

MOBFLOW PLATAFORMA DE DESPLIEGUE DISTRIBUIDO DE PROCESOS DE NEGOCIO EN SISTEMAS MÓVILES DE INFORMACIÓN

ANEXOS



Gustavo Adolfo Aponzá Villaquirán

Henry William Dorado Gomez

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Línea de investigación en Ingeniería de Sistemas Telemáticos

Popayán, Junio de 2010

MOBFLOW PLATAFORMA DE DESPLIEGUE DISTRIBUIDO DE PROCESOS DE NEGOCIO EN SISTEMAS MÓVILES DE INFORMACIÓN

ANEXOS



Gustavo Adolfo Aponzá Villaquirán

Henry William Dorado Gomez

Director

Dr. Ing. Juan Carlos Corrales

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones

Departamento de Sistemas

Línea de investigación en Ingeniería de Sistemas Telemáticos

Popayán, Junio de 2010

TABLA DE CONTENIDO

ANEXO A.	
ARTEFACTOS OBTENIDOS EN LA PLANEACIÓN, CONSTRUCCIÓN Y ELABORACIÓN .	3
ANEXO B.	
MANUAL DE INSTALACIÓN Y DE USUARIO	59
ANEXO C.	
FORMATO DE LOS PROCESOS BPEL Y WSDL SOPORTADOS POR LA PLATAFORMA MOBFLOW	73
ANEXO D.	
EJEMPLOS DETALLADOS PARA LOS ALGORITMOS PROPUESTOS	82
ANEXO E.	
TABLAS DE RESULTADOS DE LA PRUEBA DE RENDIMIENTO Y ESTABILIDAD	94
ANEXO F.	
PRUEBAS CON EMULADOR DE SUN JAVA WIRELESS TOOLKIT	121

ANEXO A. ARTEFACTOS OBTENIDOS EN LA PLANEACIÓN, CONSTRUCCIÓN Y ELABORACIÓN

A1. FASE DE PLANEACIÓN Y ELABORACIÓN

En esta sección se presentan el análisis y diseño de las dos herramientas software desarrolladas en el presente trabajo de grado, MobFlow y cliente móvil MobFlow.

A1.1 Casos de Uso

En esta sección se presentan los diagramas de casos de uso y su descripción para las dos herramientas software desarrolladas en el presente trabajo de grado, MobFlow y cliente móvil MobFlow. Estos casos de uso se identificaron a partir de las funcionalidades que ofrece el sistema a los diferentes actores.

A1.1.1 Diagrama de casos de uso MobFlow

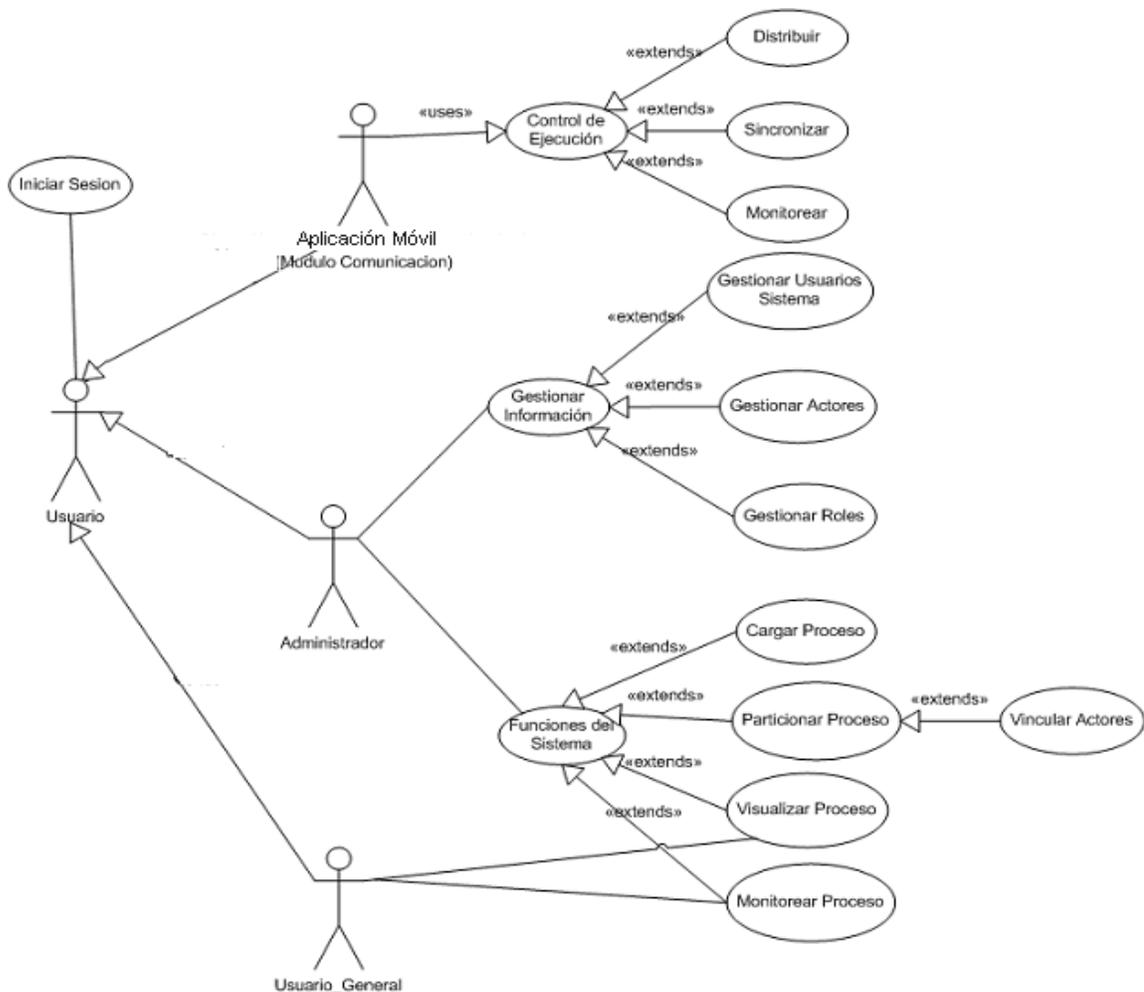


Figura A1. Diagrama de casos de uso MobFlow

La figura A1 presenta el diagrama de casos de uso general para la aplicación Web MobFlow, donde se pueden apreciar los tipos de usuarios que interactúan con la aplicación y las funcionalidades del sistema

A1.1.2 Casos de uso en formato expandido MobFlow

En esta sección se presenta los casos de uso en formato expandido para la aplicación Web MobFlow

Caso de uso:	Iniciar Sesión (Autenticación)	
Actores:	Usuario (Iniciador).	
Propósito:	Ingresar al sistema.	
Resumen:	La persona ingresa en el sistema para realizar las diferentes tareas disponibles que brinda la plataforma.	
Tipo:	Primario y extendido.	
Curso normal de eventos		
Acción de los Actores		Respuesta del sistema
<ol style="list-style-type: none"> 1. Este caso de uso se inicia cuando un usuario quiere ingresar al sistema. 2. El usuario llena los campos Nick y Password para realizar su ingreso. 3. El usuario selecciona ingresar. 		<ol style="list-style-type: none"> 4. El sistema presenta la interfaz inicial de MobFlow, al nuevo usuario para que pueda hacer uso de los diferentes servicios que ofrece la plataforma.

Tabla A1 - Caso de uso en formato expandido iniciar sesión.

Caso de uso:	Cargar proceso de negocio.	
Actores:	Administrador (Iniciador).	
Propósito:	Cargar los procesos de negocio en la plataforma Web.	
Resumen:	Esta interfaz, permite cargar los procesos de negocio representados por un documento BPEL y un archivo WSDL, además permite ingresar la información del nombre y la descripción del proceso.	
Tipo:	Primario y extendido.	
Curso normal de eventos		
Acción de los Actores		Respuesta del sistema
<ol style="list-style-type: none"> 1. Este caso de uso se inicia cuando el usuario Administrador quiere cargar un proceso de negocio definido previamente. 2. El usuario llena el campo nombre, que corresponde al nombre del proceso de negocio. 3. El usuario llena el campo Descripción, que corresponde a la descripción del proceso. 4. El usuario selecciona los archivos BPEL y WSDL del proceso de negocio. 		

5. El usuario carga el proceso de negocio.	6. El sistema carga la información de proceso de negocio en la base de datos y almacena el documento BPEL y WSDL correspondiente. 7. El sistema despliega la información del proceso.
Curso alternativo 1 – Línea 4	
Acción de los Actores	Respuesta del sistema
4.1. El usuario realiza la búsqueda del recurso que quiere subir (documento BPEL).	4.2 El sistema carga la ruta del documento BPEL que se quiere subir.
5.1. El usuario realiza la búsqueda del recurso que quiere subir (documento WSDL).	5.2 El sistema carga la ruta del documento WSDL que se quiere subir.

Tabla A2. Caso de uso en formato expandido Cargar proceso de negocio.

Caso de uso:	Particionar proceso de negocio.
Actores:	Administrador (Iniciador).
Propósito:	Lleva a cabo el particionamiento de los procesos de negocio, en subprocesos sincronizados y guardar la información en la base de datos de la plataforma.
Resumen:	Al iniciar el particionamiento del proceso previamente cargado, la plataforma crea uno a uno los documentos BPEL correspondientes a los subprocesos de negocio sincronizados. Una vez terminado el particionamiento se muestra la información en pantalla.
Tipo:	Primario y extendido.
Curso normal de eventos	
Acción de los Actores	Respuesta del sistema
1. Este caso de uso se inicia cuando un usuario quiere particionar el proceso de negocio cargado previamente. 2. El usuario da inicio al proceso de particionamiento.	3. El sistema particiona el proceso de negocio en documentos BPEL sincronizados, y almacena la información de los subprocesos obtenidos. 4. El sistema despliega una ventana con los datos de los subprocesos obtenidos.

Tabla A3. Caso de uso en formato expandido Particionar proceso de negocio.

Caso de uso:	Vincular Actores.	
Actores:	Administrador (Iniciador).	
Propósito:	Vincular los actores que ejecutan los subproceso de negocios generados por la plataforma.	
Resumen:	Una vez particionado el proceso de negocio, se procede a vincular los actores que ejecutaran cada subproceso, para ellos se seleccionan los subprocesos generados y se vinculan los actores dependiendo del rol que desempeñen en la ejecución del proceso.	
Tipo:	Primario y extendido.	
Curso normal de eventos		
Acción de los Actores		Respuesta del sistema
<p>1. Este caso de uso se inicia, cuando un usuario va a vincular un subproceso de negocio que fue particionado.</p> <p>2. El usuario selecciona el subproceso que va a vincular.</p> <p>4. El usuario selecciona el actor encargado de ejecutar el subproceso.</p> <p>5. El usuario llena el campo número de dispositivo, que corresponde al dispositivo que ejecutara el subproceso de negocio.</p> <p>6. El usuario vincula el actor del subproceso de negocio seleccionado.</p>		<p>3. El sistema despliega una nueva ventana, con las opciones necesarias para vincular el actor del subproceso.</p> <p>7. El sistema despliega la información del actor y dispositivo vinculado.</p>

Tabla A4. Caso de uso en formato expandido Vincular actores.

Caso de uso:	Visualizar el proceso de negocio.	
Actores:	Administrador (Iniciador).	
Propósito:	Visualiza un proceso de negocio cargado.	
Resumen:	Después de cargar el documento BPEL con la descripción del proceso de negocio, la página de visualización muestra una descripción gráfica del proceso.	
Tipo:	Primario y extendido.	
Curso normal de eventos		
Acción de los Actores		Respuesta del sistema
<p>1. Este caso de uso se inicia, cuando un usuario va a visualizar un proceso de negocio que ha sido cargado previamente.</p> <p>2. El usuario da inicio al proceso de despliegue.</p>		<p>3. El sistema despliega una ventana, con la tabla de procesos y la imagen del proceso seleccionado.</p>

Tabla A5. Caso de uso en formato expandido Visualizar el proceso de negocio.

Caso de uso:	Monitoreo de los procesos de negocio.	
Actores:	Administrador (Iniciador).	
Propósito:	Monitoreo de la ejecución de un proceso de negocio.	
Resumen:	Permite monitorear la ejecución de los procesos de negocio, para realizar dicha acción se tuvo que haber cargado, particionado, configurado y ejecutado el proceso de negocio, para mostrar en la interfaz de monitoreo el estado de la ejecución de cada proceso.	
Tipo:	Primario y extendido.	
Curso normal de eventos		
Acción de los Actores	Respuesta del sistema	
<p>1. Este caso de uso se inicia, cuando un usuario va a monitorear la ejecución de un proceso de negocio que ha sido ejecutado previamente.</p> <p>2. El usuario selecciona el subproceso que quiere monitorear.</p>	<p>3. El sistema despliega una ventana, en esta interfaz se muestra un grafico con el proceso de negocio y las actividades que se han ejecutado, las cuales aparecen marcadas, además se muestra la información de la ejecución del proceso en donde se encuentran los datos como: actividades que se ejecutan, el actor, el subproceso, el dispositivo y la fecha y hora en que fueron ejecutadas dichas actividades en los dispositivos móviles.</p>	

Tabla A6. Caso de uso en formato expandido Monitoreo de los procesos de negocio.

Caso de uso:	Monitorear.	
Actores:	Administrador (Iniciador).	
Propósito:	Registra las actividades ejecutadas.	
Resumen:	Una vez ejecutado el subproceso de negocio en la aplicación móvil, se procede a registrar cada una las actividades ejecutadas en la plataforma.	
Tipo:	Primario y extendido.	
Curso normal de eventos		
Acción de los Actores	Respuesta del sistema	
<p>1. Este caso de uso se inicia, cuando un actor ejecuta un proceso de negocio.</p> <p>2. El usuario ejecuta una actividad del proceso y manda la información de dicha actividad a la plataforma Web.</p>	<p>3. El sistema móvil se conecta con la aplicación Web, enviando la información de la actividad ejecutada.</p> <p>4. La aplicación Web recibe la información y registra los datos correspondientes a la actividad ejecutada.</p>	

Tabla A7. Caso de uso en formato expandido Monitorear.

Caso de uso:	Distribuir.
Actores:	Administrador (Iniciador).
Propósito:	Distribuye los subprocesos generados por la aplicación Web.
Resumen:	Una vez generados los subprocesos de negocio y vinculado cada actor del proceso la plataforma distribuye los subprocesos BPEL a los dispositivos móviles.
Tipo:	Primario y extendido.
Curso normal de eventos	
Acción de los Actores	Respuesta del sistema
<ol style="list-style-type: none"> Este caso de uso se inicia, cuando un actor procede a descargar el subproceso correspondiente. El actor ingresa sus datos y los envía a la plataforma para descargar los archivos. 	<ol style="list-style-type: none"> La aplicación Web autentica la información y retorna la ubicación de los archivos del proceso. La aplicación móvil descarga los archivos del proceso

Tabla A8. Caso de uso en formato expandido Distribuir

Caso de uso:	Sincronizar.
Actores:	Administrador (Iniciador).
Propósito:	Sincroniza la ejecución de los subprocesos de negocio.
Resumen:	Una vez se inicia la ejecución del proceso de negocio, se procede a sincronizar cada subproceso con el siguiente en la ejecución.
Tipo:	Primario y extendido.
Curso normal de eventos	
Acción de los Actores	Respuesta del sistema
<ol style="list-style-type: none"> Una vez cambiados los valores de una variable del proceso, se registra la información en la plataforma Web. Cuando el subproceso termina su ejecución llama a la plataforma Web informándole del evento. 	<ol style="list-style-type: none"> La aplicación Web recibe la información de la variable y la registra en la plataforma. La aplicación Web recibe la información y retorna la información del siguiente subproceso, en la ejecución, para que se informe al siguiente dispositivo en la ejecución que debe iniciar.

Tabla A9 Caso de uso en formato expandido Sincronizar.

A1.1.3 Diagrama de casos de uso Cliente Móvil MobFlow

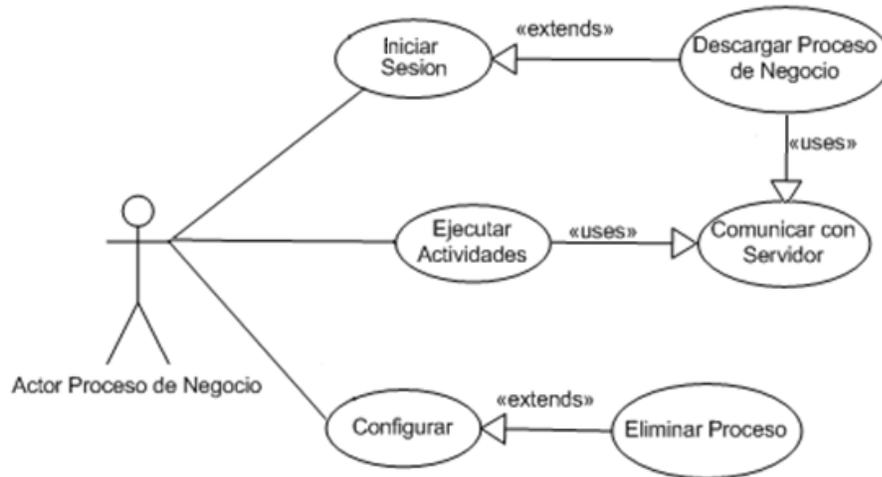


Figura A2. Diagrama de casos de cliente móvil MobFlow.

En la figura A2, se presentan los diagramas de casos de uso para la aplicación cliente móvil MobFlow.

A1.1.4 Casos de uso en formato expandido MobFlow

Caso de uso:	Iniciar Sesión (Autenticación)	
Actores:	Actor Proceso de Negocio	
Propósito:	Iniciar sesión en la plataforma Web.	
Resumen:	En este caso de uso, el usuario actor ingresa sus datos en el sistema móvil de información, para iniciar sesión en la aplicación Web MobFlow.	
Tipo:	Primario.	
Curso normal de eventos		
Acción de los Actores	Respuesta del sistema	
1. Este caso de uso se inicia cuando un usuario quiere ingresar al sistema. 3. El usuario llena los campos Nombre de Usuario y Contraseña para realizar su ingreso. 4. El usuario selecciona aceptar.	2. El sistema presenta una interfaz de usuario con las opciones requeridas para iniciar sesión. 5. El sistema presenta una interfaz de usuario mostrándole el proceso correspondiente, y brinda las opciones para la descarga o no del proceso.	
Curso alternativo 1 – línea 5		
	6. El sistema no encuentra la información del usuario y presenta un mensaje de Error	

Tabla A10. Caso de uso en formato expandido Iniciar Sesión

Caso de uso:	Descargar Proceso de Negocio	
Actores:	Actor Proceso de Negocio	
Propósito:	Obtener el respectivo proceso de negocio	
Resumen:	El usuario, actor de determinado proceso de negocio, accede a la plataforma, una vez validados sus datos y obtiene su respectivo proceso de negocio descrito en BPEL, junto con el archivo WSDL del proceso de negocio en general. Este proceso queda almacenado de manera persistente en el dispositivo móvil.	
Tipo:	Primario.	
Curso normal de eventos		
Acción de los Actores	Respuesta del Sistema	
<ol style="list-style-type: none"> 1. Este caso de uso inicia luego que el usuario ha iniciado sesión en el sistema. 2. El usuario presiona aceptar para iniciar la descarga de los procesos correspondientes. 3. El usuario selecciona aceptar. 	<ol style="list-style-type: none"> 4. El sistema inicia la descarga de los procesos de negocio (BPEL y WSDL). 5. El sistema almacena de manera persistente los archivos descargados. 	

