en cuenta los patrones de descomposición, generalización y uso del tipo de estilo módulo. Así mismo se bosquejó la aplicación del tipo de estilo C&C teniendo en cuenta el patrón del Pizarrón (el cuál será de mayor utilidad en la vista de procesos).

5.1 Parte Estructural: descomposición modular

5.1.1 Primer Nivel de Refinamiento

Para un primer entendimiento del sistema se presentará un primer nivel de refinamiento el cual es presentado en la figura 4. Este modelo está basado en la descomposición funcional y la distribución física expuesta en la vista de Deployment. En este modelo se visualiza la aparición en forma bosquejada de un estilo basado en descomposición, uso y generalización.

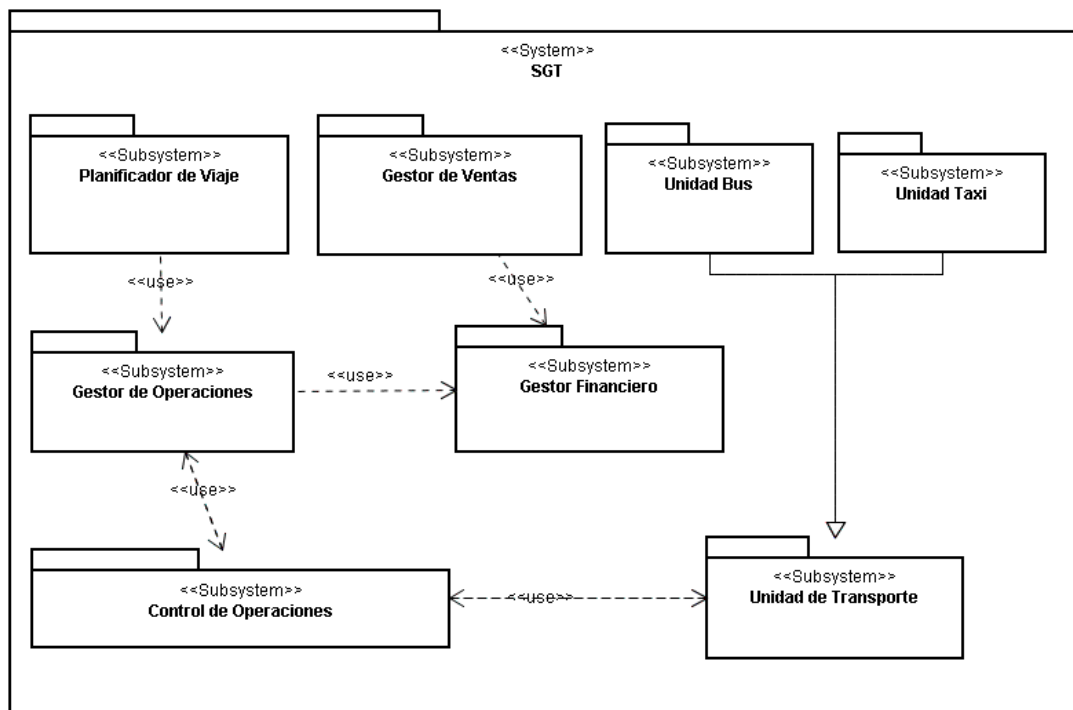


Figura 4. Primer Nivel de Refinamiento

Una descripción de los subsistemas presentados en la figura 4 se presenta en la tabla 1 junto con la información heredada de la vista de casos de uso.

Módulo	Descripción	Atributos/Escenarios De Calidad	Casos de uso
Planificador de Viaje	Ayuda a estructurar los viajes con información veraz y actualizada.	QS5, QS6	CU3
Gestor de Operaciones	Administra la información de las operaciones y notifica de los cambios a los subsistemas interesados.	QS5, QS6	CU2
Control de Operaciones	Coordina la información entre las unidades y el gestor de operaciones.	QS5, QS6	CU2, CU3, CU4
Gestor Financiero	Recopila la información financiera y facilita la construcción de reportes financiero.	QS3, QS5, QS6	CU4, CU5
Unidad de Transporte	Coordina la operación de una unidad de transporte (p.e. bus)	QS1, QS2, QS5	CU1
Gestor de Ventas	Soporta los procesos de cargado y activación de tarjetas, así como el envío de ventas consolidadas al subsistema financiero.	QS3, QS5, QS6	

Tabla 1. Localización de requisitos en los subsistemas respectivos.

* QS4 no fue considerado como driver (Ver Rationale)

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

5.1.2 Segundo Nivel de Refinamiento

Para el segundo nivel de refinamiento cada uno de los subsistemas del modelo anterior es descompuesto recursivamente.

5.1.2.1 Planificador de Viaje

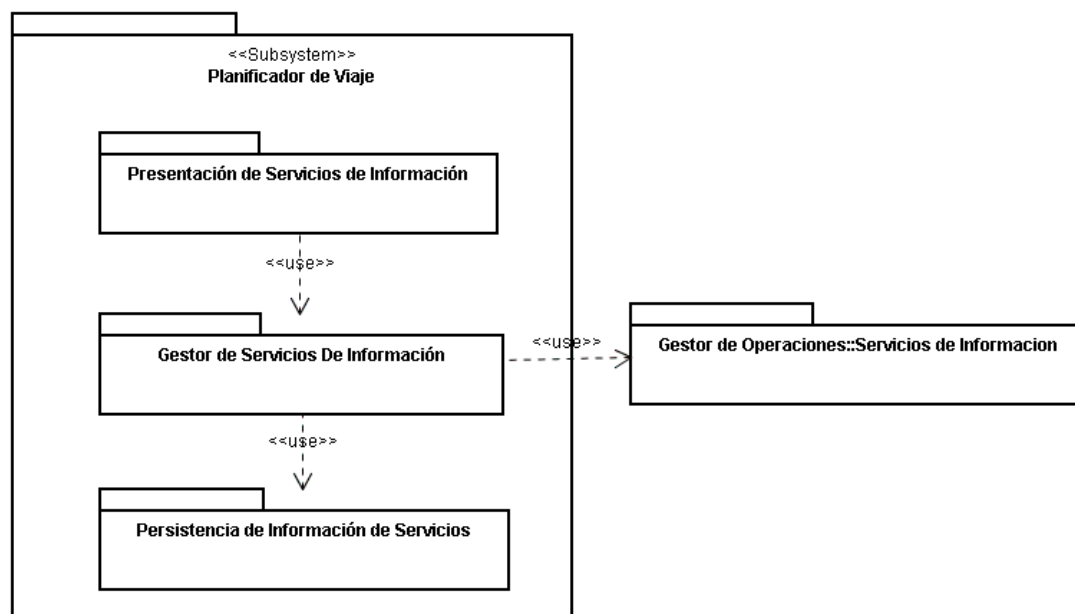


Figura 5. Descomposición del Subsistema Planificador de Viaje

Este subsistema ha sido descompuesto, de acuerdo a la Figura 5 siguiendo un estilo de capas. Las capas identificadas son:

- **Capa de Persistencia de Información de Servicios:** Esta capa es la encargada de mantener y manejar toda la información sobre los distintos recorridos necesaria para poder planificar un viaje, esto incluye información cartográfica. Esta información incluye información estática de los recorridos así como información frecuentemente actualizada de las unidades de transporte y rutas.
- **Capa de Gestión de Servicios de Información:** Esta capa tiene la responsabilidad de administrar las peticiones de los usuarios (Pasajeros, Operadores Telefónicos), así como la frecuente comunicación con el subsistema Gestor de Operaciones para mantener la información dinámica actualizada de unidades de transporte y rutas.
- **Capa de Presentación de Servicios de Información:** Esta capa se encarga de presentar los servicios disponibles para la planificación del viaje, así como la visualización de los mapas.

5.1.2.2 Gestor de Ventas

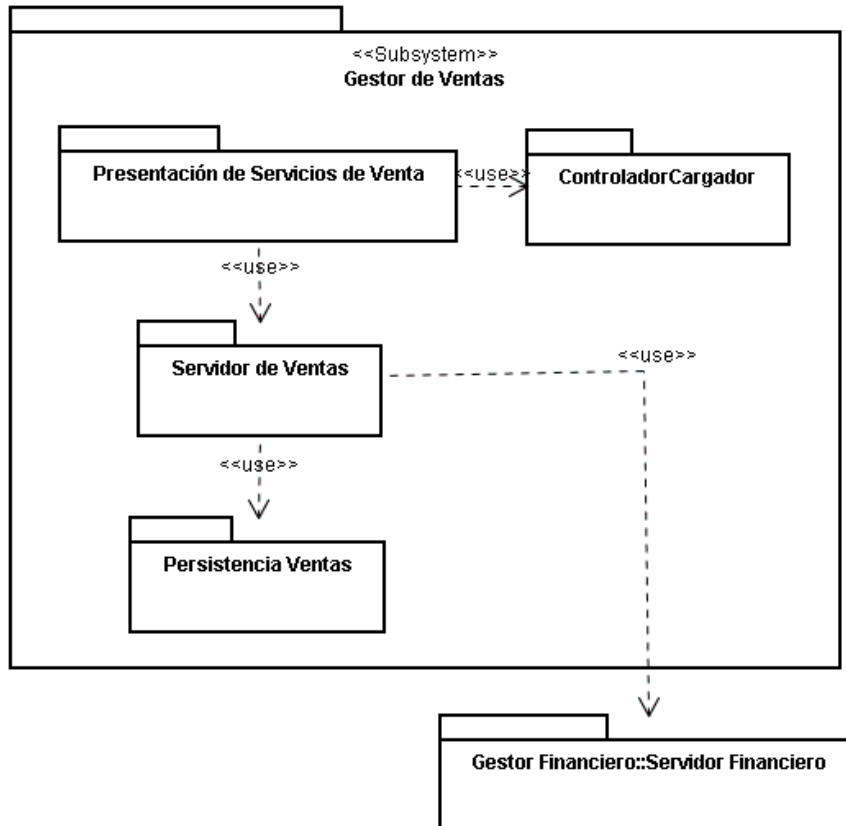


Figura 6. Descomposición del Subsistema Gestor de Ventas

Este subsistema ha sido descompuesto siguiendo los estilos de descomposición, uso y capas. Los módulos identificados son:

- **Capa de Persistencia de Ventas:** Se encarga de almacenar la información referente a las ventas y recargas de tarjetas.
- **Capa Servidor de Ventas:** Se encarga de coordinar la información de todos los centros de ventas y posteriormente de la transferencia de estos al Gestor Financiero, y también al Módulo de Persistencia.
- **Capa de Presentación de Servicios de Ventas:** Son los encargados de manejar la interacción con el usuario (vendedor).
- **Módulo Controlador de Cargador:** Es el encargado de manejar el dispositivo de cargador en el proceso de activación y cargado de saldo de tarjetas.

5.1.2.3 Gestor de operaciones

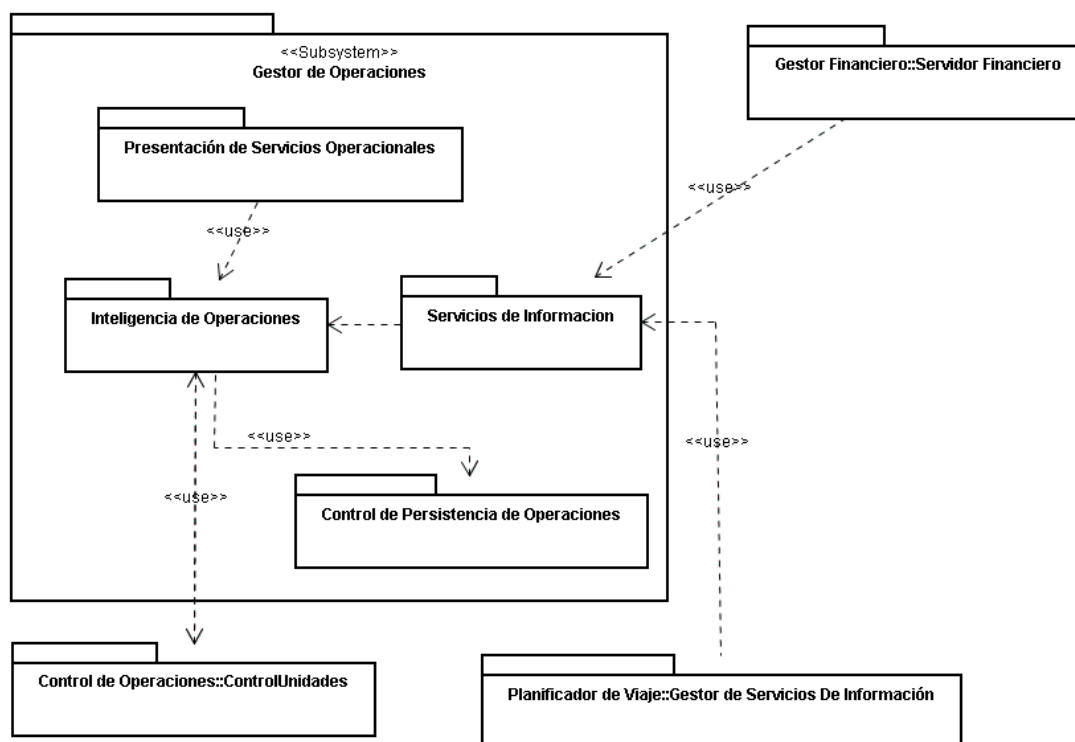


Figura 7. Descomposición del Subsistema Gestor de Operaciones

Este subsistema ha sido descompuesto siguiendo los estilos de descomposición, uso y capas, así como el soporte para que todo el subsistema actúe como un elemento coordinador de la lógica operacional del SGT.

Los módulos identificados son:

- **Capa de Presentación de Servicios Operacionales:** se encarga de presentar de manera visual la información de la operación real del sistema de transporte con el fin de la toma de decisiones en la operación diaria. Esto incluye el estado de las rutas, la demanda de pasajeros, así como el monitoreo de las unidades en el contexto de los recorridos o en su interior (video, número pasajeros) si es necesario.
- **Capa de Inteligencia de Operaciones y Servicios de información:** se encarga de coordinar el pizarrón. Esta capa se divide en dos módulos como alternativa de separación de preocupaciones. La inteligencia de operaciones se encarga de recibir y notificar información (operaciones del Transmontevideo y órdenes provenientes del gestor de operaciones) al subsistema de control de operaciones. El módulo de servicios de información se encarga de recibir y notificar cambios en la información al subsistema planificador de Viaje, así como al Gestor Financiero. Este último módulo es más un helper con un buffer de almacenamiento intermedio para no saturar el módulo de inteligencia de operaciones.

- **Capa de Persistencia:** permite almacenar el estado definitivo de las operaciones diarias, esto incluye consumos, recorridos, órdenes de control realizadas por los operadores y un Log del subsistema.

5.1.2.4 Control de operaciones

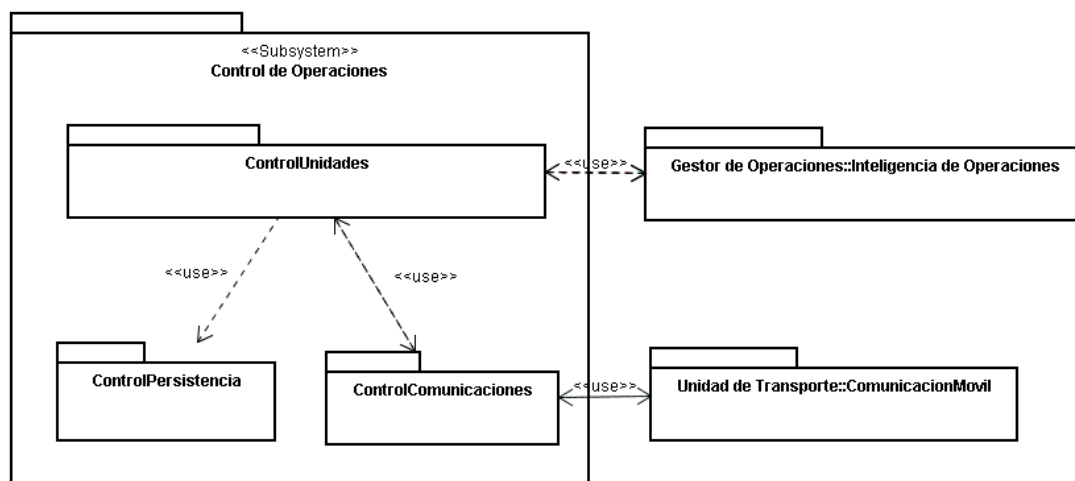


Figura 8. Descomposición del Subsistema Control de Operaciones

Este subsistema ha sido descompuesto usando los estilos de descomposición y uso. A continuación se describen los módulos que lo componen:

- **Módulo de Control de Unidades:** Tiene como responsabilidad el monitoreo de las unidades para controlar su frecuencia y su ruta dada alguna eventualidad. Además se debe poder activar la grabación de la cámara de seguridad del bus y también se debe poder observar los registros de dicha cámara, siempre y cuando la comunicación lo permita. Por último se encarga también de enviar la información recopilada de las operaciones al Gestor de Operaciones.
- **Módulo de Control de Persistencia:** Debido al intensivo procesamiento de este subsistema, esto se usa para almacenar temporalmente las operaciones registradas por el Control de Unidades, para que éste las envíe al Gestor de Operaciones cuando se disponga de capacidad de procesamiento ello.
- **Módulo de Control de Comunicaciones:** Es el encargado de comunicarse con las unidades de transporte, es decir se encarga de recibir la información que éstas mandan (transacciones, posición o información multimedia) para enviarla al Control de Unidades, y de enviar mensajes a las unidades cuando se les debe informar de alguna eventualidad, como desvíos o choques o que a otra unidad de Transporte se le quemó la batería.

5.1.2.5 Gestor financiero

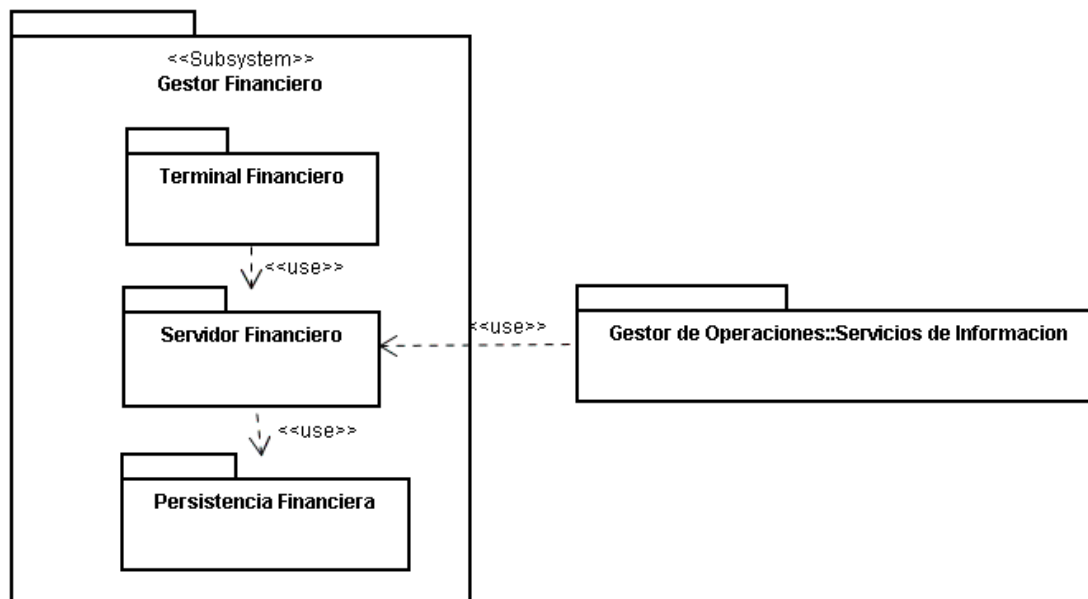


Figura 9. Descomposición del Subsistema Gestor Financiero

- **Persistencia Financiera:** Almacena la información usada por el subsistema. En particular se encarga de almacenar los datos referentes a las ventas y cargas de todos los centros. Además guarda la información recibida por el Gestor de Operaciones, la cual consiste en los distintos usos del servicio de transportes.
- **Servidor Financiero:** Se encarga de recibir la información mandada por el Gestor de Operaciones, referente al uso del sistema por parte de los usuarios. Además se preocupa de enviar, al módulo de Persistencia Financiera, los datos que serán almacenados en este subsistema.
- **Terminal Financiero:** Se utiliza para presentar al usuario la información referente a las ventas y cargas de tarjetas, además de el las finanzas sobre el uso del servicio transporte.

5.1.2.6 Unidad de Transporte

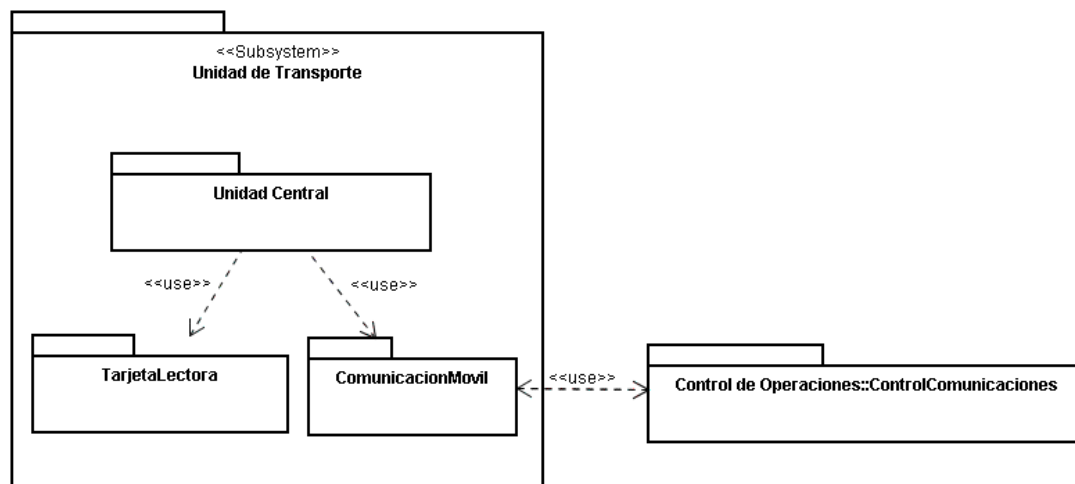


Figura 10. Descomposición de la Unidad de Transporte

- **ComunicacionMovil:** Este módulo se encarga de transferir toda la información capturada por el bus.
- **Unidad Central:** Módulo encargado de la administración de la información capturada por los distintos dispositivos dentro del bus o taxi. En particular, decide cuándo es necesario enviar la información al Control de Operaciones, utilizando el módulo de ComunicacionMovil.
- **TarjetaLectora:** Se preocupa de la comunicación con la tarjeta BIP! y así capturar los datos referentes al uso del servicio de transporte.