Tabla A11. Caso de uso en formato expandido Descargar Proceso de Negocio.

Caso de uso:	Ejecutar Actividades	
Actores:	Actor Proceso de Negocio	
Propósito:	Inicia la ejecución de su respectivo proceso de negocio en el dispositivo móvil	
Resumen:	Este caso de uso permite al usuario, realizar las actividades correspondientes a la ejecución del proceso de negocio BPEL (sus tareas del proceso de negocio), a partir del proceso descargado.	
Tipo:	Primario.	
Curso normal de eventos		
Acción de los Actores	Respuesta del Sistema	
<ol style="list-style-type: none"> 1. Este caso de uso empieza cuando el usuario selecciona iniciar la ejecución del proceso. 3. El usuario ingresa los datos, según la interfaz de usuario generada por el sistema, que corresponde a sus tareas. 	<ol style="list-style-type: none"> 2. El sistema genera una interfaz gráfica de acuerdo al subproceso generado 4. El sistema procesa la información ingresada por el usuario, y envía registro a la plataforma Web. 	

Tabla A12. Caso de uso en formato expandido Ejecutar Actividades.

Caso de uso:	Comunicar con Servidor:	
Actores:	Actor Proceso de Negocio	
Propósito:	Conectar aplicación móvil con la aplicación Web	
Resumen:	Este caso de uso permite realizar una conexión y comunicación con el servidor donde está alojada la aplicación Web MobFlow.	
Tipo:	Primario.	
Curso normal de eventos		
Acción de los Actores	Respuesta del Sistema	
	<ol style="list-style-type: none"> 1. El sistema establece que se requiere una conexión con la aplicación Web. 2. Obtiene los parámetros necesarios según el tipo de petición requerida. 3. Realiza la petición con el servidor Web y espera por una respuesta. 4. Procesa la respuesta obtenida desde el servidor y continúa con la ejecución en caso de que sea necesario. 	

Tabla A13. Caso de uso en formato expandido Comunicar con Servidor.

Caso de uso:	Eliminar Proceso	
Actores:	Actor Proceso de Negocio	
Propósito:	Eliminar el subproceso almacenado en el dispositivo móvil.	
Resumen:	Mediante este caso de uso, el usuario actor del proceso de negocio elimina el subproceso de negocio almacenado en el dispositivo móvil.	
Tipo:	Secundario	
Curso normal de eventos		
Acción de los Actores	Respuesta del Sistema	
<ol style="list-style-type: none"> 1. Este caso de uso inicia cuando el usuario selecciona eliminar un proceso. 3. El usuario presiona la opción aceptar para eliminara el proceso. 	<ol style="list-style-type: none"> 2. El sistema presenta una interfaz para confirmar la eliminación del subproceso. 4. El sistema elimina el subproceso del dispositivo móvil. 	

Tabla A14. Caso de uso en formato expandido Eliminar Proceso.

A1.2 Modelo Conceptual Preliminar

A partir de los requisitos, se identificaron las entidades proceso de negocio, subprocesso de negocio, actores, monitoreo, sincronizar subprocesso. Así mismo, se definió las relaciones como se muestra en el siguiente diagrama conceptual presentado en la figura A.3:

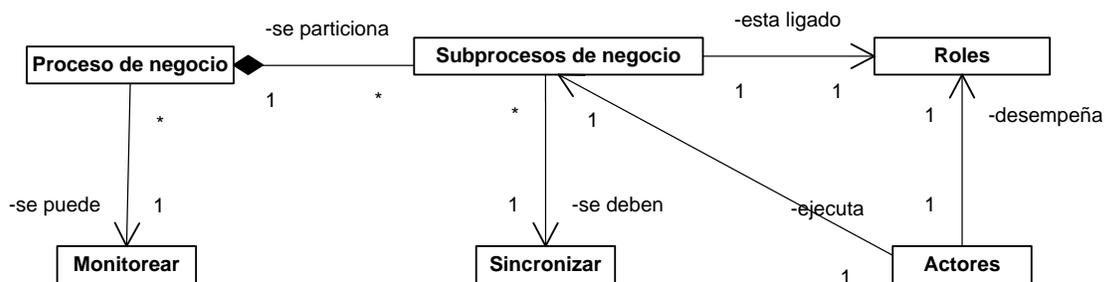


Figura A.3. Diagrama conceptual para la gestión de procesos, para la aplicación Web MobFlow.

A2. FASE DE CONSTRUCCIÓN

A2. 1. Diagramas de Secuencias

A continuación se presentan los diagramas de secuencia para las dos herramientas software desarrolladas en este proyecto.

A2.1.1 Diagramas de secuencias MobFlow

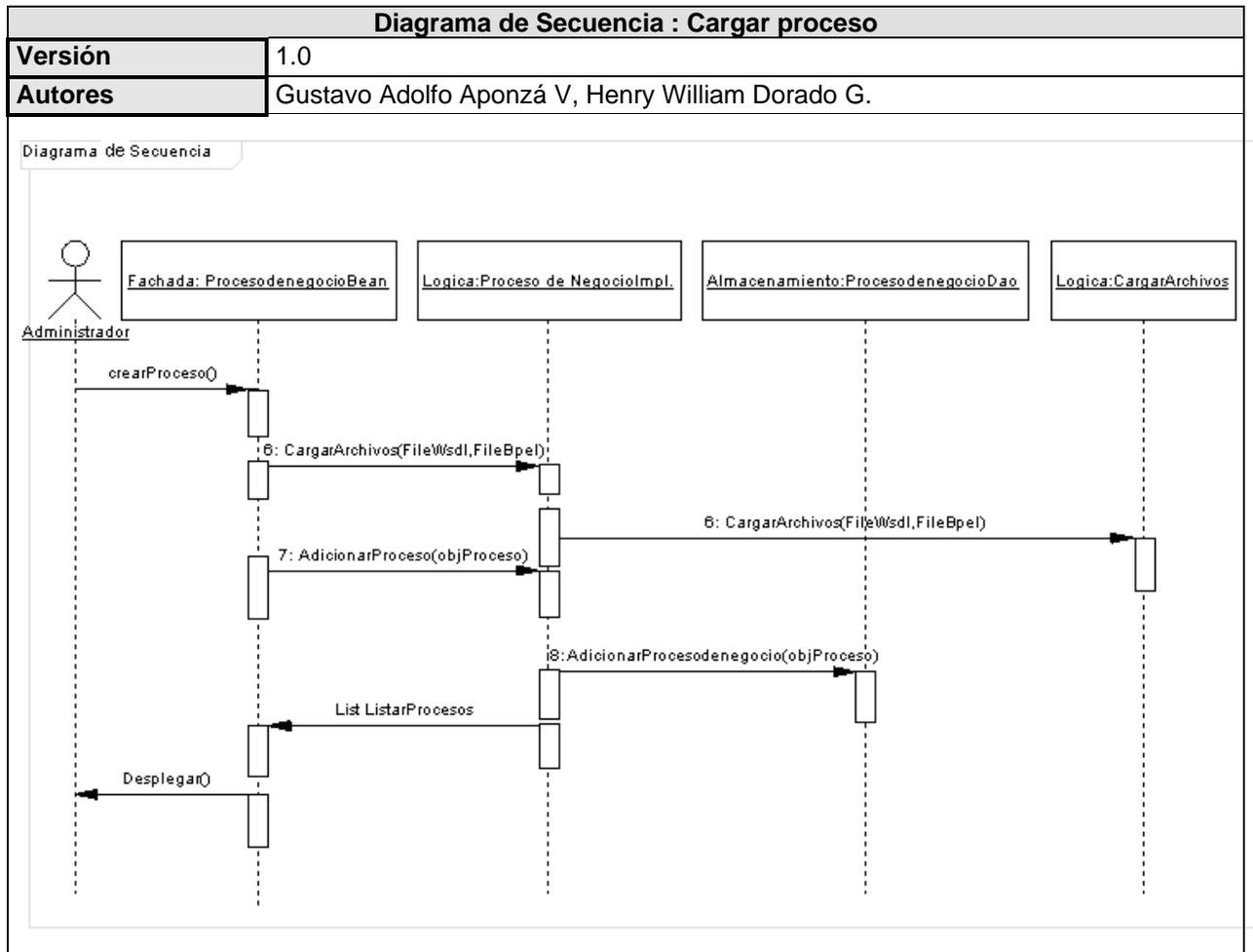


Tabla A15. Diagrama de Secuencia Cargar Proceso

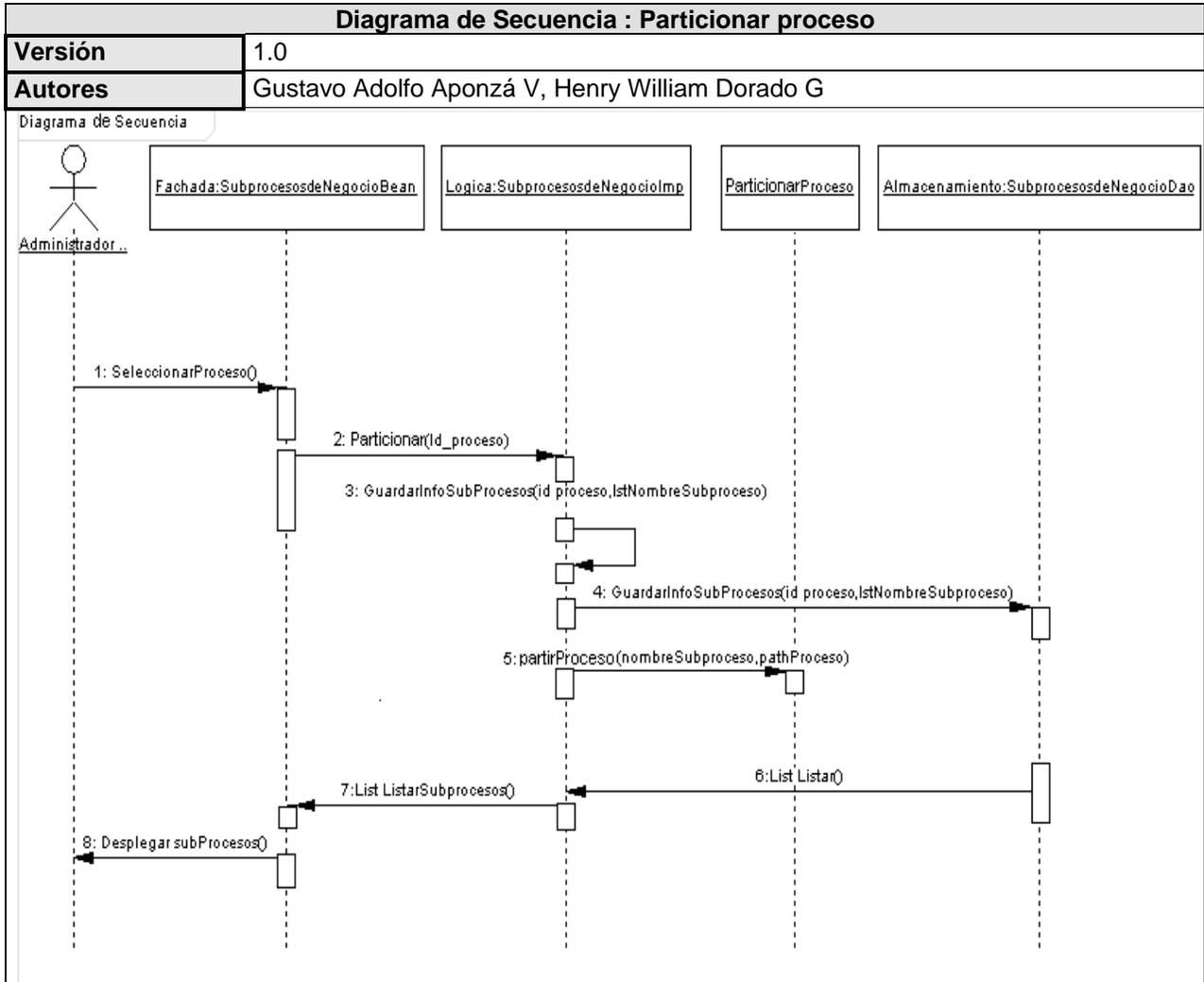


Tabla A16. Diagrama de Secuencia Particionar Proceso.

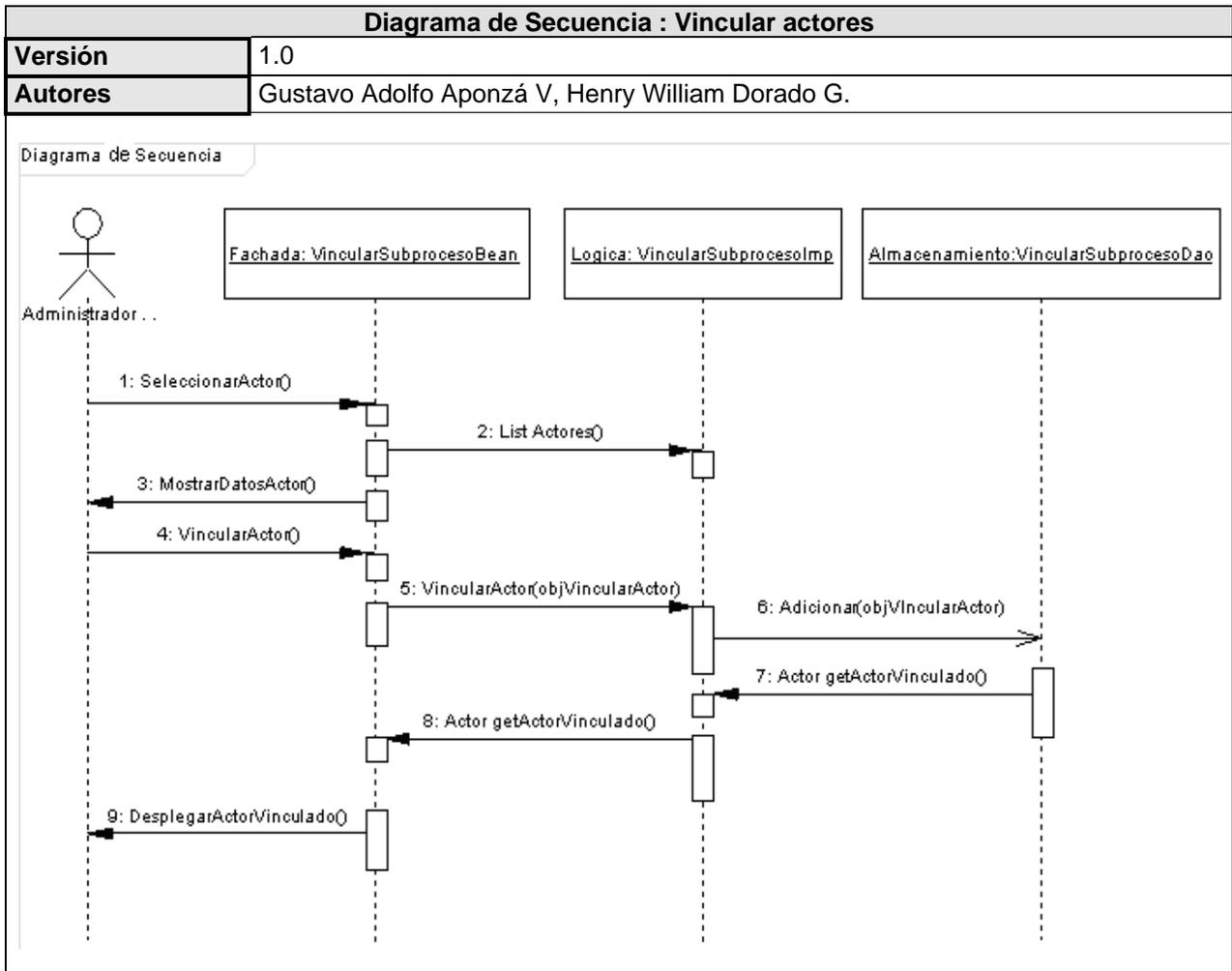


Tabla A17. Diagrama de Secuencia Vincular actores.