5.1.2.7 Unidad de Bus

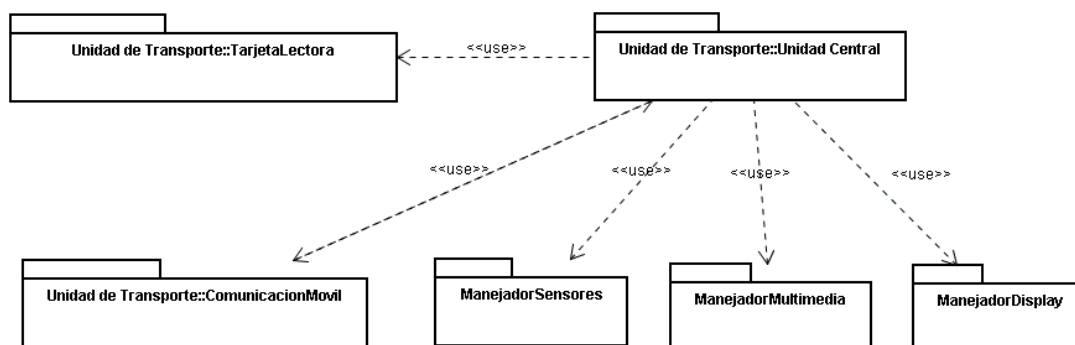


Figura 11. Descomposición de la Unidad de Bus

- **ManejadorSensores:** Se encarga de administrar todos los datos captados por los sensores al interior del bus. En específico se cuenta con los sensores de

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

puertas.

- **ManejadorMultimedia:** Se encarga de la comunicación con los dispositivos que captan datos de video y sonido al interior del bus.
- **ManejadorDisplay:** Despliega información útil para el conductor, manejada por el área de operaciones.

5.1.2.8 Unidad de Taxi

Ídem Unidad de Transporte.

5.2 Parte Dinámica (Comportamiento)

La parte dinámica (behavior) de la vista lógica es modelada a través del view type C&C, siguiendo la notación brindada en la guía y complementada con el trabajo presentado por [Roh04]. De acuerdo a esta perspectiva y de la aplicación del método ADD con los drivers identificados los patrones que satisfacen la arquitectura y que pueden ser aplicados en conjunto son los siguientes:

- En la primera descomposición
 - ✓ Los componentes principales del negocio se estructuran a través de un esquema **Publicar/Suscribir**. Todos los componentes interactúan con un rol de publicador/subscriptor. El componente de negocio concentra la información de gestión general para soportar todo el negocio y actúa como repositorio activo (sin embargo la interacción corresponde más al patrón publicador subscriptor que al de un pizarrón), recibiendo los cambios y actualizaciones de las otras componentes y notificando los cambios a las componentes interesadas de los cambios. La aplicación de este estilo permite separar completamente la lógica del negocio de Gestión Financiera, la información de planificación de viaje y la información necesaria para el control de operaciones. Así favorece la modificabilidad en estos componentes y el crecimiento en cuanto al soporte de nuevas unidades de negocio o nuevas instancias de las unidades existentes.
 - ✓ El Control de Operaciones y las unidades de transportes conforman una dinámica **Publicar/Subscribir**, debido a que en su mayor parte se controlan mediante eventos (de ida y regreso, y de manera síncrona y asíncrona), esto le da soporte a la escalabilidad (el número de unidades puede crecer y del otro lado quien atiende las solicitudes puede crecer de manera transparente para poder soportar el crecimiento de la carga) y finalmente, las componentes de las terminales se independizan del control de operaciones brindando la flexibilidad necesaria para ganar terreno en la modificabilidad.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

- ✓ La interacción entre la unidad de negocio de Ventas, quien actúa como cliente de la Unidad Financiera, hacen que esta interacción sea de tipo solicitud/respuesta de manera asimétrica identificándose bien al módulo de ventas como un cliente del módulo de gestión financiera como servidor, llevando a un patrón de interacción **Cliente/Servidor**.
- Para la segunda descomposición:
 - ✓ En general cada subsistema identificado incluye diferentes niveles de funcionalidad que será desarrollado, empaquetado e instalado en diferentes módulos, componentes y nodos. Estos niveles que en la vista estructural corresponden a capas en la vista dinámica coinciden, en este caso, con una dinámica de **n-tiers**. Esta coincidencia radica en que precisamente los sistemas de información típicos presentan esta singular separación que afecta tanto la vista modular como la de Deployment, hecho que generalmente se presta para confusiones entre los dos estilos arquitectónicos.
 - ✓ Algunos subsistemas identifican componentes que interactúan a través de mecanismos petición/respuesta de manera simétrica llevando a usar para su descomposición el patrón **peer to peer**.
 - ✓ La unidad de transporte sigue el patrón **Publicador/Subscriber** para unidad de transporte, en donde el componente central y los componentes drivers de los dispositivos actúan como publicadores y subscribers dependiendo del componente.

Para organizar la descripción, en el primer nivel de refinamiento se describen dos partes relevantes en los view packet Unidades de Negocio Principales y en la relación entre las unidades de Venta y Gestión financiera. Para el segundo nivel sólo fue necesario agregar el view packet Unidad de Transporte debido a que agregaba información adicional que resultó relevante. Los componentes de cada view packet del primer nivel de refinamiento fueron refinados siguiendo un patrón de descomposición y el patrón que resultó del rationale localizado.

5.2.1 View packet Unidades de Negocio Principales

5.2.1.1 Vista de Componentes y Conectores

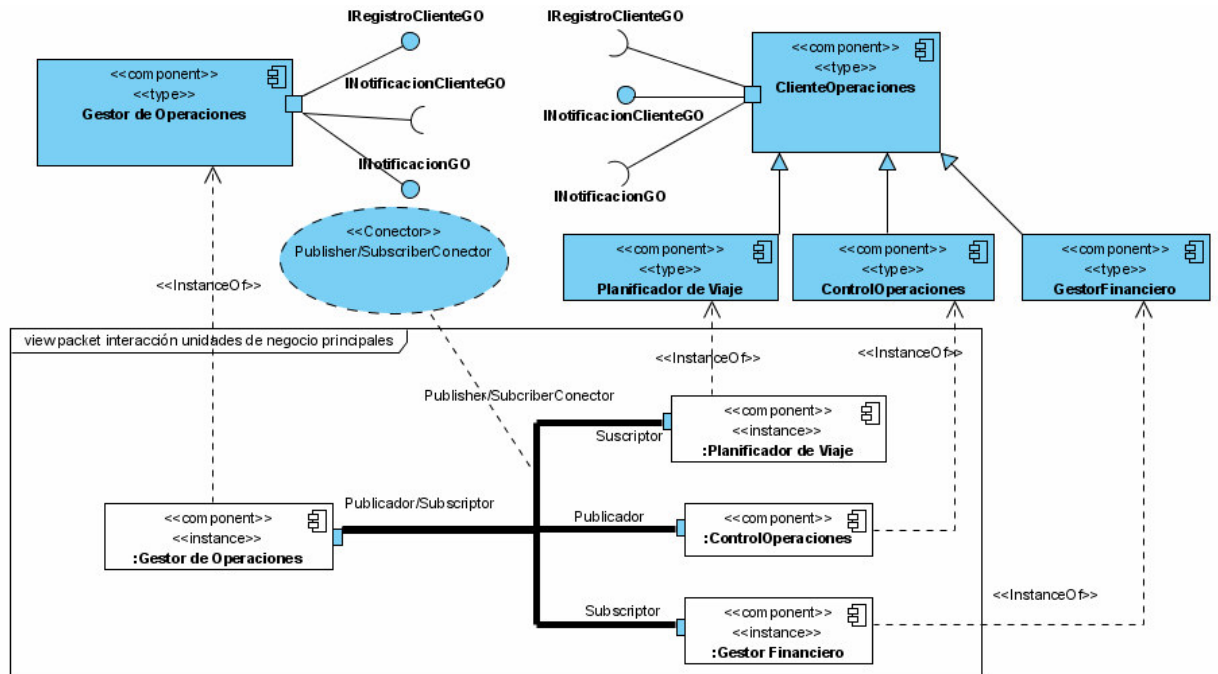


Figura 12. C&C View para el View Packet Pizarrón

5.2.1.2 Realización de la interacción del conector Publisher/SubscriberConector

Con el fin de conocer la dinámica de interacción que brinda un conector, asociado a este se ha definido la realización de dicha interacción, haciendo uso de los roles participantes del conector, es posible mediante un diagrama de secuencia o de colaboración delinear la forma en que los participantes interactuarían de manera dinámica. La Figura 13 presenta la dinámica del publicador suscriptor.

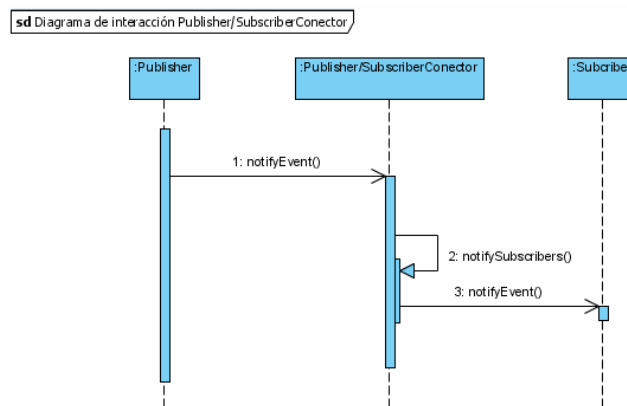
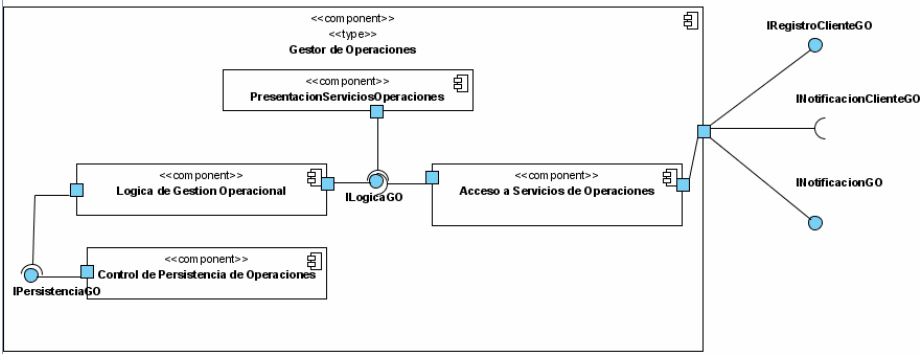
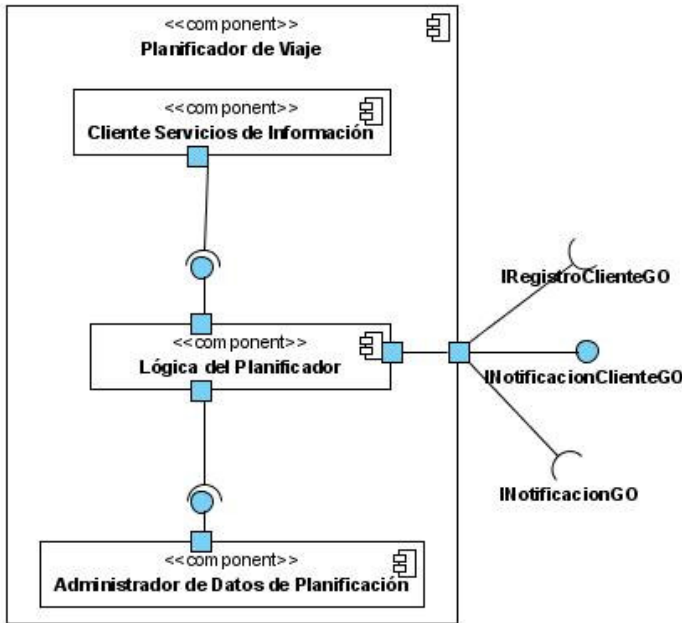
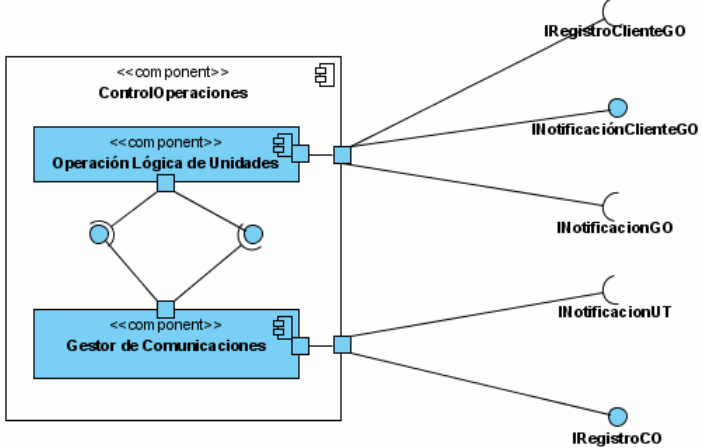
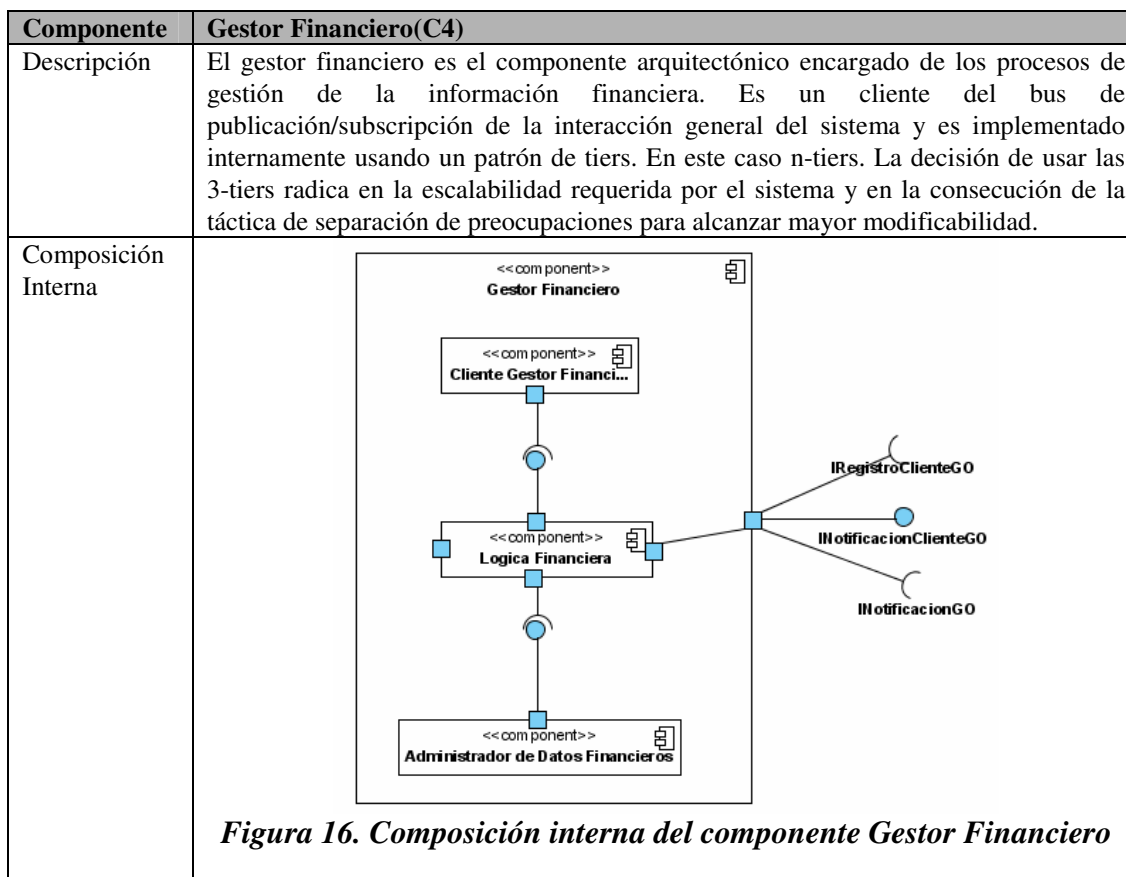


Figura 13. Realización de la interacción del conector Publisher/Suscriber Conector

5.2.1.3 Catálogo de componentes

Nombre del View Packet		Unidades de Negocio principales
Rationale		El patrón publicador suscriptor permite alcanzar los atributos de modificabilidad y escalabilidad. A través de este patrón de interacción definido, la lógica de negocio repartida entre las diferentes componentes se aísla y cualquier unidad nueva de acción puede ser incluida en el futuro. La lógica de interacción entre las unidades de negocio y la información compartida entre los componentes pueden ser modificadas, puesto que el patrón de interacción se ha dejado lo más genérico posible.
Catálogo de Componentes		
Componente:	Gestor de Operaciones (C1)	
Descripción	Es el principal cliente del bus de publicación suscripción (inicialmente pensado como un pizarrón) y por tanto facilita el acceso a la información de la gestión de operaciones presentada en el view type de módulos. Internamente este componente sigue un patrón cliente-servidor.	
Composición interna	 Figura 14. Composición interna del componente Gestor de Operaciones	
Componente:	Planificador de Viaje(C2)	
Descripción	El planificador de viaje es la componente arquitectónica encargada de prestar los servicios de recuperación de información estática y dinámica para la planificación de los viajes.	

Composición Interna	 The diagram shows the internal composition of the 'Planificador de Viaje' component. It contains three sub-components: 'Cliente Servicios de Información', 'Lógica del Planificador', and 'Administrador de Datos de Planificación'. 'Cliente Servicios de Información' is connected to 'Lógica del Planificador' via a required interface (half-circle) and a provided interface (square). 'Lógica del Planificador' is connected to 'Administrador de Datos de Planificación' via a required interface (half-circle) and a provided interface (square). Additionally, 'Lógica del Planificador' is connected to three external interfaces: 'IRegistroClienteGO', 'INotificacionClienteGO', and 'INotificacionGO'. Figura 15. Composición interna del componente Planificador de Viaje
Componente	Control de Operaciones(C3)
Descripción	Control de Operaciones, que debe suscribirse a los eventos enviados por las unidades de transporte y debe publicar las órdenes de modificaciones de rutas y/o velocidad a cada unidad. De igualmente actúa como publicador/subscriber de eventos de actualización con respecto a eventos de actualización de información en el gestor de operaciones. Internamente utiliza el patrón peer to peer para establecer una colaboración bidireccional entre la lógica de comunicación y la lógica de control de operaciones como tal.
Composición Interna	 The diagram shows the internal composition of the 'ControlOperaciones' component. It contains two sub-components: 'Operación Lógica de Unidades' and 'Gestor de Comunicaciones'. 'Operación Lógica de Unidades' is connected to 'Gestor de Comunicaciones' via two provided interfaces (squares) and two required interfaces (half-circles), forming a peer-to-peer relationship. Additionally, 'Operación Lógica de Unidades' is connected to three external interfaces: 'IRegistroClienteGO', 'INotificacionClienteGO', and 'INotificacionGO'. 'Gestor de Comunicaciones' is connected to two external interfaces: 'INotificacionUT' and 'IRegistroCO'. Figura 16. Composición interna del componente ControlOperaciones



5.2.2 View-packet Sistema Financiero

5.2.2.1 Vista de componentes y conectores

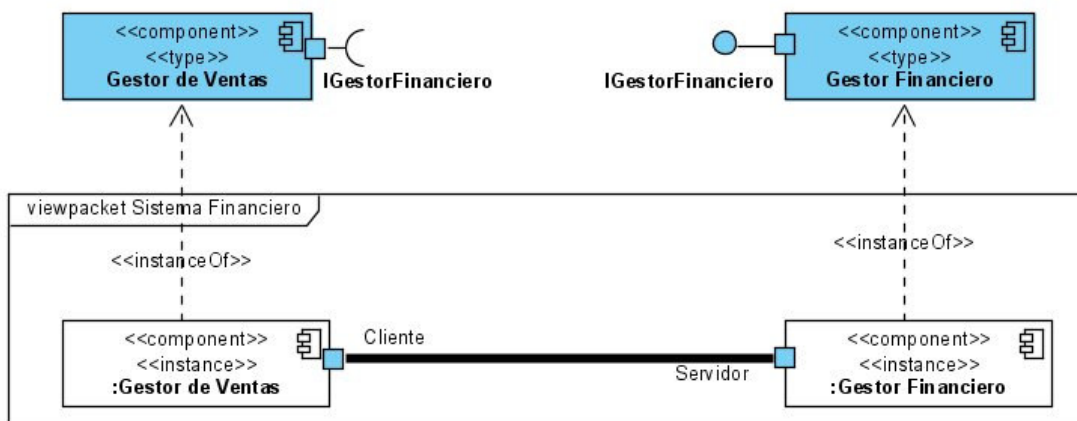


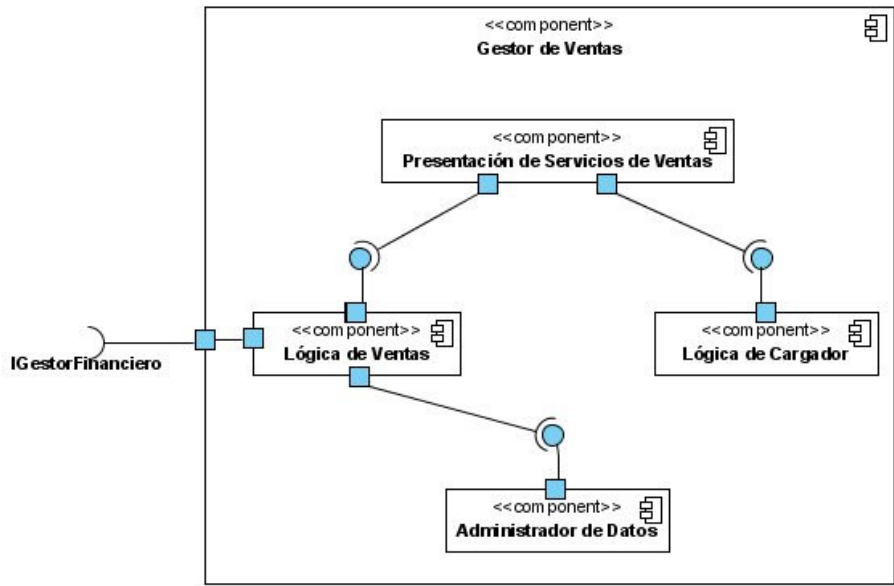
Figura 17. C&C View para el View Packet Sistema Financiero

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

5.2.2.2 Realización de la interacción Cliente-Servidor

Debido a la simplicidad del mensaje petición-respuesta propia del estilo cliente-servidor esta realización se ha omitido hacerse gráficamente.