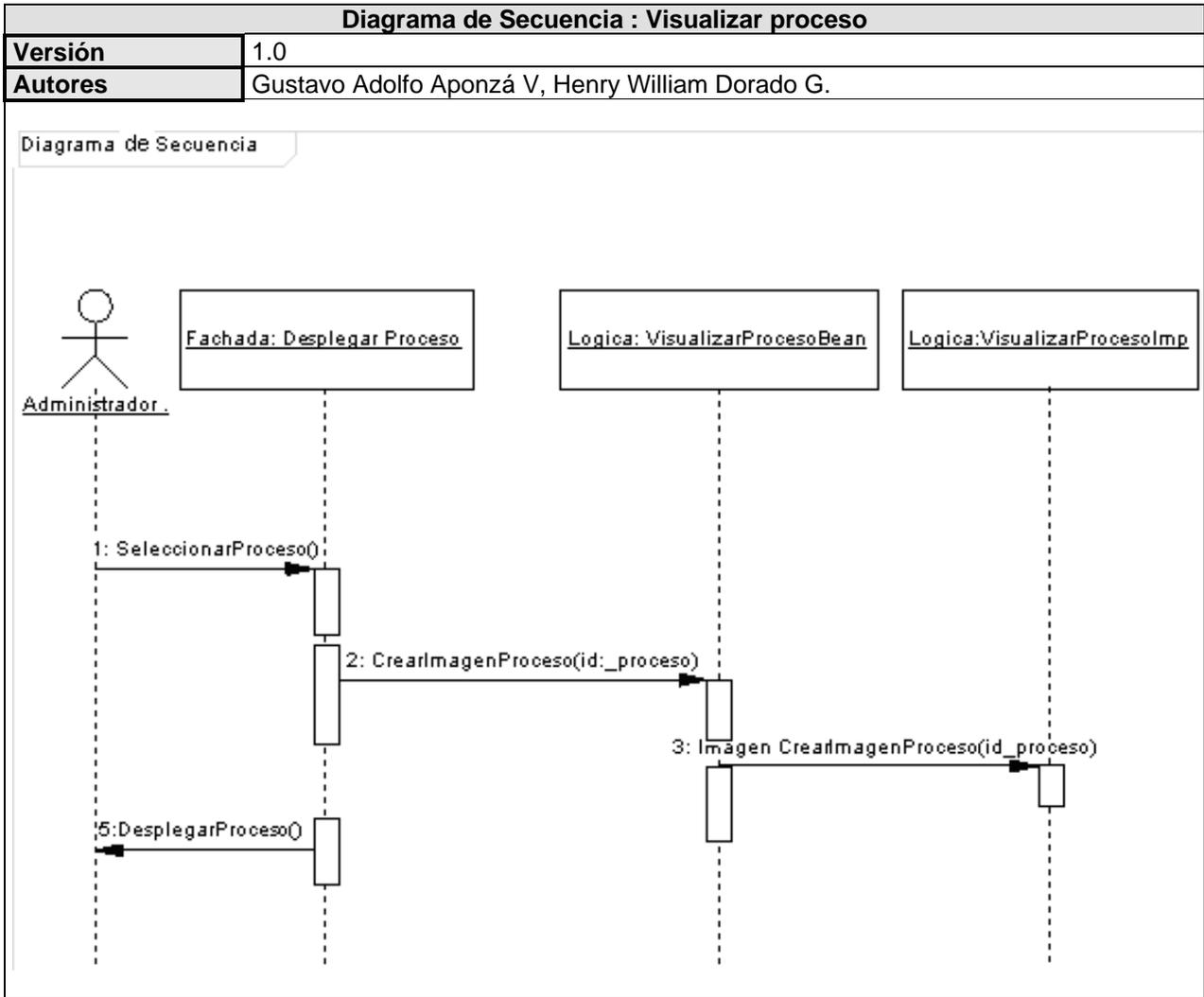


Tabla A18. Diagrama de Secuencia Visualizar Proceso.

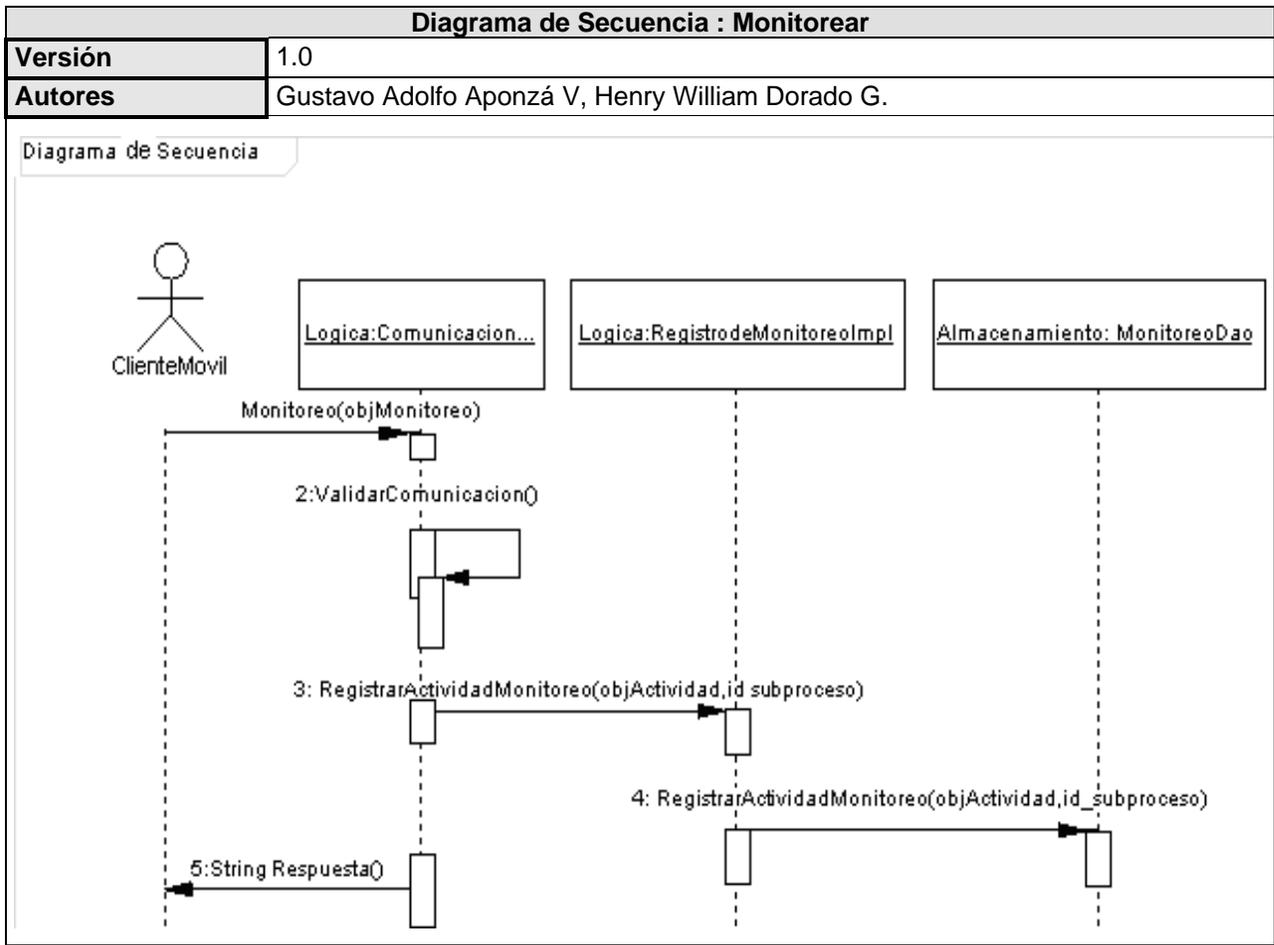


Tabla A19. Diagrama de Secuencia Monitorear.

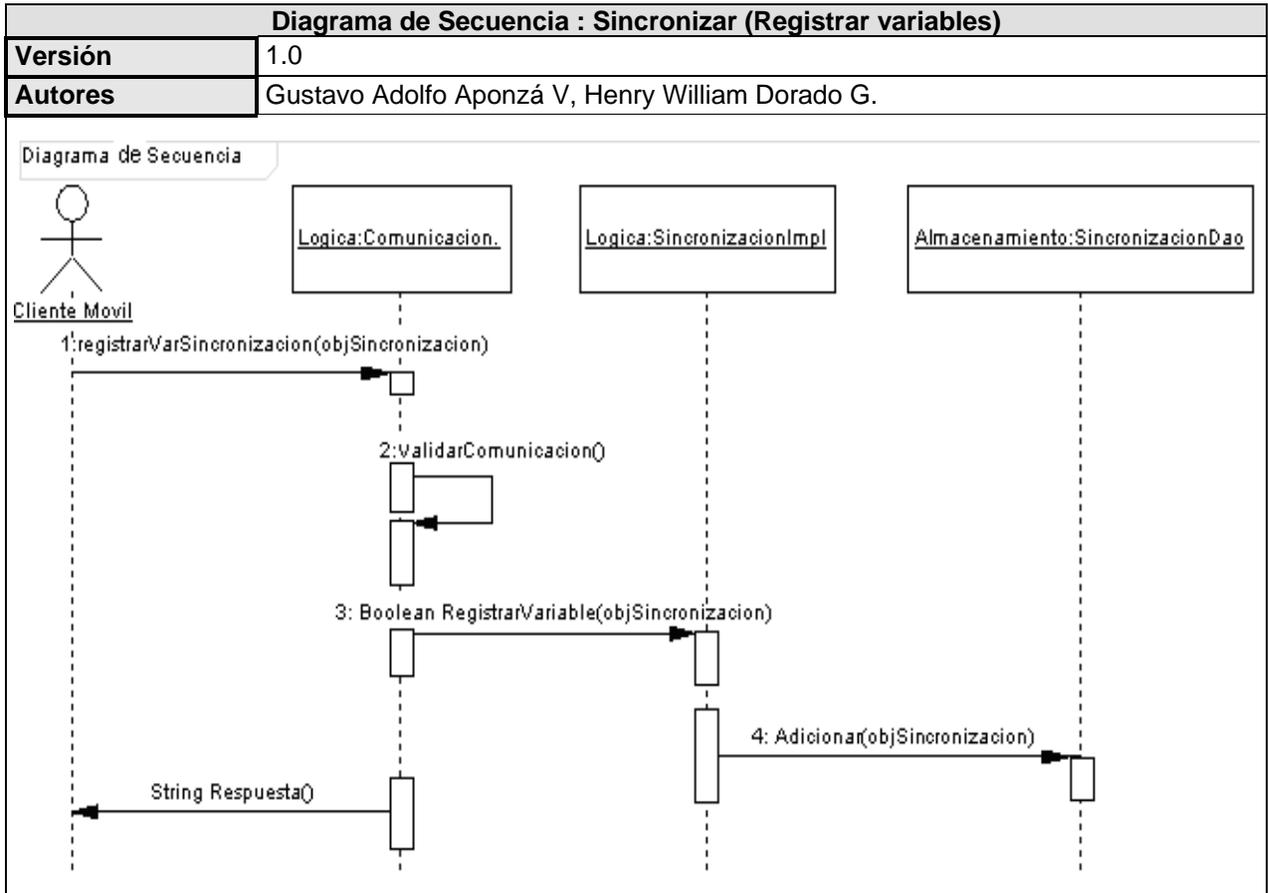


Tabla A20. Diagrama de Secuencia, Sincronizar (Registrar variables).

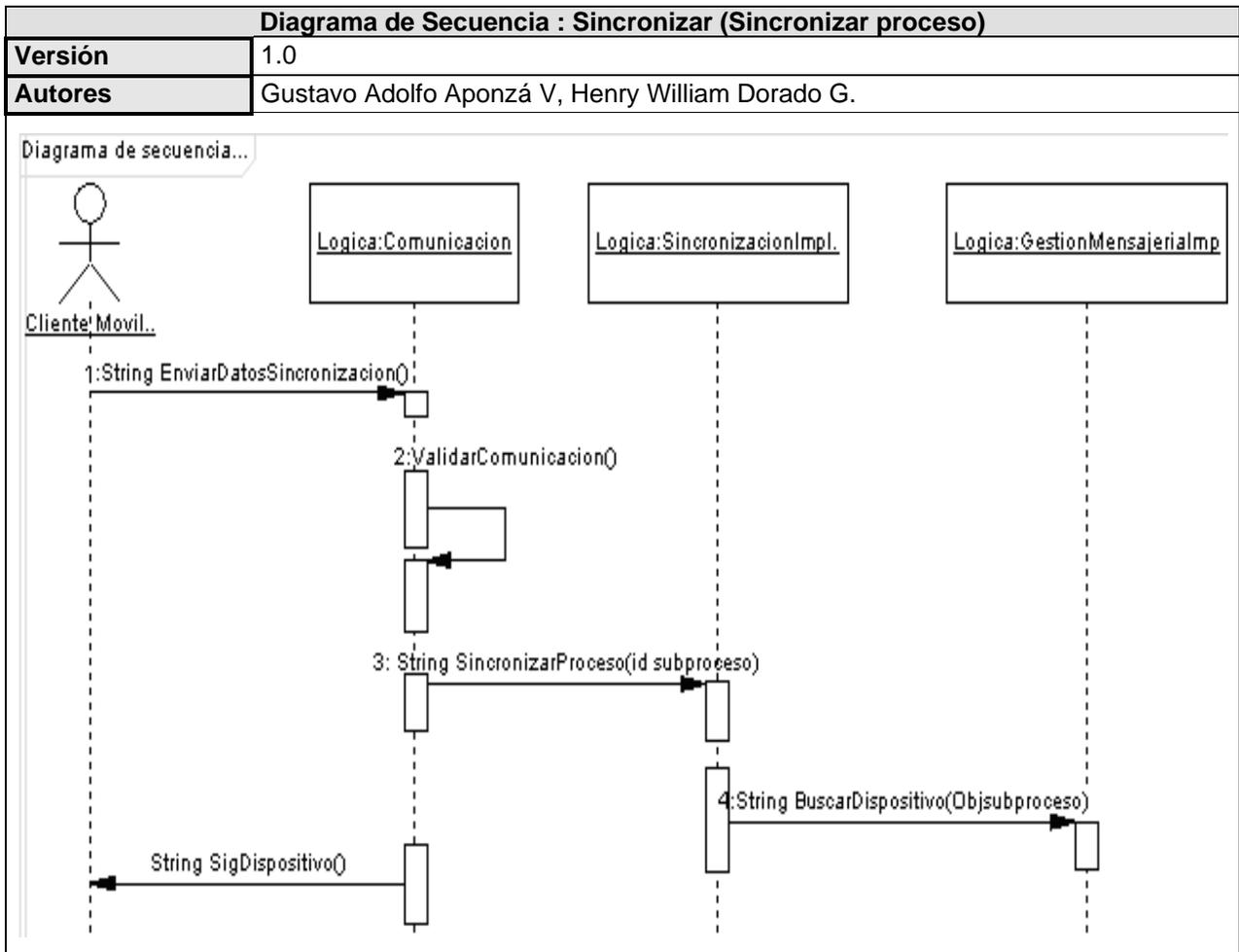


Tabla A21. Diagrama de Secuencia, Sincronizar (Sincronizar proceso).