5.2.2.3 Descripción del view packet Sistema Financiero

Nombre del View Packet		Sistema de Finanzas
Rationale		El patrón arquitectónico de <i>Cliente-Servidor</i> se considera adecuado puesto que brinda gran facilidad de agregar nuevos clientes sin un mayor impacto en la arquitectura. Debido a la naturaleza propia del problema (varios puntos de venta y recarga) es bastante deseable esta propiedad, producto de la posible escalabilidad que puede tener el servicio, lo que podría significar el aumento de los <i>gestores de venta</i> . Además la facilidad inherente de implementar protocolos de solicitud-respuesta nos da un marco para manejar los posibles errores que se pueden presentar en la transferencia de datos con una performance adecuada.
Catálogo de Componentes		
Componente:	Gestor Financiero (C4)	
Descripción	<i>Puede ser vista en el view packet interacción de componentes principales</i>	
Composición interna		
Componente:	Gestor de Ventas(C5)	
Descripción	Su labor consiste en enviar al <i>Gestor Financiero</i> todos los datos relativos a las ventas y recargas realizadas en el sistema físico asociado.	
Composición Interna	 <i>Figura 18. Composición interna del componente Gestor de Ventas</i>	

5.2.3 View packet Unidad de Transporte (Del segundo nivel de refinamiento)

5.2.3.1 Vista de Componentes y Conectores

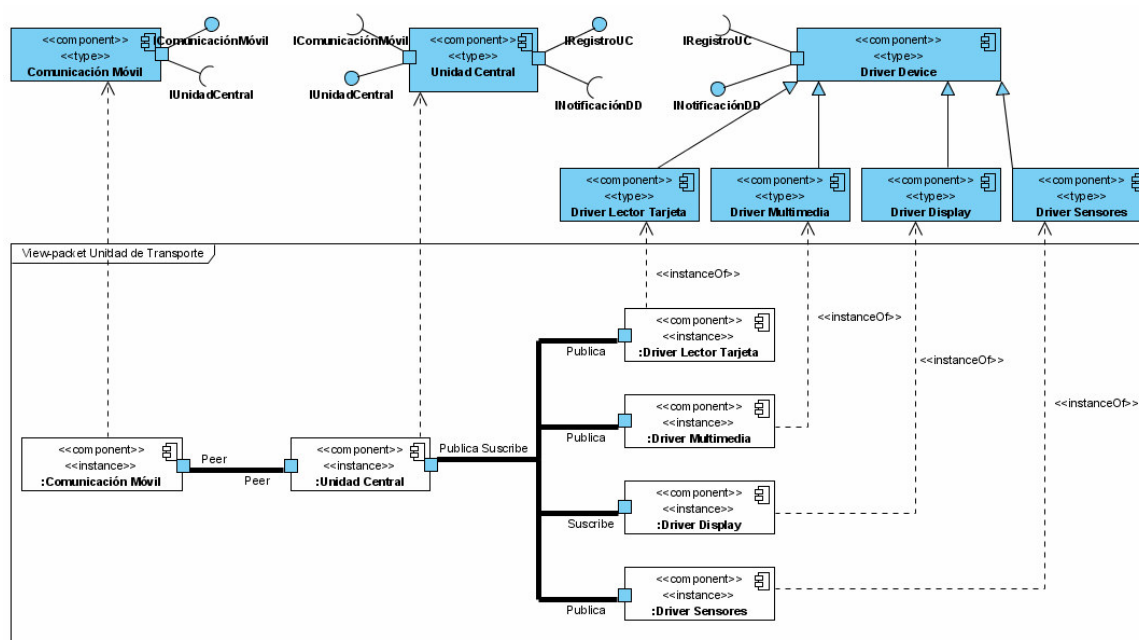


Figura 19. View packet Unidad de Transporte

5.2.3.2 Protocolo de Interacción

Ídem 5.2.1.2

5.2.3.3 Catálogo de Componentes

Nombre del View Packet	Unidad de Transporte View-packet
Rationale	Este view-packet se organizó usando dos patrones. Primero Peer-to-Peer para la interacción entre las componentes de comunicación y de Unidad Central, debido a que ambas partes necesitan solicitar servicios o información de la otra; la Unidad Central debe usar la Comunicación Móvil para enviar su información al Control de Operaciones mientras que éste le debe transmitir la información y los comandos provenientes del Control de Operaciones. Además, el patrón Publicador-Suscriptor para que los dispositivos comuniquen a la Unidad Central los eventos que están captando y el display muestre los eventos que le publica la Unidad Central.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Catálogo de Componentes	
Componente:	Comunicación Móvil(C6)
Descripción	Componente encargada del envío de datos desde la unidad hasta el Control de Operaciones.
Composición interna	No requiere descomposición arquitectónica (puesto que corresponde al segundo nivel de refinamiento)
Componente:	Unidad Central(C7)
Descripción	Tiene como labor canalizar los datos recibidos a través de los distintos devices presentes en la unidad.
Composición Interna	No requiere descomposición arquitectónica.
Componente	Driver Lector Tarjeta(C8)
Descripción	Encargada de controlar el dispositivo que lee una tarjeta BIP.
Composición Interna	No requiere descomposición arquitectónica.
Componente	Driver Multimedia(C9)
Descripción	Componente que controla los devices que obtienen datos multimediales dentro de la unidad, como la cámara y el micrófono.
Composición Interna	No requiere descomposición arquitectónica.
Componente	Driver Display(C10)
Descripción	Componente encargada de controlar lo que mostrará el display del chofer.
Composición Interna	No requiere descomposición arquitectónica.
Componente	Driver Sensores(C11)
Descripción	Controla los sensores del bus.
Composición Interna	No requiere descomposición arquitectónica.

5.3 Relación entre Componentes del view type C&C y los Módulos del View Type Módulos

		COMPONENTES Y CONECTORES	
		View-packets Involucrados	Componentes
Planificador Viaje	Presentación de Servicios de Información	Unidades Principales de Negocio view packet	<i>Cliente Servicios de Información</i>
	Gestor de Servicios de Información		<i>Lógica del Planificador</i>
	Persistencia de la Información de Servicios		<i>Administrador de Datos de Planificación</i>
Gestor Ventas	Presentación de Servicios de Ventas	Sistema de Gestión Financiero View-packet.	<i>Presentación de Servicios de Venta.</i>
	Controlador de Cargador		<i>Lógica de Cargador</i>
	Servidor de Ventas		<i>Lógica de Ventas</i>
	Persistencia de Ventas		<i>Administrador de Datos</i>
Gestor Operaciones	Presentación de Servicios de Operacionales	Unidades Principales de Negocio view packet	<i>ResentaciónServiciosOperaciones</i>
	Inteligencia de Operaciones		<i>Lógica de Gestión Operacional</i>
	Servicios de Información		<i>Acceso a Servicios de Operaciones</i>
	Control de Persistencia de Operaciones		<i>Control de Persistencia de Operaciones</i>
Control Operaciones	Control Unidades	Unidades Principales de Negocio view packet	<i>Operación Lógica de Unidades</i>
	Control Persistencia		<i>Operación Lógica de Unidades</i>
	Control Comunicaciones		<i>Gestor de Comunicaciones</i>
Gestor Financiero	Terminal Financiero	Unidades Principales de Negocio view packet	<i>Cliente Gestor Financiera</i>
	Servidor Financiero		<i>Lógica Financiera</i>
	Persistencia Financiera	Sistema de Finanzas View-packet.	<i>Administrador de Datos</i>
Unidad de Transporte	Unidad Central	Unidad de Transporte View-packet.	<i>Unidad Central</i>
	TarjetaLectora		<i>Driver Lector Tarjeta</i>
	ComunicaciónMovil		<i>Comunicación Móvil</i>
	ManejadorSensores		<i>Driver Sensor</i>
	ManejadorMultimedia		<i>Driver Multimedia</i>
	ManejadorDisplay		<i>Driver Display</i>

Tabla 2. Relación entre Componentes del view type C&C con los módulos del view type Módulos

6. Vista de Procesos

Para el desarrollo de esta vista se tuvieron en cuenta la vista de Deployment previamente definida y la lógica de la interacción presentada en el view type C&C de la vista lógica. Adicionalmente se definió cómo estos procesos son armados a partir de qué componentes arquitectónicos. Su relación con la vista de Deployment es presentada en dicha vista.

6.1 Modelo de Procesos para el SGT

El modelo de procesos es presentado en las siguientes figuras. Se ha realizado en dos partes particularmente por que la arquitectura general del sistema reutiliza la infraestructura de Procesos y Threads de Aplicaciones Cliente, Servidores de Presentación (por ejemplo. web), servidores de componentes y servidores de gestión de bases de datos. De manera particular, una parte crítica de la tecnología propuesta es la comunicación de los procesos que coordinan la interacción peer to peer entre el proceso del control de comunicaciones y los procesos de cada unidad de transporte.

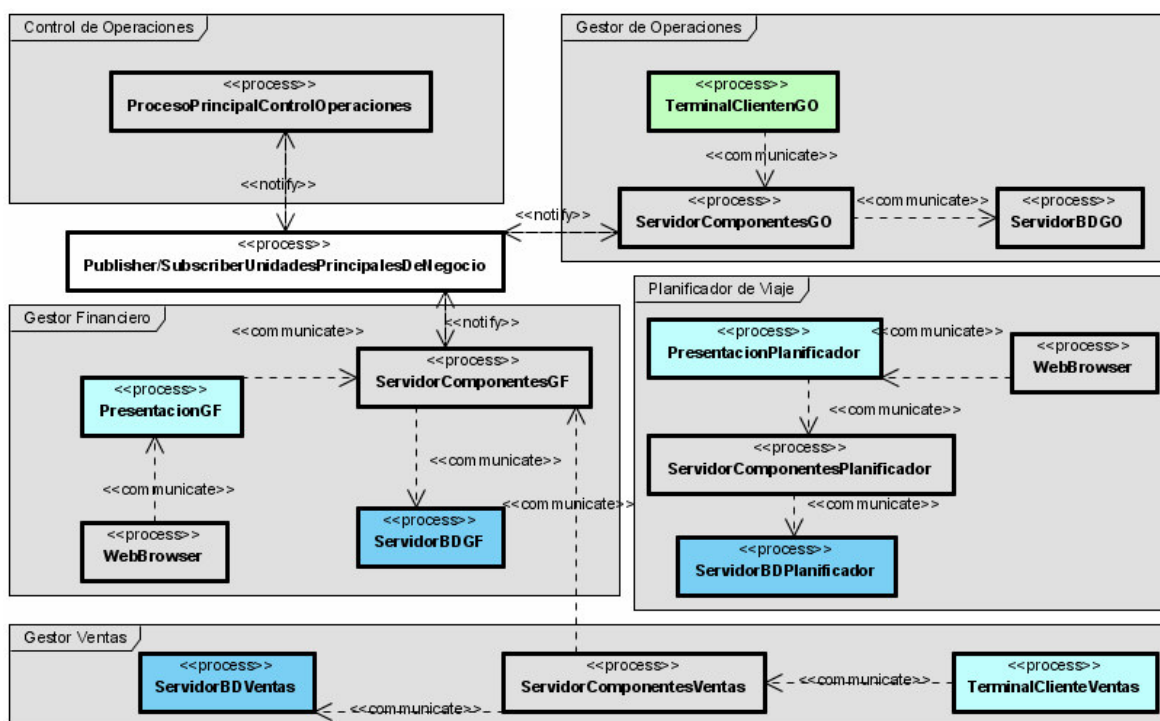


Figura 20. Modelo de Procesos General del Sistema

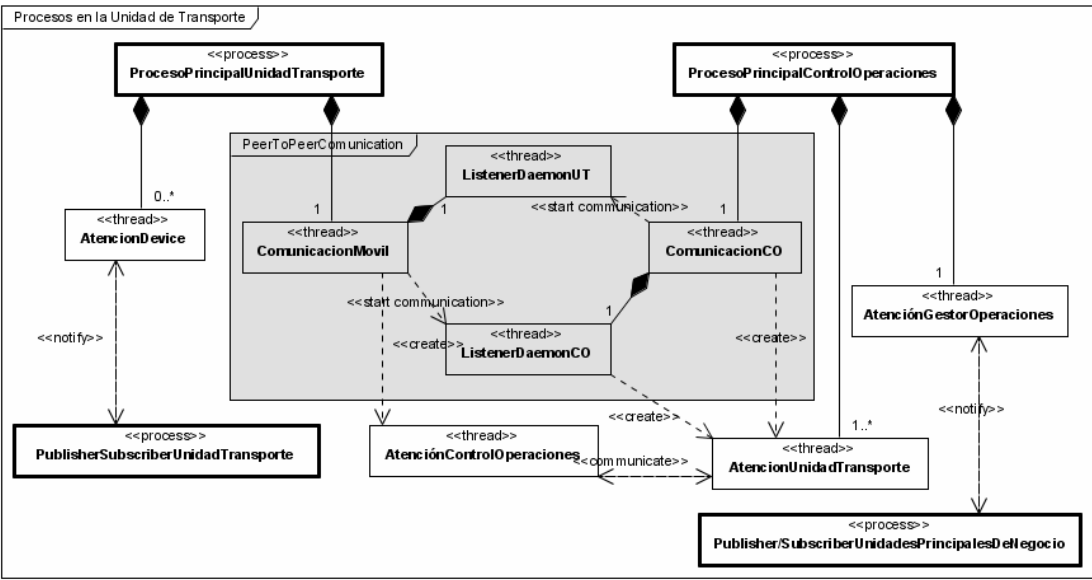


Figura 21. Modelo de Procesos Particular de una Unidad de Transporte

6.2 Relación con la Vista Lógica

A continuación se describe la forma en que los componentes forman los procesos correspondientes. Realmente los procesos en general corresponden a servidores con contenedores donde los componentes son puestos. Sin embargo un mismo componente puede abarcar dos o más procesos o viceversa.

La siguiente tabla presenta la relación correspondiente, en la vertical se presentan los procesos:

		COMPONENTES Y CONECTORES	
		Componentes	
Planificador r Viaje	Web Browser	Componente disponible en cualquier cliente web	
	Servidor Presentación	Presentación	
	Servidor de Componentes	Lógica del Planificador	
	Administrador de Datos de Planificación	Administrador de Datos de Planificación	
Gestor Ventas	Terminal Cliente Venta	Presentación de Servicios de Venta.	
	Terminal Cliente Venta	Lógica de Cargador	
	ServidorComponentesVenta	Lógica de Ventas	
	ServidorBDVentas	Administrador de Datos	
Gestión deOperacion es	TerminalCleinteGO	ResentaciónServiciosOperaciones	
	ServidorComponentesGO	Lógica de Gestión Operacional	
	ServidorComponentesGO	Acceso a Servicios de Operaciones	
	ServidorBDGO	Control de Persistencia de Operaciones	
Control Operaci ones	ProcesoPrincipalControlOperaciones	Operación Lógica de Unidades	
		Gestor de Comunicaciones	
		Cliente Gestor Financiera	
Gestor Financiero	BrowserWeb	Lógica Financiera	
	ServidorComponentesGF	Administrador de Datos	
	ServidorBDGF	Unidad Central	
Unidad de Transporte	Proceso Principal Unidad de Transporte	Driver Lector Tarjeta	
		Comunicación Móvil	
		Driver Sensor	
		Driver Multimedia	
		Driver Display	
		Publisher/Subscriber Unidades principales de negocio	
Conectores	Publisher/SubscriberUnidadPrincipal de Negocio (Servidor de Mensajería)		
	Proceso Principal Unidad de Transporte	Publisher subscriber Unidad de Transporte	

Tabla 3. Relación entre procesos y componentes del view type C&C de la vista lógica

7. Vista de Implementación

Para el desarrollo de la vista de implementación se consideró como punto de partida el primer nivel de refinamiento, parte del segundo y la vista de deployment. Por ello

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

en esta primera aproximación, los componentes incluidos corresponden a los principales componentes que serán instalados en los diferentes nodos definidos en la vista de deployment. La figura 22 muestra esta vista.

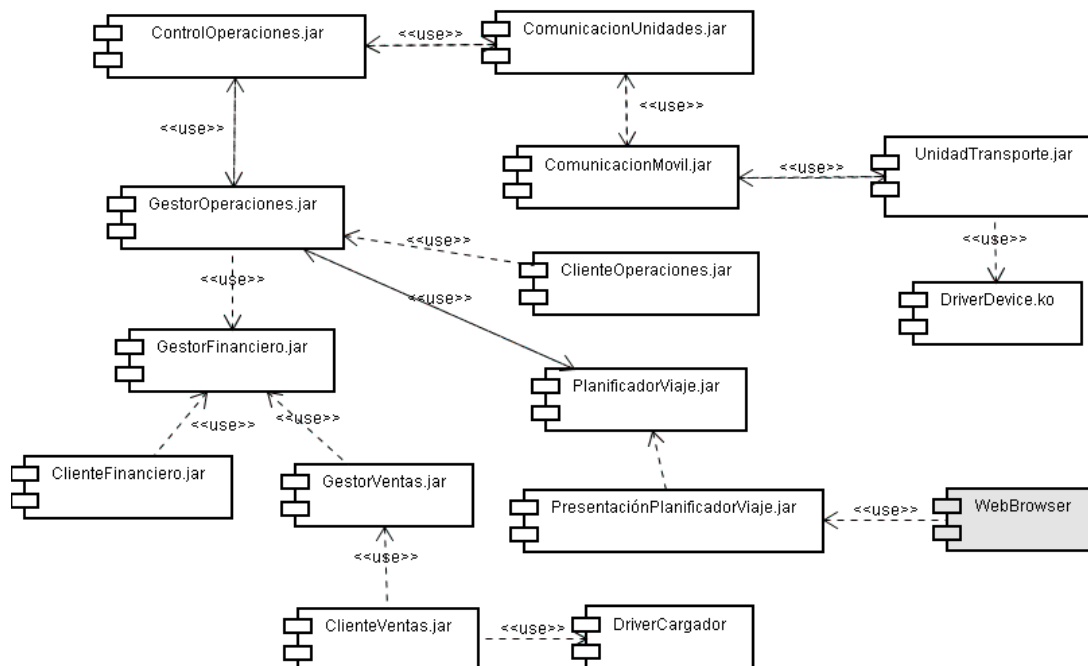


Figura 22. Modelo de Implementación

7.1 Relación con la vista lógica

7.1.1 Relación con el view type de módulos

La Tabla 4 presenta la relación de implementación entre los componentes de la vista de implementación y los módulos de la vista lógica.

		<i>ControlOperaciones.jar</i>	<i>GestorOperaciones.jar</i>	<i>ClienteOperaciones.jar</i>	<i>ComunicacionMovil</i>	<i>UnidadTransporte.jar</i>	<i>ComunicacionUnidades.jar</i>	<i>DriverDevice.ko</i>	<i>PlanificadorViaje.jar</i>	<i>PresentationPlanificadorViaje.jar</i>	<i>GestorFinanciero.jar</i>	<i>ClienteFinanciero.jar</i>	<i>GestorVentas</i>	<i>ClienteVentas.jar</i>	<i>DriverCargador.</i>
Planificador Viaje	Presentación de Servicios de Información									X					
	Gestor de Servicios de Información								X						
	Persistencia de la Información de Servicios								X						
Gestor Ventas	Presentación de Servicios de Ventas													X	
	Controlador de Cargador														X
	Servidor de Ventas												X		
	Persistencia de Ventas												X		
Gestor Operaciones	Presentación de Servicios de Operacionales			X											
	Inteligencia de Operaciones		X												
	Servicios de Información		X												
	Control de Persistencia de Operaciones		X												
Control Operaciones	Control Unidades	X													
	Control Persistencia	X													
	Control Comunicaciones						X								
Gestor Financiero	Terminal Financiero											X			
	Servidor Financiero										X				
	Persistencia Financiera										X				
Unidad de Transporte	Unidad Central					X									
	TarjetaLectora							X							
	ComunicaciónMovil				X										
	ManejadorSensores					X									
	ManejadorMultimedia					X									
	ManejadorDisplay					X									

Tabla 4. Relación de implementación entre componentes de la vista de implementación y los módulos de la vista lógica

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

7.1.2 Relación con la view Type C&C: Tecnología de Implementación de los conectores de la Arquitectura

Los conectores serán implementados a través de middleware de acuerdo a la tecnología seleccionada, así:

- El cliente servidor vía Servidores de aplicación, servidores web, servidores de componentes y servidores de gestión de bases de datos.
- El conector publicador-Subscriber será implementado a través de mensajería vía un servidor de mensajería y componentes de negocio manejados por mensajes o eventos (por ejemplo en el caso de una implementación J2EE Message Driven EJBs).
- El conector peer to peer puede ser implementado vía Hilos de ejecución y Sockets.

7.2 Directrices generales para la implementación de los componentes

Los componentes de la implementación deberán ser empaquetados en los respectivos servidores como:

- **Componentes de Cliente:** componentes relacionados con la lógica de presentación del lado del cliente.
- **Componentes de presentación:** componentes relacionados con la lógica de presentación del lado del servidor.
- **Componentes de negocio:** componentes relacionados con la lógica del negocio.
- **Componentes de mensajería:** componentes relacionados con la publicación y suscripción de eventos en el conector de publicación/suscripción.
- **Componentes de acceso a datos:** componentes relacionados con el acceso a los repositorios de datos.

8. Vista de Despliegue (Deployment) o Física

La vista de deployment presenta la infraestructura necesaria para dar soporte a TransMontevideo. Aquí se presenta la arquitectura técnica indicando los nodos presentes en una primera aproximación a lo que es el modelo físico. El modelo de Deployment es presentado en la Figura 3. Este modelo presenta una aproximación preliminar de los principales nodos de la solución y una breve justificación de su inclusión. Para esta descripción se han omitido los elementos de comunicación como la red de área metropolitana, la red de área local, la Internet y la red de posicionamiento satelital como nodos y se ha adjudicado esta información al enlace entre los nodos por medio de los estereotipos MAN, LAN, Internet/http y GPSN

respectivamente.