A2.1.2 Diagramas de secuencias Cliente Móvil MobFlow

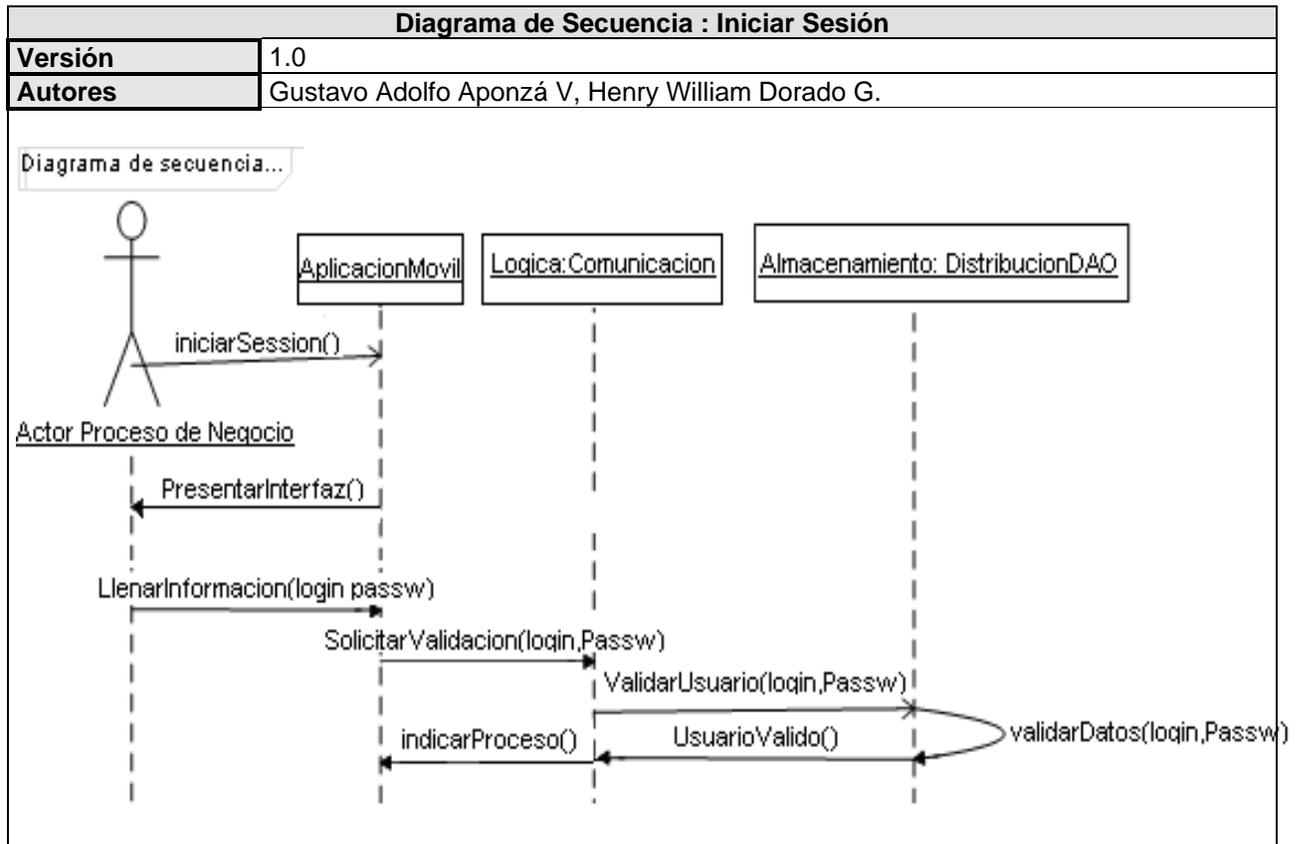


Tabla A22. Diagrama de Secuencia Iniciar Sesión Cliente Móvil MobFlow

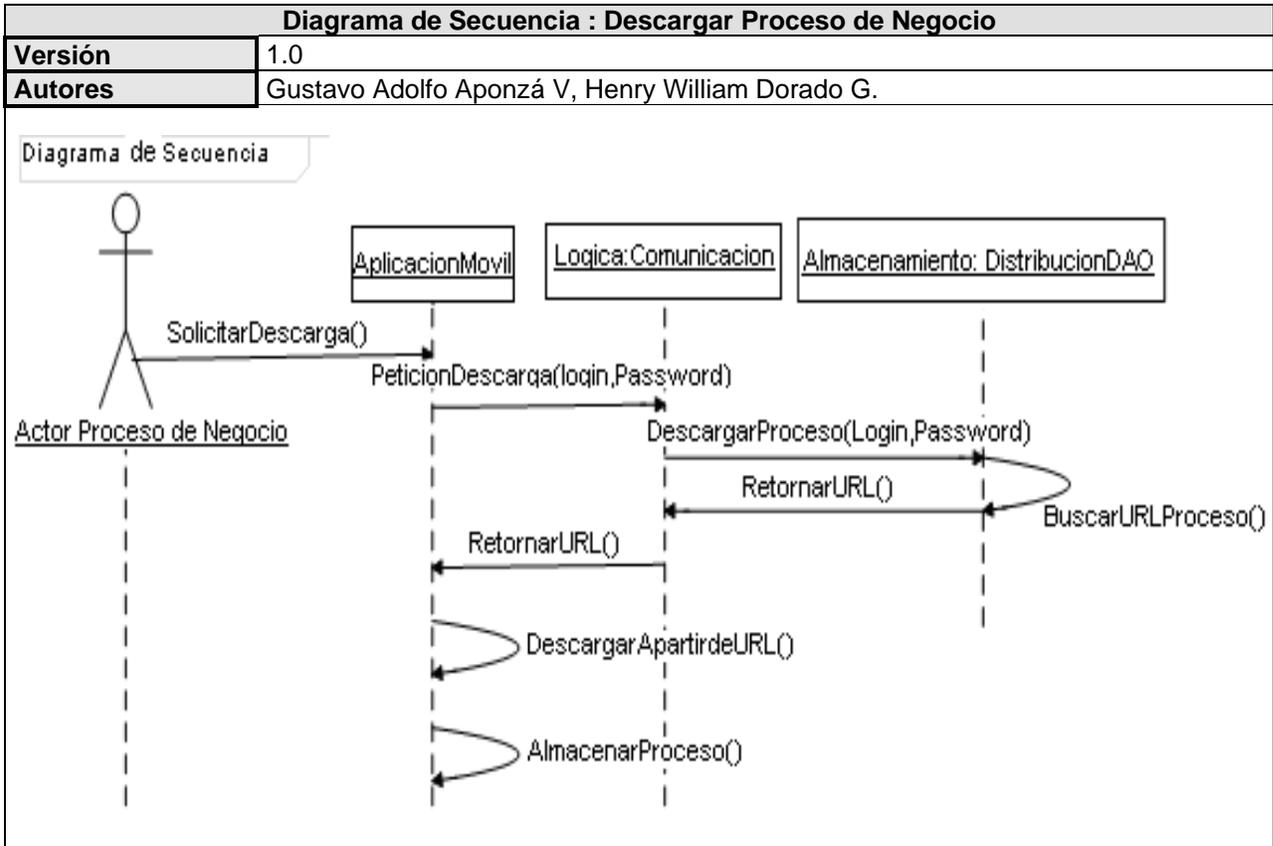


Tabla A23. Diagrama de Descargar Proceso de Negocio

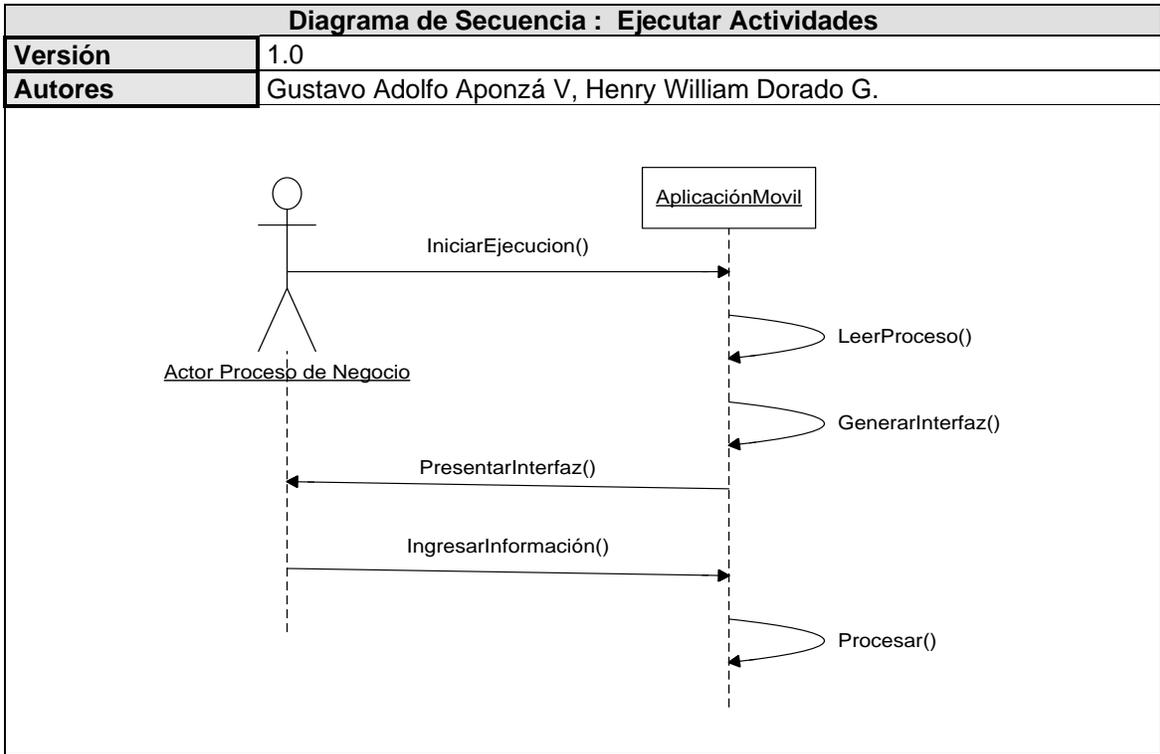


Tabla A24. Diagrama de Secuencia Ejecutar Actividades.

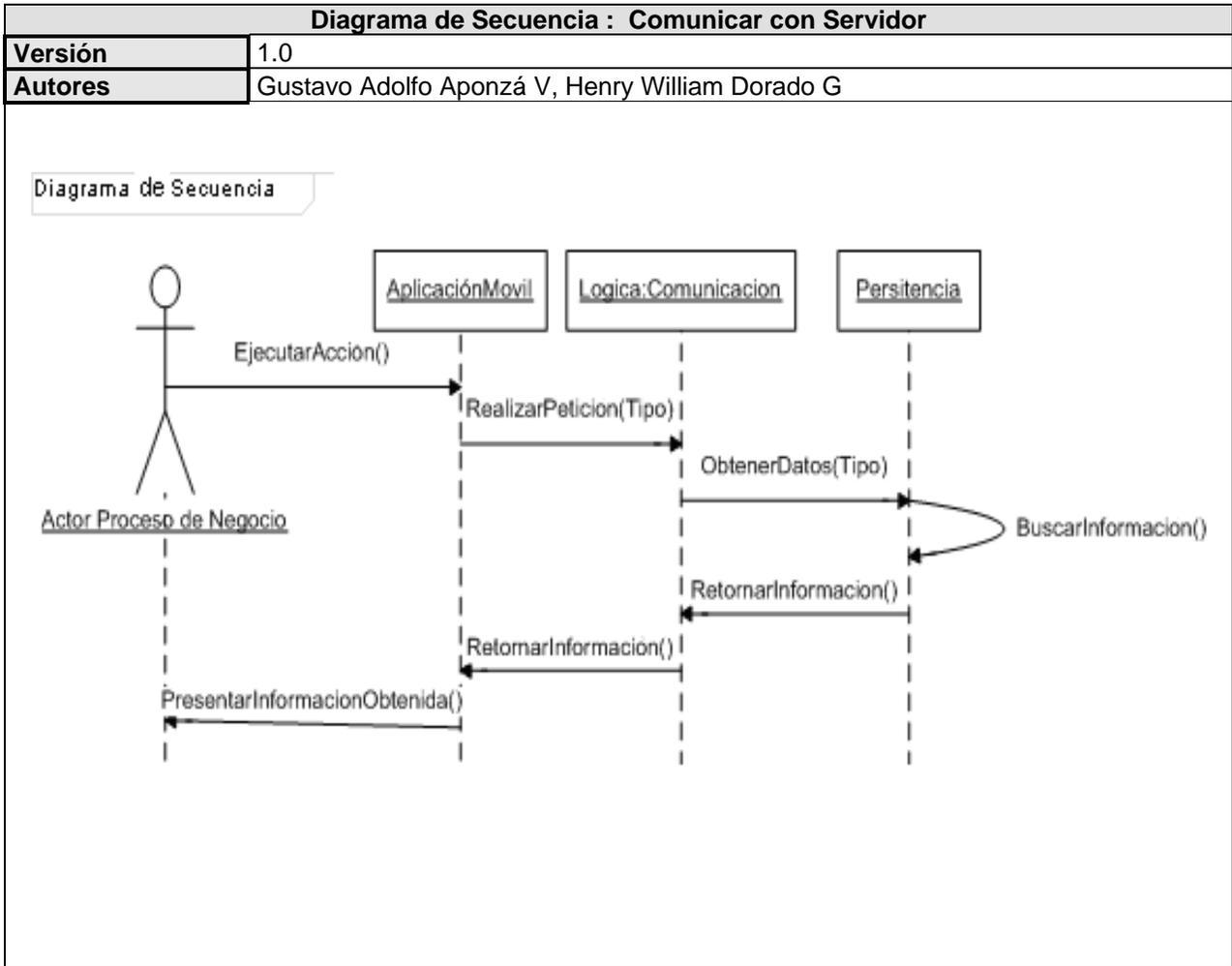


Tabla A25. Diagrama de Secuencia Comunicar con Servidor

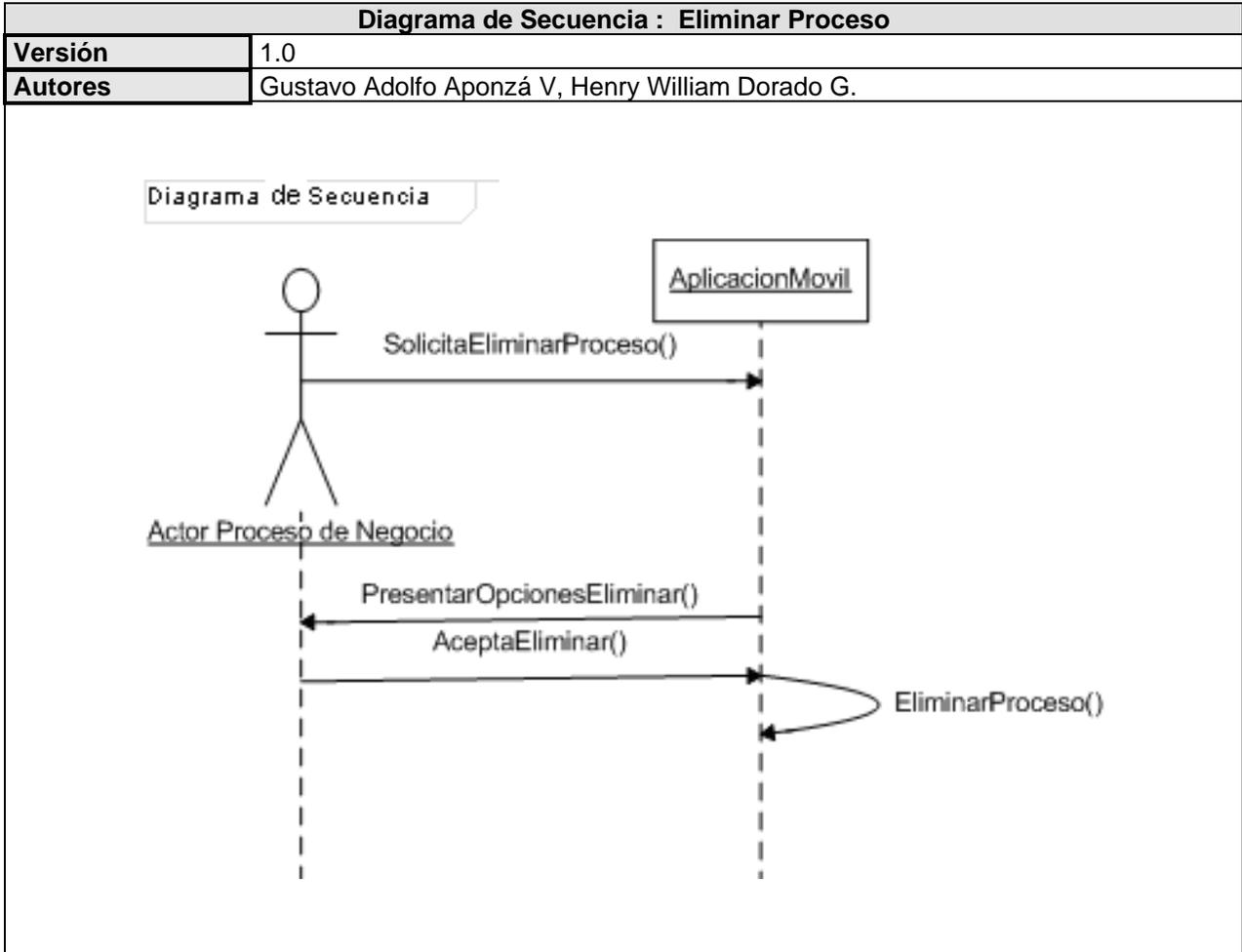


Tabla A26. Diagrama de Secuencia Eliminar Proceso

A2. 2. Diagramas de Casos de Uso Reales

A2.2.1 Diagramas de casos de uso reales MobFlow

A continuación se presentan los casos de uso reales para la herramienta Web de la plataforma.



Figura A4. Interfaz Inicio de sesión.

Caso de uso:	Iniciar Sesión (Autenticación)	
Actores:	Usuario (Iniciador).	
Propósito:	Ingresar al sistema.	
Resumen:	La persona ingresar en el sistema para realizar las diferentes tareas disponibles que brinda la plataforma.	
Tipo:	Primario y real.	
Curso normal de eventos		
Acción de los Actores	Respuesta del sistema	
<p>1. Este caso de uso se inicia cuando un usuario quiere ingresar al sistema.</p> <p>2. El usuario selecciona (1 y 2) de la figura A4 y llena los campos Nick y Password para realizar su ingreso.</p> <p>3. El usuario presiona el botón (3) de la figura A4 para ingresar.</p>	<p>4. El sistema presenta la interfaz inicial de MobFlow, al nuevo usuario para que pueda hacer uso de los diferentes servicios que ofrece la plataforma.</p>	

Tabla A27. Caso de uso Real iniciar sesión.

Cargar Proceso de Negocio	
Nombre	1 <input type="text"/>
Descripcion	2 <input type="text"/>
Documento WSDL	3 <input type="button" value="Seleccionar archivo"/> No se h... archivo
Documento BPEL	4 <input type="button" value="Seleccionar archivo"/> No se h... archivo
	5 <input type="button" value="Enviar"/>

PROCESO DE NEGOCIO			
NOMBRE	DESCRIPCION	ARCHIVO	ELIMINAR
Prueba	Proceso de prueba	Prueba.bpel	<input type="button" value="X"/>

Figura A5. Interfaz para cargar los procesos de negocio.

Caso de uso:	Cargar proceso de negocio.	
Actores:	Administrador (Iniciador).	
Propósito:	Cargar los procesos de negocio en la plataforma Web.	
Resumen:	Esta interfaz, permite cargar los procesos de negocio representados por un documento BPEL y un archivo WSDL, además de permitir ingresar la información del nombre y la descripción del proceso.	
Tipo:	Primario y real.	
Curso normal de eventos		
Acción de los Actores	Respuesta del sistema	
<p>1. Este caso de uso se inicia cuando un usuario Administrador quiere cargar un proceso de negocio, que se definió previamente.</p> <p>2. El usuario selecciona (1) de la figura A5 y llena el campo nombre, que corresponde a la nombre del proceso de negocio.</p> <p>3. El usuario selecciona (2) de la figura A5 y llena el campo Descripción, que corresponde a la descripción del proceso.</p> <p>4. El usuario selecciona los archivos BPEL, y WSDL del proceso de negocio.</p> <p>5. El usuario selecciona (5) de la figura A5 para cargar el proceso de negocio.</p>	<p>6. El sistema carga la información de proceso de negocio en la base de datos y almacena el documento BPEL y WSDL correspondiente.</p>	

	7. El sistema despliega la información del proceso en la tabla (6) que muestra la figura A5.
Curso alternativo 1 – Línea 4: Selección de archivos	
Acción de los Actores	Respuesta del sistema
4.1. El usuario presiona el botón (3) de la Figura A5 y realiza la búsqueda del recurso que quiere subir (documento BPEL).	4.2 El sistema carga la ruta del documento BPEL que se quiere subir.
5.1. El usuario presiona el botón (4) de la Figura A5 y realiza la búsqueda del recurso que quiere subir (documento WSDL).	5.2 El sistema carga la ruta del documento WSDL que se quiere subir.

Tabla A28. Caso de uso Cargar proceso de negocio.



Figura A6. Interfaz para particionar los procesos de negocio.

Caso de uso:	Particionar proceso de negocio.
Actores:	Administrador (Iniciador).
Propósito:	Lleva a cabo el particionamiento de los procesos de negocio, en subprocesos sincronizados y guardar la información en la base de datos de la plataforma.
Resumen:	Al iniciar el particionamiento del proceso previamente cargado, la plataforma crea uno a uno los documentos BPEL correspondientes a los subprocesos de negocio sincronizados. Una vez terminado el particionamiento se muestra la información en pantalla.
Tipo:	Primario y real.
Curso normal de eventos	
Acción de los Actores	Respuesta del sistema
1. Este caso de uso se inicia cuando un usuario quiere particionar el proceso de negocio cargado previamente.	

<p>2. El usuario selecciona en (1) de la figura A6, para dar inicio al proceso de particionamiento.</p>	<p>3. El sistema particiona el proceso de negocio en documentos BPEL sincronizados, y almacena la información de los subprocesos obtenidos.</p> <p>4. El sistema despliega una ventana con los datos de los subprocesos obtenidos como se muestra en la figura A6.</p>
---	--

Tabla A29. Caso real de uso Particionar proceso de negocio.

Vincular Actor	
Rol del Actor:	Rol Actor
Actor: 1	<input type="text" value="actor"/>
Nombres:	<input type="text" value="nombres del actor"/>
Apellidos:	<input type="text" value="apellidos del actor"/>
Cedula:	<input type="text" value="5423135453"/>
Celular: 2	<input type="text" value="300159889"/>
3	<input type="button" value="Adicionar"/>

SUBPROCESOS DE NEGOCIO

NOMBRES
ROL
CELULAR
ELIMINAR

4

Figura A7. Interfaz para vincular los actores de los procesos de negocio.

Caso de uso:	Vincular Actores.
Actores:	Administrador (Iniciador).
Propósito:	Vincular los actores que ejecutan los subproceso de negocios generados por la plataforma.
Resumen:	Una vez particionado el proceso de negocio, se procede a vincular los actores que ejecutaran cada subproceso, para ellos se seleccionan los subprocesos generados y se vinculan los actores dependiendo del rol que desempeñen en la ejecución del proceso.
Tipo:	Primario y real.
Curso normal de eventos	
Acción de los Actores	Respuesta del sistema
<p>1. Este caso de uso se inicia, cuando un usuario va a vincular un subproceso de negocio que fue particionado.</p> <p>2. El usuario selecciona en la tabla el subproceso que va a vincular. Esta interfaz de nuestro en (2) de la figura A6.</p>	<p>3. El sistema despliega una nueva ventana con las opciones necesarias para vincular el actor del subproceso, esta interfaz se muestra en la</p>

<p>4. El usuario selecciona el actor encargado de ejecutar el subproceso de negocio en (1) de la figura A7.</p> <p>5. El usuario selecciona (2) de la figura A7 y llena el campo numero de dispositivo, que corresponde a el dispositivo que ejecutara el subproceso de negocio.</p> <p>6. El usuario presiona el botón (5) de la figura A7, para vincular el actor del subproceso de negocio seleccionado.</p>	<p>figura A7.</p> <p>7. El sistema despliega en la tabla (4) de la figura A7, la información del actor y dispositivo vinculado.</p>
---	---

Tabla A30. Caso real de uso Vincular actores.

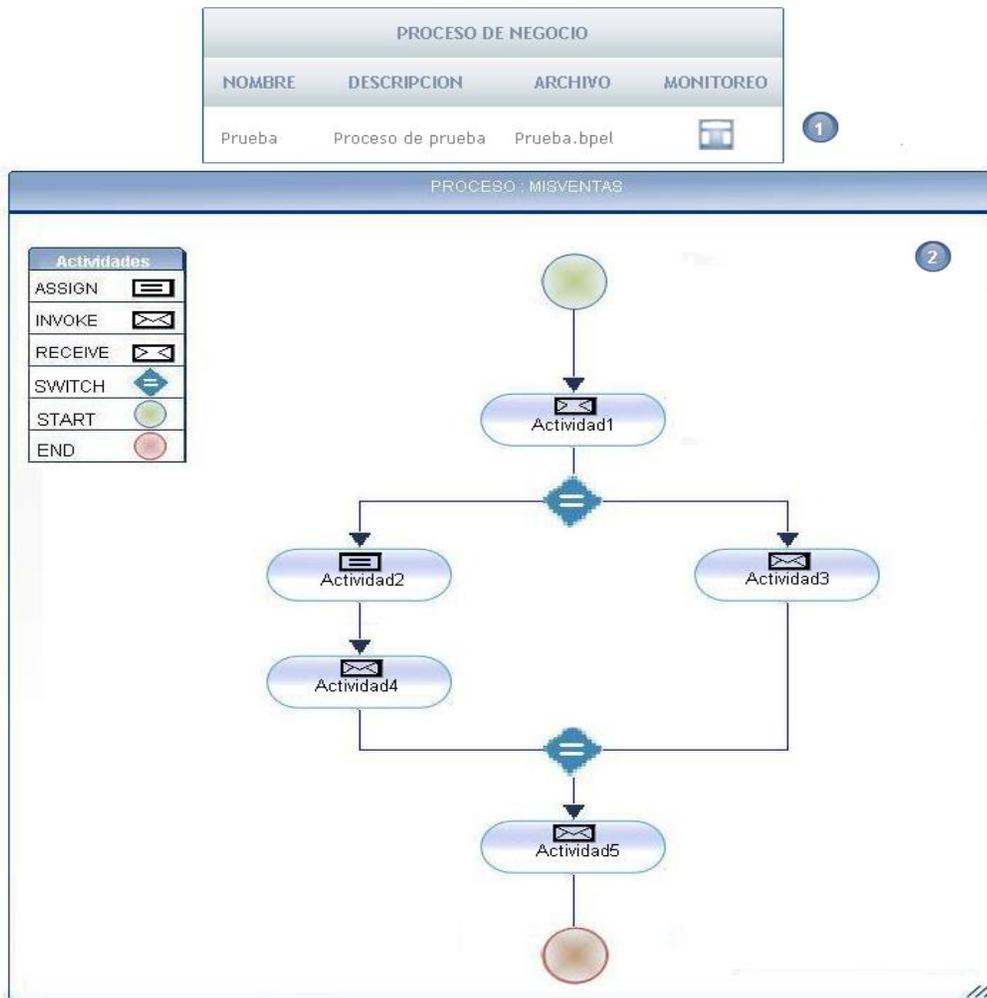


Figura A8. Interfaz para la Visualización del proceso previamente cargado.

Caso de uso:	Visualizar el proceso de negocio.	
Actores:	Administrador (Iniciador).	
Propósito:	Visualiza un proceso de negocio cargado.	
Resumen:	Después de cargar el documento BPEL con la descripción del proceso de negocio, la página de visualización muestra una descripción gráfica del proceso.	
Tipo:	Primario y real.	
Curso normal de eventos		
Acción de los Actores	Respuesta del sistema	
<p>1. Este caso de uso se inicia cuando un usuario va a visualizar un proceso de negocio que ha sido cargado previamente.</p> <p>2. El usuario selecciona el icono (1) el cual se muestra en la figura A8, el cual da inicio al proceso de despliegue.</p>	<p>3. El sistema despliega una ventana con, la tabla de procesos y la imagen del proceso seleccionado, esta imagen se muestra en (2) de la figura A8.</p>	

Tabla A31. Caso real de uso visualizar el proceso de negocio.

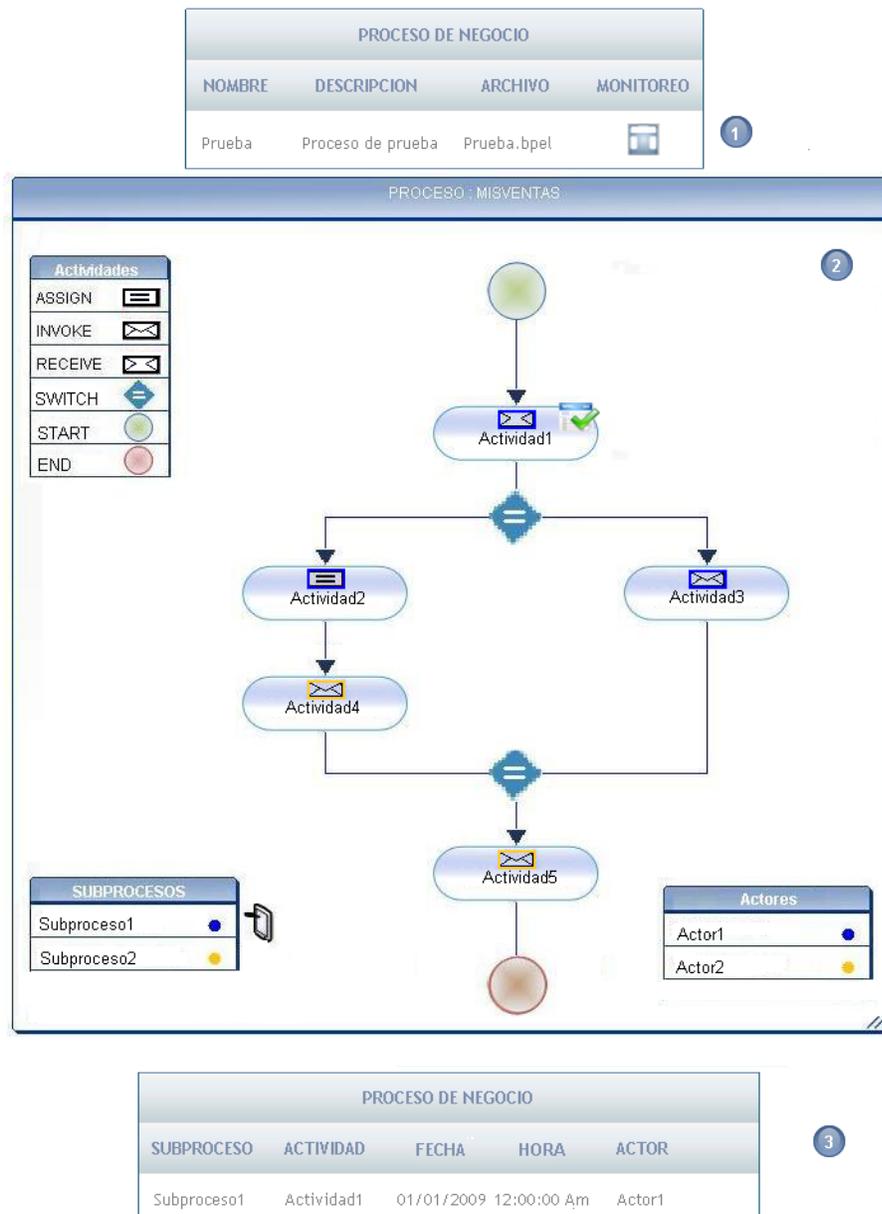


Figura A9. Interfaz para el Monitoreo del proceso de negocio.

Caso de uso:	Monitoreo de los procesos de negocio.
Actores:	Administrador (Iniciador).
Propósito:	Monitoreo de la ejecución de un proceso de negocio.
Resumen:	Permite monitorear la ejecución de los procesos de negocio, para realizar dicha acción se tuvo que haber cargado, particionado, configurado y ejecutado el proceso de negocio, posteriormente se muestra en la interfaz de monitoreo el estado de la ejecución de cada proceso.
Tipo:	Primario y real.
Curso normal de eventos	

Acción de los Actores	Respuesta del sistema
<p>1. Este caso de uso se inicia, cuando un usuario va a monitorear la ejecución de un proceso de negocio que ha sido ejecutado previamente.</p> <p>2. El usuario selecciona el icono (1) que corresponde al subproceso que quiere monitorear, el cual se muestra en la figura A9.</p>	<p>3. El sistema despliega una ventana, en esta interfaz se muestra un grafico con el proceso de negocio y las actividades que se han ejecutado las cuales aparecen marcadas, como se muestra en (2) de la figura A9, además de una tabla con la información de la ejecución del proceso en donde se encuentran los datos como: actividades que se ejecutan, el actor, el subproceso, el dispositivo y la fecha y hora en que fueron ejecutadas dichas actividades en los dispositivos móviles, como se muestran en la tabla (3) de la figura A9.</p>

Tabla A32. Caso real de uso monitoreo de procesos de negocio.

A2.2.2 Diagramas de casos de uso reales Cliente Móvil MobFlow

A continuación se presentan los casos de uso reales para la aplicación móvil de la plataforma.

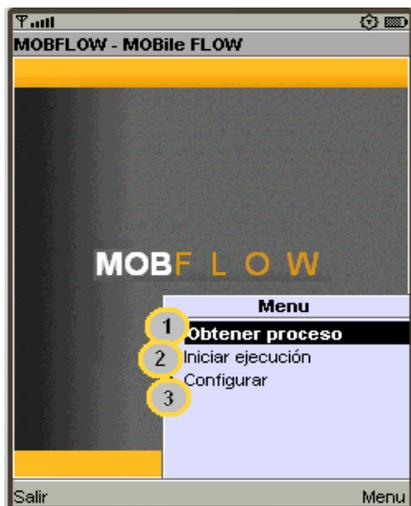


Figura A10. Menú principal aplicación móvil.

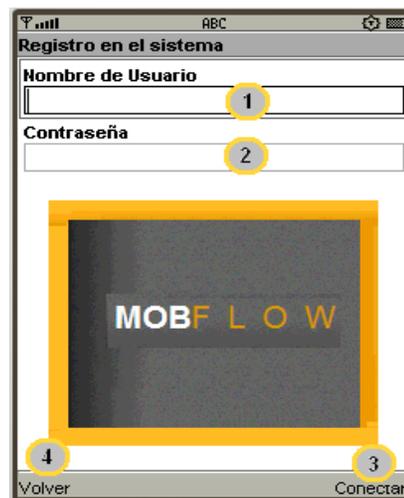


Figura A11. Inicio de sesión desde aplicación móvil.

Caso de uso:	Iniciar Sesión (Autenticación)
Actores:	Actor Proceso de Negocio
Propósito:	Iniciar sesión en la plataforma Web.
Resumen:	En este caso de uso, el usuario actor ingresa sus datos en el sistema móvil de información para iniciar sesión en la aplicación Web MobFlow.
Tipo:	Primario.
Curso normal de eventos	
Acción de los Actores	Respuesta del sistema

<p>1. Este caso de uso el usuario selecciona la opción (1) del menú principal (figura A10).</p> <p>3. El usuario llena los campos Nombre de Usuario opción (1) figura A11 y Contraseña opción (2) figura 11 para realizar su ingreso.</p> <p>4. El usuario selecciona el botón Conectar opción (3) figura A11.</p>	<p>2. El sistema presenta una interfaz de usuario con las opciones requeridas para iniciar sesión (figura A11).</p> <p>5. El sistema presenta una interfaz de usuario mostrándole el proceso correspondiente a inicial de MobFlow, y brinda las opciones para la descarga o no de los subproceso.</p>
Curso alternativo 1 – línea 5	
	<p>6. El sistema no encuentra la información del usuario y presenta un mensaje de Error</p>

Tabla A32. Caso real de uso Iniciar Sesión Cliente Móvil MobFlow.

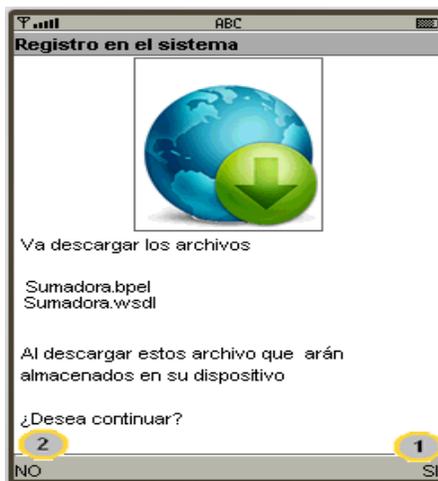


Figura A12. Interfaz gráfica para descarga de proceso

Caso de uso:	Descargar Proceso de Negocio
Actores:	Actor Proceso de Negocio
Propósito:	Obtener el respectivo proceso de negocio
Resumen:	Con este caso de uso, el usuario actor de determinado proceso de negocio accede a la plataforma, una vez validados sus datos y obtiene su respectivo proceso de negocio descrito en BPEL, junto con el archivo WSDL del proceso de negocio en general. Este proceso queda almacenado de manera persistente en el dispositivo móvil.
Tipo:	Primario.
Curso normal de eventos	
Acción de los Actores	Respuesta del Sistema

<p>1. Este caso de uso inicia luego que el usuario ha iniciado sesión en el sistema.</p> <p>2. El usuario presiona el botón (1) figura A12 para iniciar la descarga de los procesos correspondientes.</p>	<p>3. El sistema inicia la descarga de los procesos de negocio (BPEL y WSDL).</p> <p>4. El sistema almacena de manera persistente los archivos descargados.</p>
---	---

Tabla A33. Caso real de uso Descargar Proceso de Negocio.



Figura A13. Interfaz para ejecución de actividades

Caso de uso:	Ejecutar Actividades	
Actores:	Actor Proceso de Negocio	
Propósito:	Inicia la ejecución de su respectivo proceso de negocio en el dispositivo móvil	
Resumen:	Este caso de uso permite al usuario, realizar las actividades correspondientes a la ejecución del proceso de negocio BPEL (sus tareas del proceso de negocio), a partir del proceso descargado.	
Tipo:	Primario.	
Curso normal de eventos		
Acción de los Actores		Respuesta del Sistema
<p>1. Este caso de uso inicia cuando el usuario selecciona iniciar la ejecución del proceso, para esto presiona el botón (2) de la figura A10.</p> <p>3. El usuario ingresa los datos según la interfaz de usuario generada por el sistema, que corresponde a sus tareas (campos (1) para este caso)</p> <p>4. El usuario presiona el botón (2) de la figura A13 para el envío de la información.</p>		<p>2. El sistema genera una interfaz gráfica de acuerdo al subproceso generado</p> <p>5. El sistema procesa la información ingresada por el usuario y envía registro a la plataforma Web.</p>

Tabla A34. Caso real de uso Ejecutar Actividades.

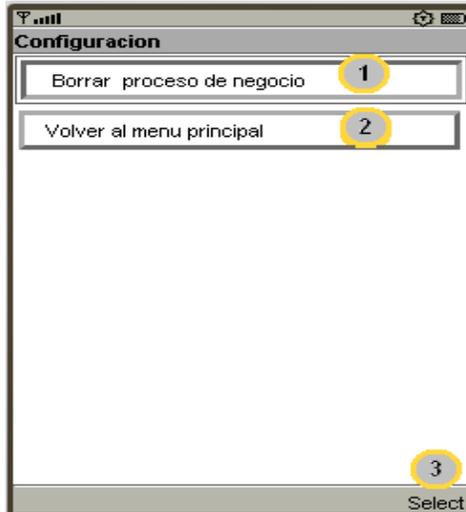


Figura A14. Menu de configuración.

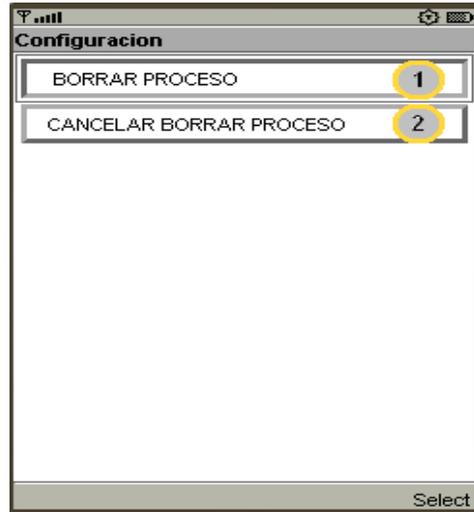


Figura A15. Interfaz para eliminar un proceso.

Caso de uso:	Eliminar Proceso
Actores:	Actor Proceso de Negocio
Propósito:	Eliminar el subproceso almacenado en el dispositivo móvil.
Resumen:	Mediante este caso de uso, el usuario actor del proceso de negocio elimina el subproceso de negocio almacenado en el dispositivo móvil.
Tipo:	Secundario
Curso normal de eventos	
Acción de los Actores	Respuesta del Sistema
<p>1. Este caso de uso inicia cuando el usuario selecciona el botón configurar (3) de la figura A10</p> <p>3. El usuario selecciona el botón Borrar proceso de negocio (1) figura A14.</p> <p>4. El usuario presiona el botón (1) figura A15 para eliminar el proceso.</p>	<p>2. El sistema presenta una interfaz con las opciones de configuración figura A14.</p> <p>4. El sistema presenta una interfaz para confirmar la eliminación figura A15.</p> <p>5. El sistema elimina el subproceso del dispositivo móvil.</p>

Tabla A35. Caso real de uso Eliminar Proceso.