8.1 Arquitectura Técnica

Según la descripción del proyecto y la información que se ha recopilado durante la descripción de los casos de uso, se han identificado los siguientes tipos de nodos, a saber, Validador Taxi, Validador Bus, Terminal de Centro de Ventas, Central de Ventas, Servidor Central de Operaciones, Terminal de Operaciones, Central de Gestores Financieros y Terminal de Gestor Financiero. La Figura 13 presenta los nodos principales.

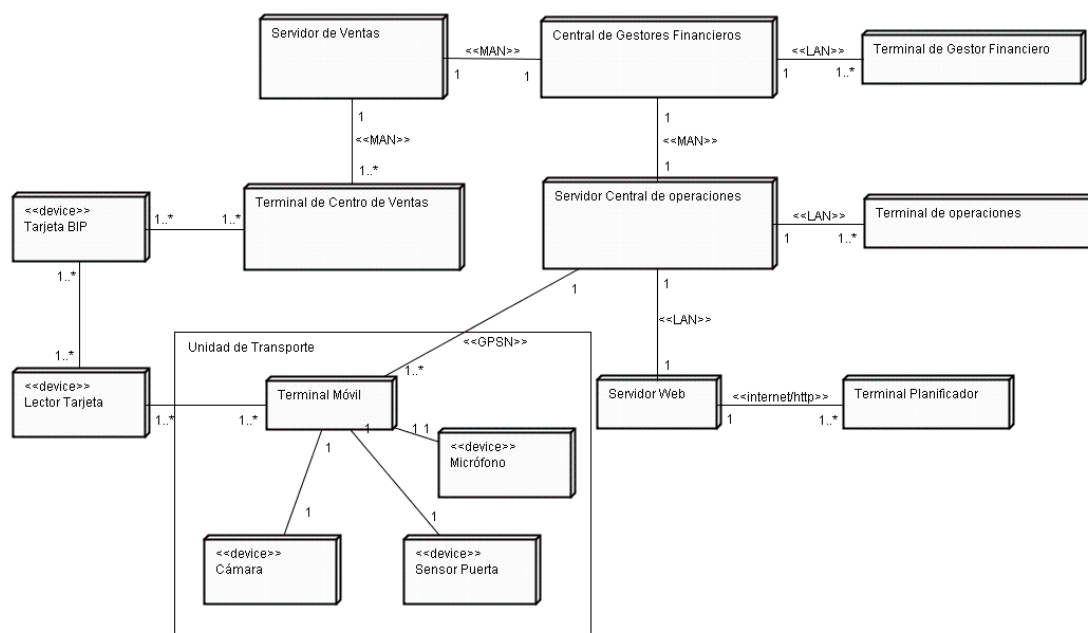


Figura 23. Modelo de Deployment del SGT

La figura 23 presenta los nodos y sus enlaces. Hay que tener en cuenta que esta es una solución minimalista, es decir, el número de nodos identificados es el mismo, sin embargo la arquitectura soporta el crecimiento en nodos haciendo uso de los mismos componentes arquitectónicamente definidos y diseñados. Cualquier local comercial que atienda al público podrá vender tarjetas bip! y ofrecer el servicio de recarga. El local comercial deberá tener un nodo de Terminal de Centro de Ventas es donde las tarjetas BIP se activan y se recargan. Estos nodos terminales están conectados a una Central de Ventas quien se encargará de recopilar la información de las recaudaciones de los Terminales para después suministrar periódicamente esa información al nodo Central de Gestores Financieros.

Los nodos Validador Bus y Validador Taxi son los que reciben la información de las tarjetas BIP para enviarla a la Central de Operaciones. El nodo Central de

Operaciones es el encargado de procesar los datos las tarjetas BIP y de los chips GPS para proveer de información a los Terminales de Operaciones y a la Central de Gestores Financieros. Los Terminales de Gestores Financieros son los encargados de mostrar la información obtenida de la Central de Gestores Financieros, para que los gestores financieros puedan observar el estado de sus recaudaciones.

Los pasajeros y operadores telefónicos a través de un nodo terminal web podrán ofrecer información para la planificación del viaje. Estos nodos se conectarán a un servidor web, el cual está integrado al nodo central de operaciones. El servidor web ofrecerá toda la información estática del plan: rutas, horarios, paraderos, terminales, lugares de compra y recarga de tarjeta, pero también ofrecerá información dinámica como un servicio de planificación de viaje que ofrezca una combinación de servicios que permita ir de un punto a otro de la ciudad, así como información en línea de la ubicación de los buses próximos a un paradero. El servidor web deberá ofrecer la solicitud de un taxi a la ubicación del pasajero.

La vista deployment con los componentes principales instalados se puede visualizar en la Figura 15.

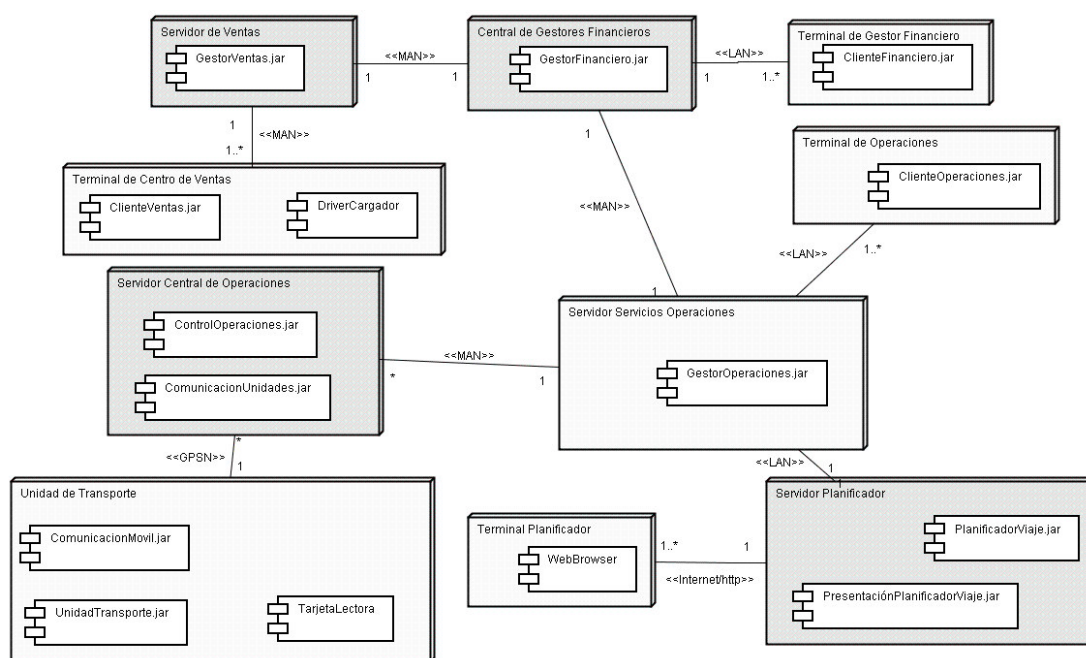


Figura 24. Modelo de Deployment del SGT con los componentes de deployment respectivamente desplegados.

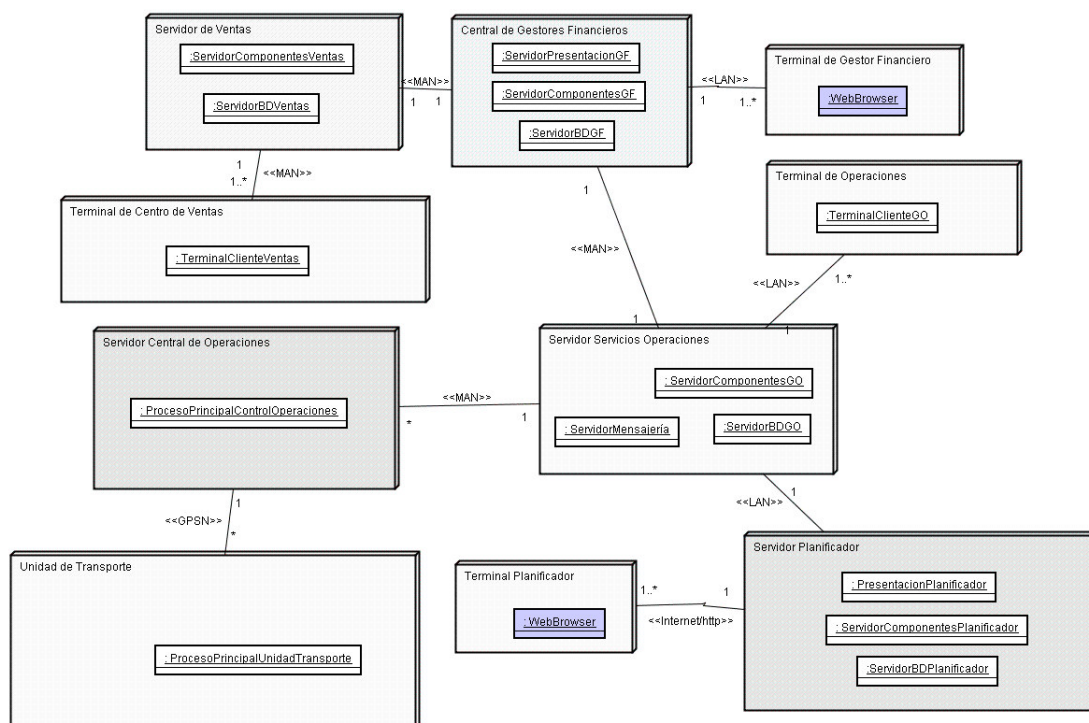


Figura 25. Modelo de Deployment del SGT con los procesos respectivamente desplegados.

En este último diagrama es importante hacer referencia a que el Servidor de Mensajería ejecutándose en el nodo *servicios operaciones* es un alias del proceso Publisher/SubscriberUnidadesPrincipales de Negocio definido en la vista de procesos. Esto es debido a que en cada vista los respectivos nombres definen mejor el elemento de modelado a representar.

El Deployment de procesos y de componentes de implementación es un caso particular de implantación. Debido a que una de las principales cualidades de la arquitectura propuesta es la escalabilidad, los componentes y procesos definidos pueden ser puestos en diferentes nodos guardando el marco general de interacción entre los tipos de nodos expresados en el modelo de Deployment. Así por, ejemplo, un proceso servidor de componentes puede ser desplegado en dos nodos diferente aliviando la carga en caso de crecimiento (escalabilidad horizontal) o los procesos y componentes pueden ser ejecutados/desplegados en diferente máquina, hasta hacer coincidir cada nivel de un n-tier con un nodo (Escalabilidad Vertical). Incluir un nuevo nivel se sale del alcance de la flexibilidad de la arquitectura propuesta. (Escalabilidad Vertical Restringida).