A2. 3. Diagramas de Clases y de Paquetes

A2.3.1. Diagrama de clases y paquetes MobFlow

Diagrama de Clases MobFlow

El diagrama de clase de la figura A16 presenta las clases implementadas en la aplicación Web MobFlow. El diseño de la aplicación se basa en dos patrones de diseño: el patrón *Session Facade* y el patrón *Objeto de Acceso a Datos* (DAO) y la aplicación Web se implementa utilizando la tecnología de Java Servlet Faces, por lo que sigue el Modelo Vista-Controlador.

Las clases cuyos nombres terminan en **-Bean**, implementan la lógica de las interfaces de usuario, encargadas de brindar las funciones permitidas a los usuarios dentro de la plataforma.

Las clases con nombres que terminan en **-Impl**, representan las diferentes clases que manejan la lógica de la aplicación para funcionalidades como monitoreo, particionamiento entre otras. Estas clases a su vez se relacionan con las clases de lógica de presentación y persistencia de datos.

Finalmente, las clases de persistencia que contienen los datos obtenidos de una fuente de datos, o los datos que serán almacenados en una fuente de datos, están definidas con el nombre de la tabla de la base de datos.

El uso de estas clases se realiza en conjunción con las clases que hace las implementaciones concretas para el acceso a una fuente de datos determinada, el nombre de estas clases terminan en **-Dao**.

Por ejemplo la clase **ProcesodeNegocioImpl**, se asocia con una clase **ProcesodeNegocio** y **ProcesodeNegocioDao** para el acceso a datos, con esto se logra una separación entre la lógica de la aplicación y el acceso a datos.

Estas clases asociadas no son presentadas en el diagrama por razones de espacio y legibilidad.

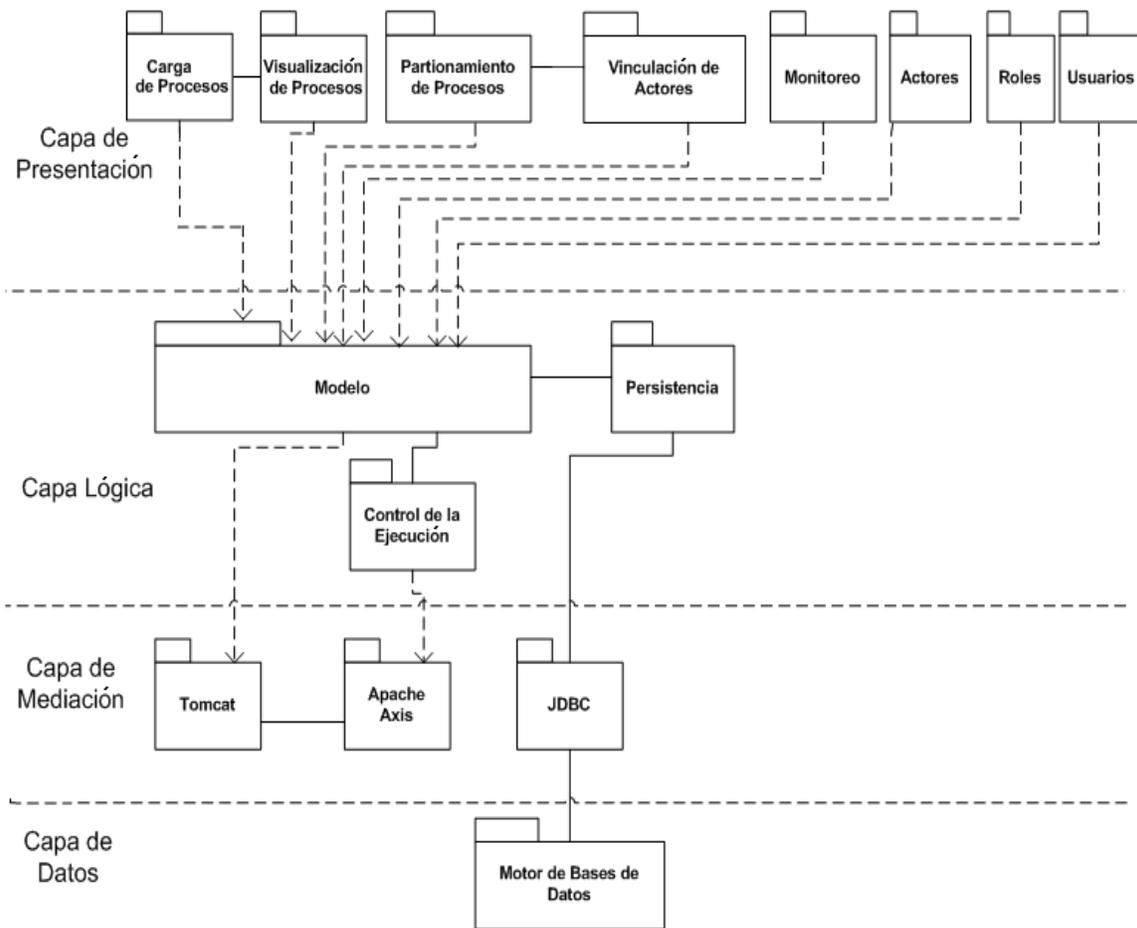


Figura A17. Diagrama de paquetes general de la aplicación Web MobFlow.

- **Capa de Presentación**
 - Carga de Procesos

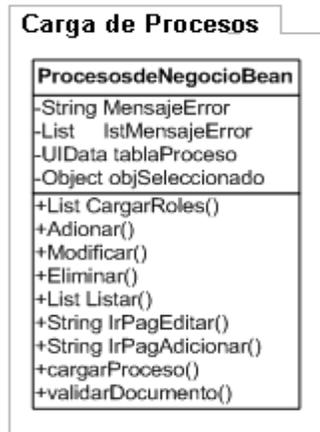


Figura A18. Paquete Carga de Procesos.

Este paquete (figura A18) contiene las clases de lógica de presentación encargadas de gestionar los procesos de negocio. Compuesto por la siguiente clase:

- ✓ *ProcesodeNegocioBean*: Esta clase presenta una interfaz gráfica de usuario para realizar la gestión de Procesos de Negocio, con esta se puede cargar, listar, y eliminar los documentos de los procesos BPEL.

- Visualización de Procesos:



Figura A19. Paquete Visualización de Procesos.

Este paquete (figura A19) contiene las clases de lógica de presentación encargadas de mostrar de forma visual los procesos de negocio. Compuesto por la siguiente clase:

- ✓ *VisualizarProcesoBean*: Esta clase permite visualizar los procesos de negocio cargados en la plataforma.

- Particionamiento de procesos:

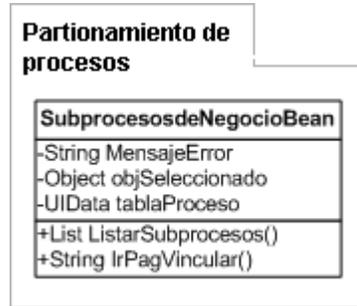


Figura A20. Paquete Particionamiento de procesos.

Este paquete (figura A20) contiene las clases de lógica de presentación encargadas de particionar los procesos de negocio, está compuesto por la siguiente clase:

- ✓ *SubProcesosdeNegocioBean*: Esta clase implementa las funcionalidades de mostrar los SubProcesos generados en el particionamiento.

- Vinculación de actores:



Figura A21. Paquete Vinculación de actores.

Este paquete (figura A21) contiene las clases de lógica de presentación encargadas de vincular los actores con los subprocesos generados en el particionamiento, contiene la siguiente clase:

- ✓ *VincularSubprocesoBean*: Esta clase implementa una interfaz gráfica de usuario que permite vincular los actores con los subprocesos generados.

- Monitoreo:

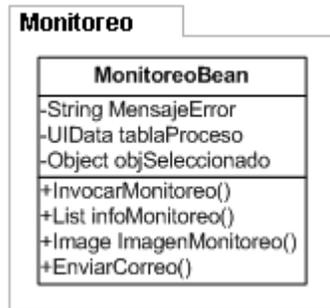


Figura A22. Paquete Monitoreo.

Este paquete (figura A22) contiene las clases de lógica de presentación encargadas de mostrar el monitoreo de los procesos ejecutados, contiene la siguiente clase:

- ✓ *MonitoreoBean*: Esta clase permite administrar las funcionalidades básicas de visualización y monitoreo de los procesos en ejecución.

- Actores:

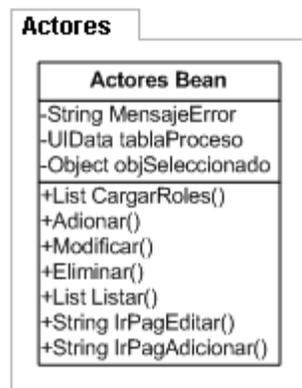


Figura A23. Paquete Actores.

Este paquete (figura A23) contiene las clases de lógica de presentación encargadas de manejar la información de los actores, contiene la siguiente clase:

- ✓ *ActoresBean*: Esta clase implementa una interfaz gráfica de usuario para realizar las funcionalidades básicas de crear, editar, consultar y eliminar los actores de los procesos de negocio.

- Roles:

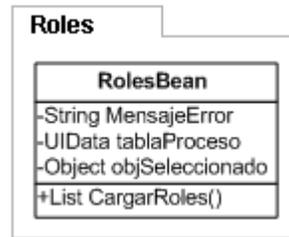


Figura A24. Paquete Roles.

Este paquete (figura A24) contiene las clases de lógica de presentación encargadas de manejar la información de los Roles, contiene la siguiente clase:

- ✓ *RolesBean*: Esta clase implementa una interfaz gráfica de usuario para realizar las funcionalidades básicas consultar y editar los roles de los procesos de negocio.

- Usuarios:

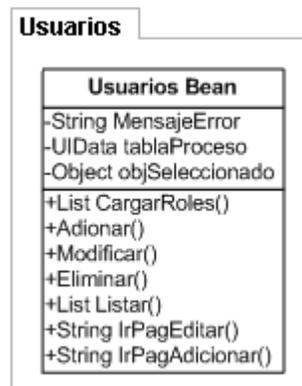


Figura A26. Paquete Usuarios.

Este paquete (figura A26) contiene las clases de lógica de presentación encargadas de manejar la información de los usuarios de la plataforma, contiene la siguiente clase:

- ✓ *UsuariosBean*: Esta clase implementa una interfaz gráfica de usuario para realizar las funcionalidades básicas de crear, editar, consultar y eliminar los usuarios de la plataforma.

- **Capa Lógica**

- Modelo:

Este paquete (figura A27) contiene toda la lógica que permite realizar las funciones relacionadas con los procesos de negocio, como son la visualización de estos, el particionamiento de procesos y la vinculación de los actores con los subprocesos generados en el particionamiento. Las clases contenidas en este paquete son:

- ✓ *ProcesodeNegocioImpl*: Clase encargada de realizar la carga de archivos del proceso, guardar la información del proceso, consultar y llamar a otras clases para desplegar, monitorear y particionar los procesos de negocio.

- ✓ *CargarArchivos*: Clase que implementa la lógica necesaria para cargar (subir) archivos a la plataforma y almacenarlo en un directorio específico.
- ✓ *VisualizarProcesoImpl*: Clase encargada de generar una imagen de monitoreo o visualización de los procesos de negocio cargados en la plataforma.
- ✓ *TransformadorBpelGrafos*: Encargada de transformar un documento BPEL en una notación de grafos. Esta clase es de vital importancia ya que facilita los procesos de visualización, particionamiento y validación de procesos de negocio.
- ✓ *ParticionarProceso*: Encargada de manejar la lógica de la aplicación para el particionamiento de procesos de negocio en subprocesos de negocio, utilizando para ello las clases *gestionarDocXml* y *TransformadorBpelGrafos*.
- ✓ *GestionarDocXml*: Esta clase contiene las funciones que permiten la manipulación de documentos Xml.
- ✓ *SubprocesosdenegocioImpl*: Esta clase se encarga de la lógica para manejar la información de los subprocesos particionados.
- ✓ *VincularSubprocesoImpl*: Encargada de manejar la lógica para la vinculación de los actores a cada subproceso generado.
- ✓ *MonitoreoImpl*: Encargada de manejar la lógica para obtener la información del estado de los procesos en ejecución.
- ✓ *RolesImpl*: Esta clase se encarga de la lógica para manejar la información de los roles de los procesos.
- ✓ *ActorImpl*: Esta clase se encarga de la lógica para manejar la información de los actores de los procesos.
- ✓ *UsuariosImpl*: Esta clase se encarga de la lógica para manejar la información de los usuarios de la plataforma Web.

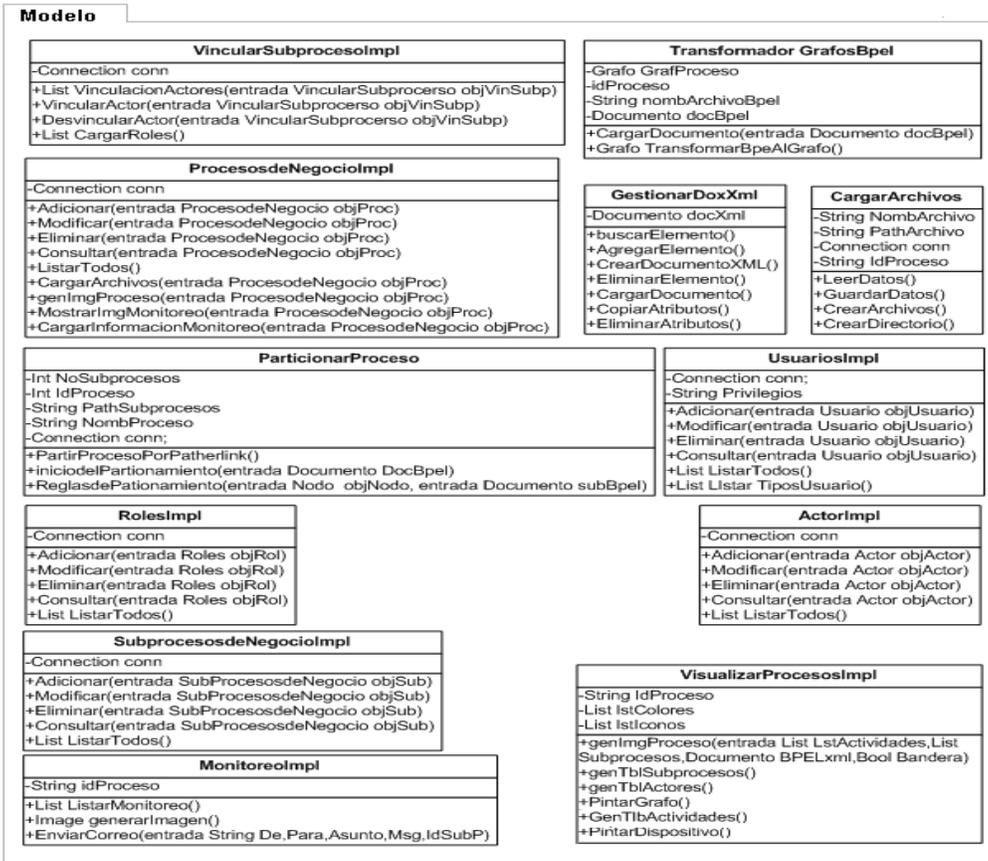


Figura A27. Paquete Modelo.

○ Control de la Ejecución:

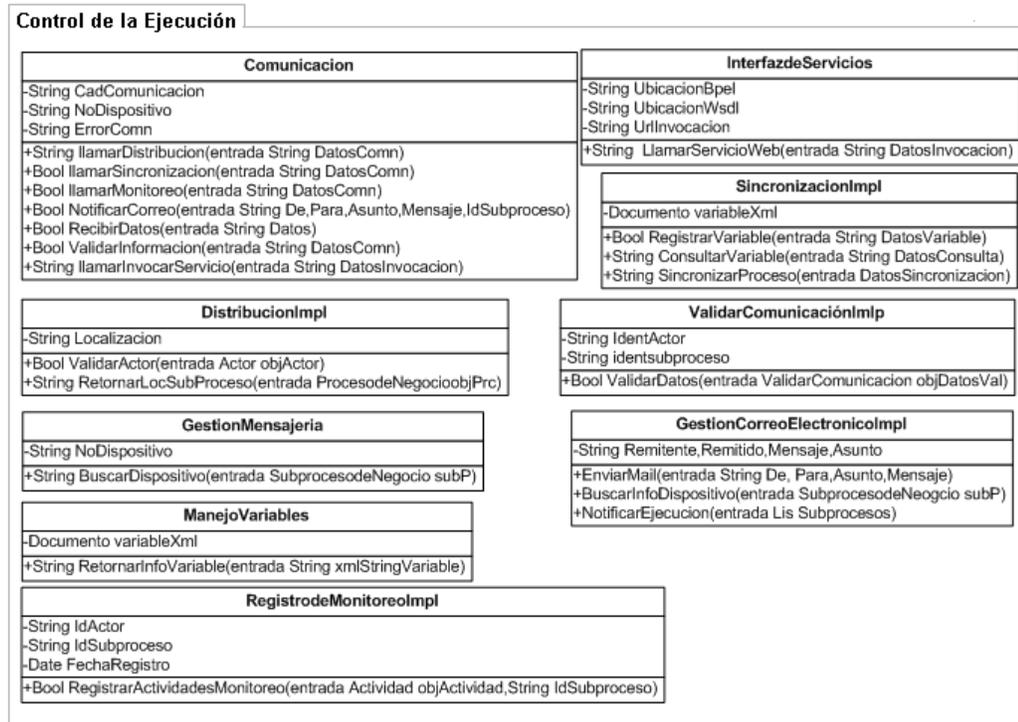


Figura A28. Paquete, Control de la Ejecución.

Este paquete (figura A28) contiene, las clases que permiten interactuar entre el dispositivo móvil y la plataforma. Las funciones involucradas son: control de la sincronización de la ejecución de procesos, control del monitoreo de procesos, distribución de subprocesos a los dispositivos móviles y llamado dinámico a servicios Web, las clases que contiene este paquete son:

- ✓ *Comunicacion*: Esta clase permite la comunicación entre la aplicación para dispositivo móvil y la aplicación Web. Esta se comunica con las otras clases de este paquete según la petición enviada por el dispositivo (sincronización, monitoreo, distribución, ejecución de servicios Web).
- ✓ *ValidarComunicaciónImpl*: Por medio de esta clase, se valida si el actor que realiza determinada petición desde el sistema móvil de información es un usuario registrado en la plataforma. Con esta validación, también se obtiene el identificador del proceso de negocio al cual está asociado el subproceso ejecutado por el actor que realiza la petición, y se lo comunica a las otras clases de este paquete a través de la clase *Comunicacion*. De esta manera las otras clases realizan sus funciones de acuerdo al proceso de negocio identificado, lo que permite manejar concurrencia.
- ✓ *SincronizaciónImpl*: Encargada de manejar la lógica de la aplicación para controlar la ejecución de los subprocesos de negocio, con el fin de gestionar aspectos de la sincronización de ejecución. Para lograr este propósito esta clase registra y consulta los datos de sincronización (variables) recibidos a través de la clase *Comunicacion*.
- ✓ *ManejoVariables*: Esta clase se encarga de manejar la información de las variables utilizadas en la ejecución de los subprocesos de negocio.

- ✓ *GestionMensajeria*: Cuando un dispositivo móvil termina la ejecución de un proceso de negocio, es necesario indicarle al sistema móvil de información que continua con la ejecución del proceso. Esta clase obtiene el número de teléfono del siguiente dispositivo involucrado en la ejecución de un proceso.
 - ✓ *RegistrodeMonitoreoImpl*: Clase que permite registrar la información de las actividades ejecutadas en los dispositivos móviles, por medio de los datos recibidos de la clase de *Comunicacion*.
 - ✓ *DistribucionImpl*: Encargada de distribuir por medio de la clase de *Comunicacion* los subprocesos generados a los correspondientes actores en sus sistemas móviles de información.
 - ✓ *InterfazdeServicios*: Encargada de realiza la invocación de los servicios Web requeridos en la ejecución de los procesos de negocio en los sistemas móviles de información. Esta clase captura la información enviada por estos y la transforma a un formato apropiado para la invocación. Finalmente retorna el resultado de esta invocación en un formato comprensible por el motor BPEL móvil.
 - ✓ *GestionCorreoElectronicoImpl*: Luego de realizado el particionamiento se encarga de obtener la información del correo electrónico de los actores que se acaban de asociar al proceso particionado y envía un correo electrónico que indica que pueden descargar el subproceso.
- Persistencia: Este paquete contiene las clases encargadas de manejar el acceso a la base de datos.
- ✓ *ProcesodeNegocio*: Esta clase representa datos persistentes para almacenar la información del Proceso de negocio.
 - ✓ *ProcesodeNegocioDAO*: Esta clase se usa en conjunto con la clase *ProcesodeNegocio*, y hace las implementaciones concretas para el acceso a la fuente de datos.
 - ✓ *VisualizarProceso*: Esta clase representa datos persistentes para almacenar la información requerida para visualizar el Proceso de negocio
 - ✓ *VisualizarProcesoDAO*: Esta clase se usa en conjunto con la clase *VisualizarProceso*, y hace las implementaciones concretas para el acceso a la fuente de datos.
 - ✓ *ValidarComunicacion*: Esta clase representa datos persistentes para almacenar la información requerida para Validar la Comunicación.
 - ✓ *ValidarComunicacionDAO*: Esta clase se usa en conjunto con la clase *ValidarComunicacion*, y hace las implementaciones concretas para el acceso a la fuente de datos.
 - ✓ *Distribucion*: Esta clase representa datos persistentes para almacenar la información requerida para la Distribución.
 - ✓ *DistribucionDAO*: Esta clase se usa en conjunto con la clase *Distribucion*, y hace las implementaciones concretas para el acceso a la fuente de datos.

- ✓ *RegistrodeMonitoreo*: Esta clase representa datos persistentes para almacenar la información requerida para el Monitoreo.
- ✓ *RegistrodeMonitoreoDAO*: Esta clase se usa en conjunto con la clase *RegistrodeMonitoreo*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- ✓ *Sincronizacion*: Esta clase representa datos persistentes para almacenar la información requerida para la sincronización.
- ✓ *SincronizacionDAO*: Esta clase se usa en conjunto con la clase *Sincronizacion*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- ✓ *SubprocesosdeNegocio*: Esta clase representa datos persistentes para almacenar la información requerida para el manejo de los subprocesos de negocio.
- ✓ *SubprocesosdeNegocioDAO*: Esta clase se usa en conjunto con la clase *SubprocesosdeNegocio*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- ✓ *VincularSubproceso*: Esta clase representa datos persistentes para almacenar la información sobre la vinculación de los actores y subprocesos.
- ✓ *VincularSubprocesoDAO*: Esta clase se usa en conjunto con la clase *VincularSubproceso*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- ✓ *Monitoreo*: Esta clase representa datos persistentes para obtener la información sobre el Monitoreo.
- ✓ *MonitoreoDAO*: Esta clase se usa en conjunto con la clase *Monitoreo*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- ✓ *GestionMensajeria*: Esta clase representa datos persistentes para almacenar y obtener la información sobre el número de teléfono del dispositivo requerido.
- ✓ *GestionMensajeriaDAO*: Esta clase se usa en conjunto con la clase *GestionMensajeria*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- ✓ *GestionCorreoElectronico*: Esta clase representa datos persistentes para almacenar y obtener la información sobre el correo electrónico de los actores del proceso requerido.
- ✓ *GestionCorreoElectronicoDAO*: Esta clase se usa en conjunto con la clase *GestionCorreoElectronico*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- ✓ *Actor*: Esta clase representa datos persistentes para almacenar y obtener información los actores del proceso requerido.
- ✓ *ActorDAO*, Esta clase se usa en conjunto con la clase *Actor*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- ✓ *Roles*: Esta clase representa datos persistentes para almacenar y obtener la información sobre los roles del proceso.

- ✓ *RolesDAO*: Esta clase se usa en conjunto con la clase *Roles*, y hace las implementaciones concretas para el acceso a la fuente de datos.
 - ✓ *Usuarios*: Esta clase representa datos persistentes para almacenar y obtener la información sobre los usuarios de la plataforma.
 - ✓ *UsuariosDAO*, Esta clase se usa en conjunto con la clase *Usuarios*, y hace las implementaciones concretas para el acceso a la fuente de datos.
- **Capa de mediación**
 - Tomcat: Representa al servidor Web y de aplicaciones, necesarios para que la plataforma esté desplegada constantemente en la Web y permita las peticiones de los diferentes dispositivos móviles de manera concurrente.
 - Apache Axis: Es una implementación OpenSource de SOAP que presenta un entorno para la ejecución de servicio Web implementados en java. Apache Axis permite realizar la invocación de servicios Web como la realiza un motor BPEL **¡Error! No se encuentra el origen de la referencia..**
 - JDBC: Permite la comunicación entre los componentes de capas superiores con el motor de base de datos.
 - **Capa de datos**
 - ✓ Motor de base de datos: Este paquete representa el motor de base datos donde son almacenados los datos requeridos por la aplicación Web.

A2.3.2. Diagrama de clases y paquetes Cliente Móvil MobFlow

Diagrama de Clases Cliente Móvil MobFlow

La figura A29 presenta el diagrama de las clases implementadas en la aplicación cliente móvil MobFlow.

Las clases que inician con *frm*, son la clases que presenta una interfaz de usuario que permiten a los actores del proceso realizar sus actividades y comunicarse con MobFlow, las clase que inician con *manejo* representa clase que permiten realizar algunas actividades BPEL y fueron implementadas para complementar algunos aspectos del motor BPEL utilizado y que es representado por la clase *motorBPEL*, más adelante en este documento se explica porque fue necesario implementar las clases contenidas en este paquete.

Las otras clases permiten que la aplicación móvil realice los aspectos de ejecución, comunicación y almacenamiento necesarios.

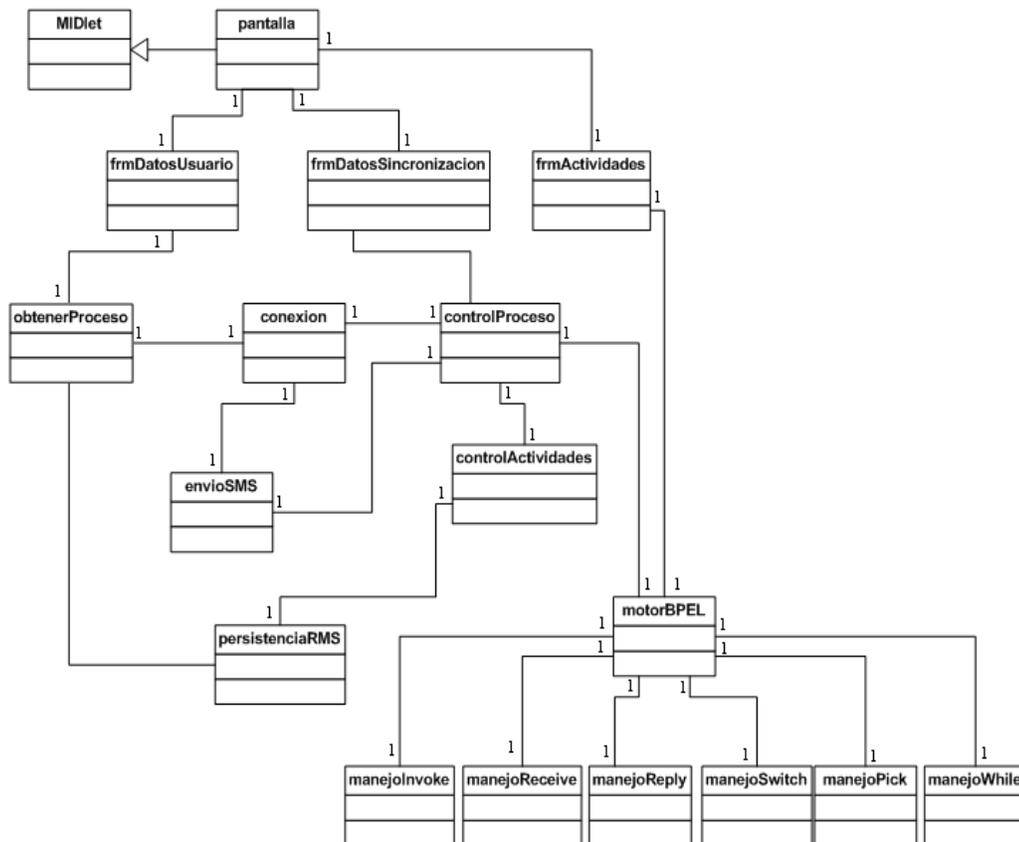


Figura A29. Diagrama de clases cliente móvil MobFlow

Diagrama de Paquetes Cliente Móvil MobFlow

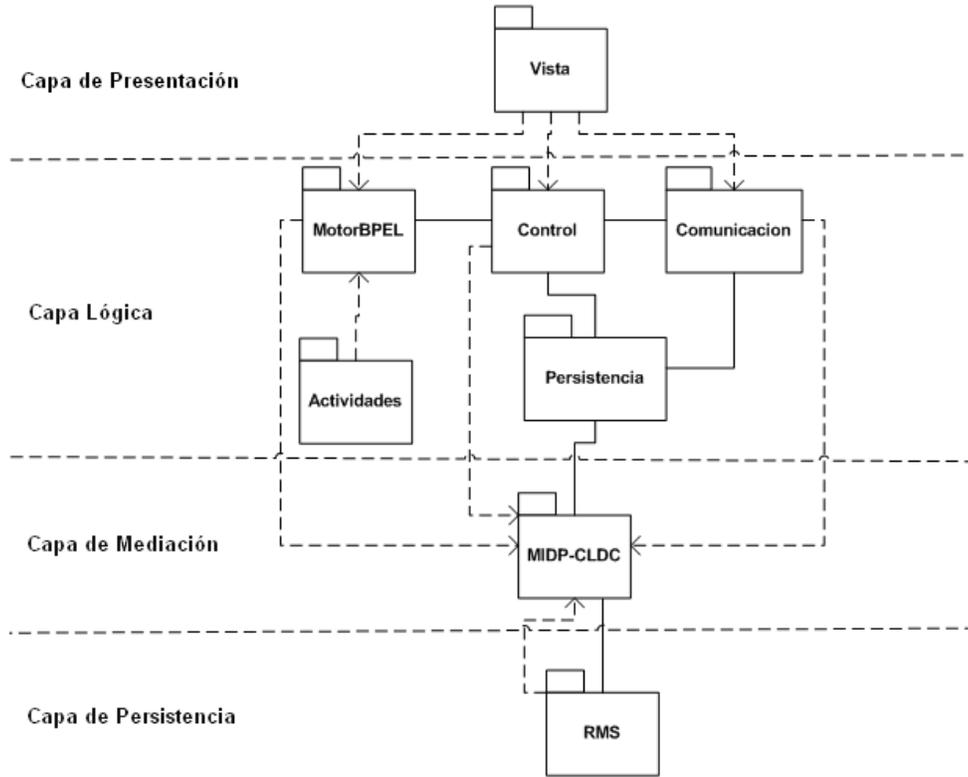


Figura A30. Diagrama de paquetes cliente movil MobFlow

La figura A30 presenta el diagrama de paquete de la aplicación cliente móvil MobFlow, a continuación se describen cada uno de ellos junto con las clases contenidas y que fueron presentadas en la figura 20.

▪ **Capa de Presentación**

- Vista: Contiene las clases que implementan las interfaces gráficas de usuario, que permiten al actor del dispositivo móvil realizar sus tareas del proceso de negocio (Figura A31). Está compuesto de las siguientes clases.

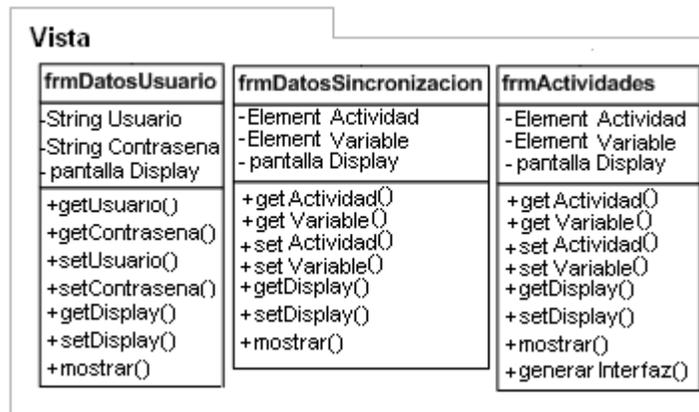


Figura A31. Paquete Vista Cliente Móvil MobFlow

- ✓ *frmDatosUsuario*: Clase que implementa la Interfaz gráfica que permite al usuario ingresar sus datos con el fin de comunicarse con la aplicación Web MobFlow y descargar su respectivo proceso de negocio.
- ✓ *frmDatosSincronizacion*: Presenta los datos de sincronización que recibe el usuario al iniciar un proceso.
- ✓ *frmActividades*: Genera una interfaz de usuario dinámica para que el actor del proceso realice sus actividades o tareas respectivas.

▪ **Capa Lógica**

- **Comunicación**: Este paquete maneja las clases encargadas de todos los aspectos de comunicación que requiere el sistema móvil de información para la ejecución satisfactoria de los procesos de negocio (Figura A32).

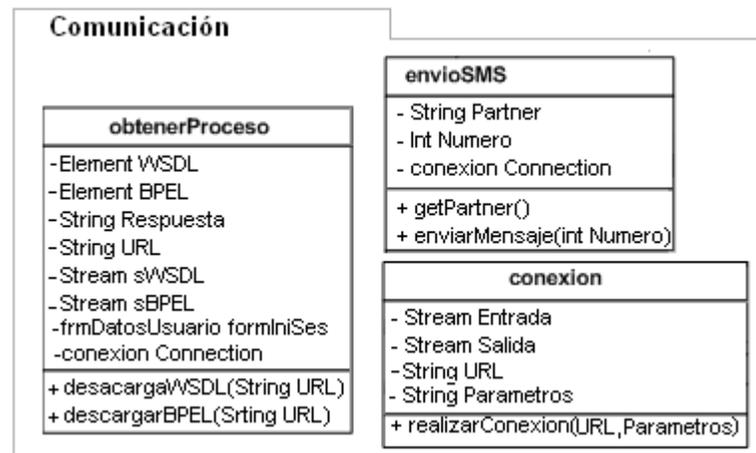


Figura A32. Paquete Comunicación Cliente Móvil MobFlow.

- ✓ *conexion*: Esta clase realiza el proceso de conexión con el servidor con el fin de establecer la comunicación que requiere la ejecución de los procesos de negocio, tales como: invocación a Servicios Web, recibir información de variables para la sincronización, u obtener el respectivo proceso de negocio del usuario.
 - ✓ *obtenerProceso*: Obtiene los datos de usuario y los parametriza para ser enviados al servidor en forma de una petición HTTP. Cuando se obtiene un respuesta del servidor, por medio de esta clase se serializa y deserializa la información enviada por esté, para formar los archivo del proceso correspondiente.
 - ✓ *envioSMS*: Esta clase permite enviar un mensaje de texto al siguiente dispositivo en la ejecución una vez finalizada la ejecución del subprocesso.
- **Motor BPEL**: Contiene las clases que componen el motor BPEL Sliver utilizado por la aplicación cliente móvil MOBFLOW, para procesar los documentos BPEL descargado por los usuarios (Figura A33). Contiene la siguiente clase.

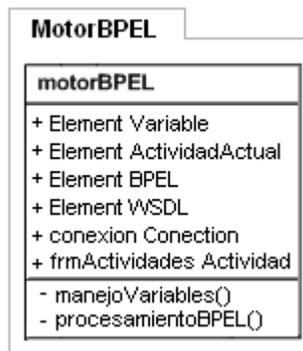


Figura A33. Paquete Motor BPEL Cliente Móvil MobFlow.

- ✓ *motorBPEL*: Representa el motor BPEL que interpreta y ejecuta los subprocesos BPEL obtenidos por el usuario. Como ya se menciono se utiliza el motor BPEL para dispositivos móviles Sliver.
- Actividades: Contiene las clases que complementan el motor BPEL Sliver, para poder realizar el procesamiento de los procesos BPEL generados por MOBFLOW e interactuar con la misma.(Estas clases se implementaron debido a algunas restricciones del motor Sliver descritas más adelante, para otras actividades BPEL no fue necesario la implementación) Contienen las siguiente clases (Figura A34):
 - ✓ *manejoInvoke*: Clase que interactúa con el paquete motor BPEL, para ejecutar actividades de tipo *<invoke>* en el proceso BPEL.
 - ✓ *manejoReceive*: Clase que interactúa con el paquete motor BPEL, para ejecutar actividades de tipo *<receive>* en el proceso BPEL.
 - ✓ *manejoReply*: Clase que interactúa con el paquete motor BPEL, para ejecutar actividades de tipo *<reply>* en el proceso BPEL.
 - ✓ *manejoSwitch*: Clase que interactúa con el paquete motor BPEL, para ejecutar actividades estructuradas de tipo *<switch>* en el proceso BPEL.