8.2 Tecnología requerida

La empresa desarrolladora ha determinado que para poder comercializar el producto

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

contar con una tecnología basada en máquinas virtuales como Java podría ayudar a ganar portabilidad (vista como una forma de modificabilidad) en la solución, así como trabajar en una arquitectura que dé soporte a los requisitos funcionales y no funcionales de la solución en el contexto de esta tecnología (si es la tecnología escogida finalmente). Así la tecnología candidata inicial, desde una perspectiva técnica es la tecnología java (aún no se determinan los proveedores):

- Sistemas Operativos Linux, WindowsXP
- Servidores Web J2EE y Framework MVC
- Servidores de Componentes J2EE.
- Servidores de Transacciones JTS
- Servidores de Mensajería JMS
- Servidores de Bases de Datos(Oracle 9i) y JDBC
- Clientes J2SE y J2ME
- Tecnología Java Card
- JCE para seguridad

Algunos nodos incluyen una o varias de estas tecnologías y en un análisis siguiente se podrán hacer las asignaciones de las tecnologías respectivas a los nodos de procesamiento de acuerdo al refinamiento del modelo de deployment y a las plataformas, frameworks y componentes que se instalen en ellos.

9. Rationale

El diseño arquitectural se realizó siguiendo el proceso unificado y el método de diseño arquitectural guiado por los atributos de calidad identificados en la etapa anterior. Una de las principales tareas fue documentar y entender el método ADD para posteriormente aplicarlo.

Dado que tuvimos dos niveles de refinamiento el ADD fue aplicado al Sistema para definir la estructura lógica entre los principales subsistemas y fue parcialmente aplicado en el segundo nivel de refinamiento. Se espera en una siguiente entrega retomar el ADD para la revisión de la descomposición de los subsistemas en gran medida porque muchos controladores de la arquitectura llevan a tácticas y patrones que requieren de otras vistas para ser representados.

Antes de descomponer el sistema completo se revisó que no hicieran falta requisitos relevantes por medio de un chequeo y una discusión entre los participantes. Posteriormente, se priorizaron los atributos de calidad y casos de uso en términos de relevancia arquitectónica y de su localidad/globalidad. Esta información fue agregada a la descripción de los casos de uso y los escenarios de calidad.

Basándonos en esta priorización se identificaron los drivers arquitectónicos para la primera descomposición, a estos drivers y a cada atributo de calidad le asociamos las tácticas¹ a tener en cuenta y los patrones que podrían ayudar a implementar las tácticas. Lo hicimos para todos los atributos, driver y no drivers, debido a que aunque algunos no resultaron ser drivers en la primera descomposición si lo fueron en la segunda en el módulo dónde se localiza el atributo de calidad y termina controlando el diseño de su arquitectura.

Así, se identificó qué patrones permiten implementar las tácticas y cuales requisitos eran localizados o eran globales, para la primera descomposición se tuvieron en cuenta los que fuesen globales (modificabilidad y escalabilidad) y la descomposición funcional que inicialmente se había logrado por medio de los casos de uso y la vista de deployment. Los patrones relevantes fueron:

Desde un view type de módulos:

- Descomposición, uso y generalización (patrones naturalmente aplicados)
- Capas para la separación de preocupaciones y ocultamiento de la información.

Desde un view type de C&C:

- Publicador subscriptor para disminuir acoplamiento (modificabilidad) y escalabilidad.
- Cliente/Servidor para conseguir escalabilidad horizontal.
- Tiers para conseguir escalabilidad horizontal y vertical.
- Peer to Peer: en situaciones de colaboración requerida entre pares que se asumen como clientes y servidores en su interacción mutua.

<i>Driver Candidato</i>	<i>Concerns</i>	<i>Tácticas</i>	<i>Patrones Satisfactores y/o Decisiones de diseño</i>
QS1	Confidencialidad	Tácticas encriptación, ocultamiento del núcleo, controlar el acceso.	Capas Tuberías y Filtros Peer to peer Tiers
QS2	Latencia	asincronía de procesos, tiempo de ejecución (parte del proceso puede ser en diferido)	Publicador/Subscriptor
QS3	Tolerancia a fallos	Tácticas prevención, prueba de correctitud, logging, backward recovery, redundancia	Cliente Servidor Peer to peer Redundancia de Protocolos

¹ Una táctica arquitectural es una decisión de diseño que influencia las propiedades de un sistema. Por ejemplo, una táctica ping-echo para detección de fallos puede ser usada durante el diseño para influenciar las propiedades de disponibilidad de un sistema. La táctica de ocultamiento de la información puede ser usada durante el diseño de la arquitectura para alcanzar las propiedades de modificabilidad.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

			Coherencia
QS4	Seguridad	Tácticas encriptación	Tuberías y Filtros Puede ser Localizado como un componente o un Aspecto.
QS5	Modificabilidad	Ocultamiento de la información, ligadura tardía, separación de intereses.	Capas Publicador Subscriptor Tiers
QS6	Escalabilidad	Independencia modular, ligadura tardía, Asícronía	Publicador subscriptor Cliente Servidor

Tabla 5. Relacionando drivers, tácticas y patrones

Se analizaron los diferentes atributos a la luz de los diferentes patrones. Del análisis se selecciona el patrón publicador/subscriptor y se instancia el patrón para el caso de estudio. Parte del rationale se encuentra distribuido a lo largo de la especificación de la arquitectura.

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Anexo 1. Descriptor del método QAW adaptado a UP

Elicitación de escenarios de atributos de calidad usando una versión liviana del Workshop de atributos de calidad(QAW)	
Propósito: el método QAW facilita a los diferentes participantes del proyecto descubrir en etapas tempranas del ciclo de vida los requisitos manejados por los atributos de calidad del software. Los puntos clave acerca del QAW son: centrado en el sistema, enfocado por los participantes, y usado antes de que la arquitectura haya sido creada.	
Rol: Analista	
Frecuencia: esta actividad ocurre cuando sea requerida, típicamente una vez por iteración en la fase de inepción y una vez en la fase de elaboración.	
Pasos	Descripción
Paso 1. Presentación e introducción del QAW	Los facilitadores del método QAW describen la motivación del método y explican cada uno de los pasos del método. Posteriormente, los facilitadores y el resto de participantes se presentan de la misma forma, brindando un resumen de su currículo y experiencia, su rol en la organización y su relación con el sistema en construcción.
Paso 2. Presentación de los participantes de negocio y/o de gestión.	Un participante, representando las incumbencias de negocio y/o gestión presenta el contexto de negocio y/o gestión para el sistema, los requisitos funcionales de alto nivel, restricciones y requisitos de atributos de calidad.
Paso 3. Presentación del plan de la solución técnica.	Un participante del área técnica presenta los planes técnicos incluyendo: a. Los planes y estrategias sobre cómo se satisfarán los requisitos de negocio y de gestión. b. Los requisitos técnicos claves , riesgos y restricciones – tales como sistemas operativos obligatorios, hardware, software y estándares – que controlarán las decisiones del sistema. c. Diagramas de contexto existentes, diagramas del sistema a alto nivel, y otras descripciones escritas. d. Arquitecturas operacionales y de sistema, frameworks arquitectónicos, herramientas y ciclo de vida de los procesos arquitectónicos siendo usados. e. Los estudios de ingeniería y prototipado internos para mitigar los riesgos conocidos.
Paso 4. Identificar los drivers de la arquitectura.	Los facilitadores comparten su lista de controladores claves que incluyen requisitos de alto nivel, controladores de negocio, restricciones y atributos de calidad.
Paso 5. Lluvia de ideas de escenarios	Los facilitadores preguntan a los participantes para obtener una lluvia de ideas de los escenarios que son operacionalmente significativos con respecto a los roles individuales de los participantes.
Paso 6. Consolidación de escenarios	Escenarios similares son consolidados cuando sea razonable
Paso 7. Priorización de escenarios	Los participantes votan para establecer la prioridad de los escenarios.
Paso 8. Refinamiento de escenario	Los escenarios de más alta prioridad son refinados en mayor detalle. Los facilitadores posteriormente elaboran cada uno la siguiente documentación: las seis partes del escenario, los objetivos del negocio/gestión que son afectados por este escenario, los atributos de calidad relevantes asociados al escenario y las formas de observación del escenario
Artefactos de entrada • Caso del negocio • Documento de visión [Plan arquitectónico]	Artefactos de salida • Caso de negocio [QAW refina los objetivos de negocio, a través de una retroalimentación al caso de negocio] • Especificación suplementaria [como repositorio de escenarios]
Herramientas de soporte: ninguna	
Más información: Barbacci, M.R.; et. Al. Quality Attribute Workshps(QWAs)	
Detalles del flujo de trabajo: • Requerimientos • Entender las necesidades de los participantes	

Sistema de Gestión del TransMontevideo - SGT	
Software Architecture Document	Date: <5/24/2007>

Anexo 2. Descriptor del método ADD adaptado a UP

Diseño de la Arquitectura de Software usando el método ADD – Attribute-Driven Design	
Propósito: el método ADD – Attribute Driven Design es un enfoque para definir arquitecturas de software basando el proceso en los requisitos de la arquitectura, los atributos de calidad. Este sigue un proceso de descomposición recursivo, donde, en cada estado de descomposición, se escogen tácticas y patrones para satisfacer un conjunto de escenario de atributos de calidad.	
Rol: Arquitecto de Software	
Frecuencia: Esta actividad es opcional en la fase de inicio. Esta debe ocurrir en la primera iteración de la fase de elaboración y puede necesitar de iteraciones siguientes si hay cambios substanciales o adiciones a la arquitectura de software a ser diseñada	
Pasos	Descripción
Paso 1. Escoger el módulo a descomponer	Usar el sistema completo como módulo inicial al empezar con ADD. Todas las entradas requeridas para este modulo deben estar disponibles (restricciones, requisitos funcionales, y requisitos de calidad)
Paso 2. Refinar el módulo de acuerdo a estos pasos.	
a. Escoger los manejadores arquitecturales	Escoger los controladores arquitecturales desde un conjunto concreto de escenarios de calidad y requisitos funcionales. Este paso determina qué es importante para su descomposición.
b. Escoger un patrón arquitectural que satisfaga los controladores de la arquitectura.	Crear o seleccionar un patrón basándose en las tácticas que pueden ser usadas para lograr los controladores. Identificar los módulos hijos requeridos para implementar las tácticas.
c. Instanciar los módulos y ubicar la funcionalidad de los casos de uso. Represente los resultados utilizando múltiples vistas.	Considerar las vistas de cada uno de los tres mayores grupos de vistas (Descomposición lógica, concurrencia, y despliegue)
d. Defina las interfaces de los módulos hijos.	La descomposición provee módulos y restricciones sobre los tipos de interacción entre módulos. Documentar esta información en los documentos de interface de cada módulo.
e. Verificar y refinar los casos de uso y escenarios de calidad y hacerlos restricciones para los módulos hijos.	Este paso verifica que no fue olvidado nada importante y prepara los módulos hijos para una siguiente descomposición o implementación.
Paso 3. Repetir los pasos anteriores para el siguiente módulo.	
Artefactos de entrada • Visión [restricciones] • Prueba de concepto arquitectónico • Modelo de casos de uso [requisitos funcionales, requisitos de calidad] • Especificaciones complementaria [requisitos de calidad]	Artefactos de salida • Documento de la arquitectura de software [descomposición de la arquitectura expresada en las vistas lógica, de concurrencia y despliegue]
Herramientas de soporte: ninguna	
Más información: Bass, L; Clements, P; & Kazman, R. Software Architecture in practice. Segunda Edición.	
Detalles del flujo de trabajo: • Análisis y Diseño ○ Definir una arquitectura candidata ○ Lograr la síntesis arquitectural.	