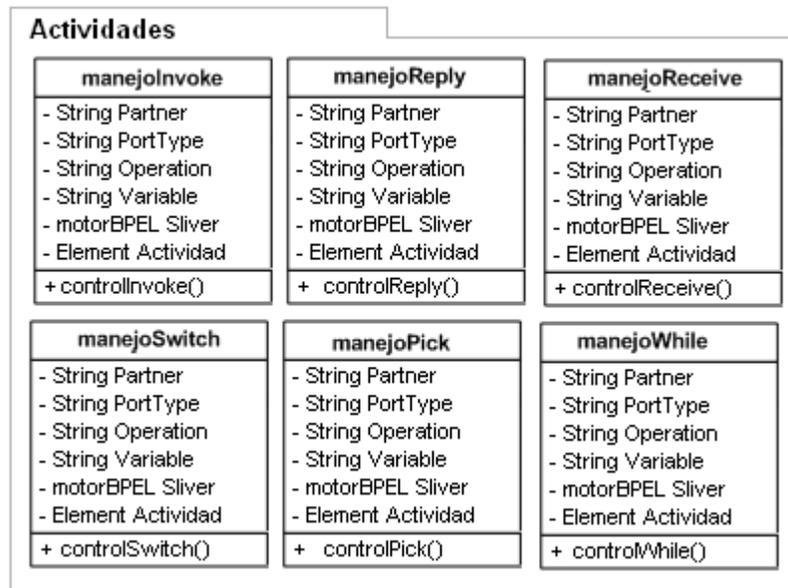


Figura A34. Paquete Actividades Cliente Móvil MobFlow.

- ✓ *manejoPick*: Clase que interactúa con el paquete motor BPEL, para ejecutar actividades estructuradas de tipo *<pick>* en el proceso BPEL.
- ✓ *manejoWhile*: Clase que interactúa con el paquete motor BPEL, para ejecutar actividades estructuradas de tipo *<while>* en el proceso BPEL.
- **Control**: Este paquete contiene las clases encargadas de controlar la ejecución del proceso para lo cual se interactúa con la aplicación Web (Figura A35). A continuación se describe cada una de ellas:

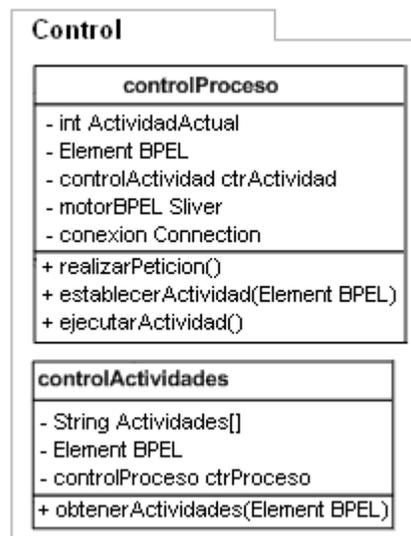


Figura A34. Paquete Control Cliente Móvil MobFlow.

- ✓ *controlActividades*: Clase que recorre el proceso BPEL y permite establecer el orden de ejecución de las actividades del proceso.
- ✓ *controlEjecucion*: Clase que permite realizar la ejecución controlada de las diferentes actividades definidas del proceso, lo hace en conjunto con la clase del paquete **motor BPEL**.
- Persistencia: Maneja las clases que implementan el almacenamiento persistente en la aplicación móvil a través de registros (Figura A35). Está compuesto por la siguiente clase:

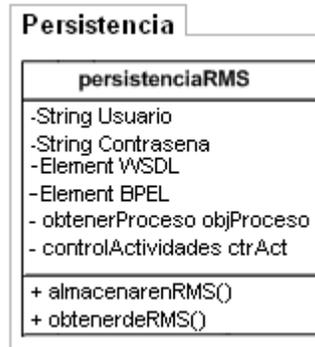


Figura A35. Paquete Persistencia Cliente Móvil MobFlow.

- ✓ *persistenciaRMS*: Se encarga de almacenar y obtener los archivos del proceso a través de registros, para que sean cargados y procesados por el motor BPEL.
- **Capa de Mediación**
 - MIDP-CLDC: Representa la estructura de la aplicación en configuraciones y perfiles. Una configuración CLDC contiene la maquina virtual Java (KVM) y un conjunto de clases genéricas (“configuración”) que pueden ser empleadas en un amplio rango de dispositivos. Por su parte, el “perfil” MIDP contiene las clases que permiten el desarrollo de las aplicaciones en un tipo de dispositivo determinado.
 - ✓ *MIDlet*: La clase MIDlet representa la clase que hereda de la clase abstracta *javax.microedition.midlet.MIDlet* y permite el control de las aplicaciones para dispositivos móviles desarrolladas bajo el perfil MIPD-CLDC.
 - ✓ *pantalla*: Cada MIDlet tiene una referencia a este objeto. Recupera información acerca del despliegue actual (Ej. Nro. colores) e incluye métodos para que los objetos sean ubicados en la pantalla.
- **Capa de Persistencia**
 - RMS: Representa el sistema de almacenamiento, en este caso se usa el Sistema Administrador de Registros RMS (*Record Management System*). Este proporciona un mecanismo que permite almacenar datos de forma persistente para su futura recuperación. Este mecanismo está implementado sobre una base de datos basada en registros.

A2.4 Modelo relacional de la base de datos de la Aplicación Web MobFlow

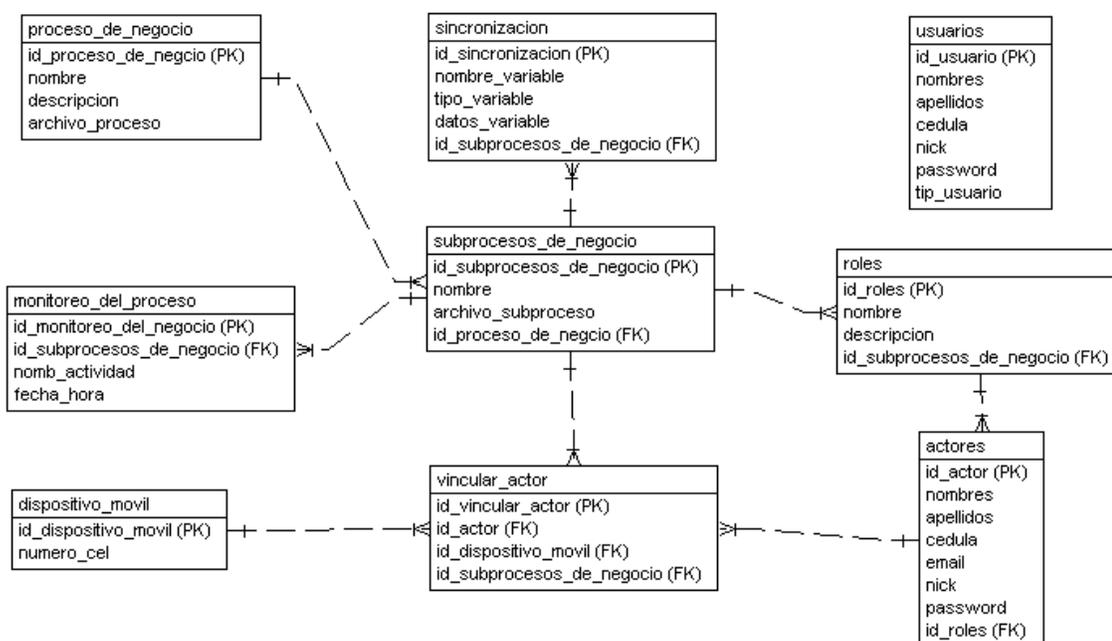


Figura A36. Modelo relacional de la aplicación Web MobFlow.

En la figura A36 se presenta el modelo relacional de la base de datos de la aplicación web MobFlow, este modelo define las tablas y relaciones necesarias para el manejo de los procesos de negocio en la plataforma Web. A continuación se describe el modelo relaciona a partien del diccionario de datos.

A.2.4.1 Diccionario de datos

Nombre de la Tabla: proceso_de_negocio.

Descripción: Permite almacenar la información de los procesos de negocio que se cargan en la plataforma.

Campo	Descripción	Tipo Campo	Tamaño Dato	Validación
id_proceso_de_negocio	Identificador del proceso de negocio.	Character varing	100 Bytes	* obligatorio * único
nombre	Nombre completo del usuario	Character varing	80 Bytes	* obligatorio
descripcion	Descripción del proceso de negocio.	Character varing	200 Bytes	* obligatorio
archivo_proceso	Guarda la ubicación de los archivos cargados.	Character varing	150 Bytes	* obligatorio

Tabla A36. Descripción de la tabla proceso_de_negocio

Nombre de la Tabla: subproceso_de_negocio.

Descripción: Permite almacenar la información de los subprocesos de negocio, que se cargan en la plataforma.

Campo	Descripción	Tipo Campo	Tamaño Dato	Validación
id_subproceso_de_negocio	Identificador del proceso de negocio.	Character varing	100 Bytes	* obligatorio * único
nombre	Nombre completo del usuario	Character varing	80 Bytes	* obligatorio
archivo_proceso	Guarda la ubicación de los archivos generados.	Character varing	200 Bytes	* obligatorio
id_proceso_de_negocio	Identificador del proceso de negocio.	Character varing	100 Bytes	* obligatorio

Tabla A37. Descripción de la tabla subproceso_de_negocio

Nombre de la Tabla: sincronización.

Descripción: Permite almacenar la información de las variables de sincronización de los subprocesos en ejecución.

Campo	Descripción	Tipo Campo	Tamaño Dato	Validación
id_sincronizacion	Identificador variable de sincronizacion	Character varing	100 Bytes	* obligatorio * único
nombre_variable	Nombre de la variable de sincronización	Character varing	80 Bytes	* obligatorio
tipo_variable	Tipo dato de la variable de sincronización	Character varing	200 Bytes	* obligatorio
datos_variable	Valor de la variable de sincronización	Character varing	150 Bytes	* obligatorio

Tabla A38. Descripción de la tabla sincronizacion.

Nombre de la Tabla: usuarios

Descripción: Permite almacenar la información de los usuarios de la plataforma.

Campo	Descripción	Tipo Campo	Tamaño Dato	Validación
id_usuario	Identificador del usuario del sistema.	Character varing	100 Bytes	* obligatorio * único
nombres	Nombres del usuario.	Character varing	80 Bytes	* obligatorio
apellidos	Apellidos del usuario.	Character varing	80 Bytes	* obligatorio
cedula	Identificador del usuario.	Character varing	20 Bytes	* obligatorio
nick	Nick del usuario.	Character varing	50 Bytes	* obligatorio
password	Clave del usuario.	Character varing	50 Bytes	* obligatorio
tip_usuario	Tipo de usuario.	Character varing	20 Bytes	* obligatorio

Tabla A38. Descripción de la tabla usuarios.

Nombre de la Tabla: actores

Descripción: Permite almacenar la información de los actores del proceso de negocio.

Campo	Descripción	Tipo Campo	Tamaño Dato	Validación
id_actor	Identificador del actor de un proceso.	Character varing	100 Bytes	* obligatorio * único
nombres	Nombres del actor.	Character varing	80 Bytes	* obligatorio
apellidos	Apellidos del actor.	Character varing	80 Bytes	* obligatorio
cedula	Identificador del actor.	Character varing	20 Bytes	* obligatorio
nick	Nick del actor.	Character varing	50 Bytes	* obligatorio
password	Clave del actor.	Character varing	50 Bytes	* obligatorio
email	Email del actor.	Character varing	20 Bytes	* obligatorio
id_rols	Identificador del rol que tiene el actor.	Character varing	100 Bytes	* obligatorio

Tabla A39. Descripción de la tabla actores.

Nombre de la Tabla: roles

Descripción: Permite almacenar la información de los roles que puede cumplir un actor en la ejecución de un proceso.

Campo	Descripción	Tipo Campo	Tamaño Dato	Validación
id_rols	Identificador de los roles.	Character varing	100 Bytes	* obligatorio * único
nombre	Nombre del rol.	Character varing	80 Bytes	* obligatorio
descripcion	Descripción del rol.	Character varing	250 Bytes	* obligatorio
id_subprocesos_de_negocio	Identificador del subproceso que identifica al rol.	Character varing	100 Bytes	* obligatorio

Tabla A40. Descripción de la tabla roles.

Nombre de la Tabla: monitoreo_del_proceso.

Descripción: Permite almacenar la información relacionada con la ejecución de actividades de un subproceso específico.

Campo	Descripción	Tipo Campo	Tamaño Dato	Validación
id_monitoreo_del _proceso	Identificador de la tabla monitoreo.	Character varing	100 Bytes	* obligatorio * único
id_subprocesos_de _negocio	Identificador del subproceso.	Character varing	100 Bytes	* obligatorio
nomb_actividad	Nombre de la actividad ejecutada.	Character varing	80 Bytes	* obligatorio
fecha_hora	Fecha y hora de la ejecución.	Date	60 Bytes	* obligatorio

Tabla A41. Descripción de la tabla monitoreo_del_proceso.

Nombre de la Tabla: dispositivo_movil.

Descripción: Permite almacenar la información del número correspondiente al dispositivo móvil que ejecuta un subproceso. Esta información fue almacenada en una tabla individual por razones de escalabilidad.

Campo	Descripción	Tipo Campo	Tamaño Dato	Validación
id_dispositivo_movil	Identificador de la tabla del dispositivo.	Character varing	100 Bytes	* obligatorio * único
numero_cel	Numero del dispositivo.	Character varing	30 Bytes	* obligatorio

Tabla A42. Descripción de la tabla dispositivo_movil.

A3. Descripción Servicios Web Caso de Estudio

Servicio Web XML para el estudio de crédito (ServicioCredito)

Este servicio Web XML es el encargado de realizar las operaciones correspondientes al estudio de crédito, cuando se realiza una compra de un artículo de esta forma. Este Servicio implementa las funciones de la aprobación o denegación del crédito.

Servicio Web XML para el Registro o facturación de una venta (ServicioRegistroVenta)

Este servicio Web XML, contiene las funciones para recibir el pago de la venta de determinado artículo, ya sea el pago completo o por cuotas, para que esta información posteriormente se confirme en el sistema central del almacén.

Servicio Web XML para la actualización del inventario de artículos (ServicioInventario)

Servicio Web XML, que contiene las funciones para actualizar el inventario del almacén después de realizada una venta, establecer el artículo que fue vendido (Serial del artículo), para que con esta información la central pueda imprimir posteriormente la factura y dar la orden de salida al artículo.

Servicio Web XML para el ingreso de una venta (ServicioDatosVenta)

Este servicio Web XML, permite capturar los datos de una venta para que sea remitida al almacén, estos datos de las ventas pueden ser de una venta a crédito o contado.

Servicio Web XML para el registro de la salida de un artículo (ServicioBodega)

Este servicio Web XML, contiene las funciones que permiten registrar la salida de un artículo de la bodega del almacén.

ANEXO B. MANUAL DE INSTALACIÓN Y DE USUARIO

B1. Manual de Instalación

Como se presento en el capítulo 3, la plataforma MobFlow propuesta en el presente trabajo de grado se basa en dos sistemas principales, los cuales están representados por las aplicaciones Aplicación Web MobFlow y Aplicación Móvil MobFlow.

Requisitos del sistema

Ya que los entornos manejados para el despliegue de las aplicaciones son de naturaleza Java, no se necesita un Sistema Operativo (SO) en específico para realizar la instalación. Sin embargo, si es necesaria la instalación previa del JDK 5 o superior instalada para la aplicación Web MobFlow en los equipos de escritorio y dispositivos móviles con soporte para Java.

▪ Herramientas Software

Las siguientes son herramientas de distribución libre para el almacenamiento y despliegue de los datos manejados por la aplicación Web MobFlow.

✓ Software Requerido para el Despliegue de las Aplicaciones Web y Móvil

Para el despliegue de las aplicaciones Web y Móvil MobFlow, se utilizo el instalador del IDE Netbeans 6.7.1 que viene instalado con Apache Tomcat 6.0.1, el cual sirve como contenedor de aplicaciones Web. Además se instalo un componente adicional llamado Apache Axis que es una implementación OpenSource de SOAP que proporciona un entorno de ejecución para Servicios Web.

Si se requiere el IDE Netbeans permite simular la aplicación Cliente Móvil MobFlow o con la aplicación Sun Java Wireless Toolkit for CLDC, en caso contrario se necesita que el equipo móvil tenga soporte para aplicaciones Java. Para lograr instalar con éxito las aplicaciones y debido a las dependencias que tienen, es necesario instalar siguiendo el orden presentado a continuación:

1. Instalar en primer lugar IDE Netbeans 6.7.1 que viene instalado con Apache Tomcat 6.0.1.
2. En segundo lugar, se instala y configura dentro del servidor Apache Tomcat el componente Apache Axis, el cual permite contener los servicios Web que se van a desplegar.
3. Por último se instala eclipse para facilitar la configuración de la aplicación Web.

▪ Instalación del Servidor Web Apache Tomcat y el IDE Netbeans

Para instalar del IDE Netbeans 6.7.1, primero que todo, se descargar el instalador para el sistema Operativo. Ya obtenido el instalador, se procede de la siguiente forma:

Al ejecutar el instalador aparece la siguiente interfaz Figura B1.



Figura B1. Interfaz de instalación.

Con objetivo de disponer del servidor de aplicaciones Apache Tomcat que se requiere, hacemos clic en "Personalizar", con lo que aparece la figura B2 dada a continuación.

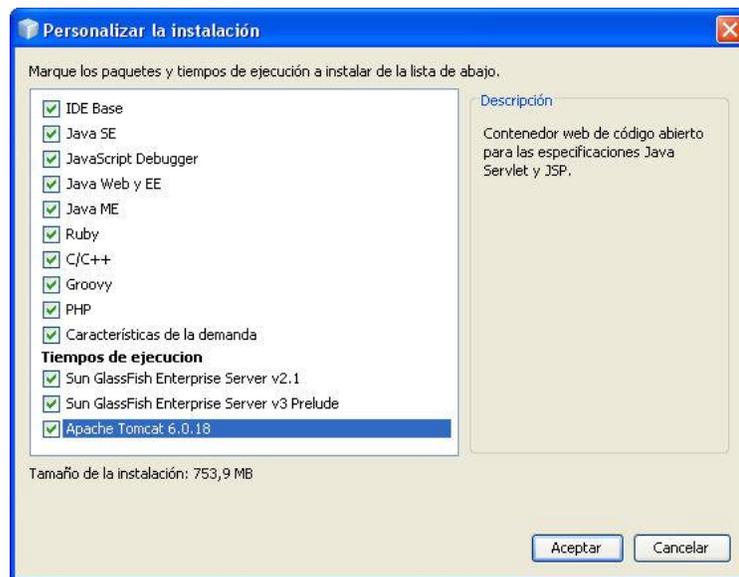


Figura B2. Opciones de instalación.

El paso siguiente es seleccionar el servidor de aplicaciones Apache Tomcat 6.0.18, una vez seleccionado damos aceptar.

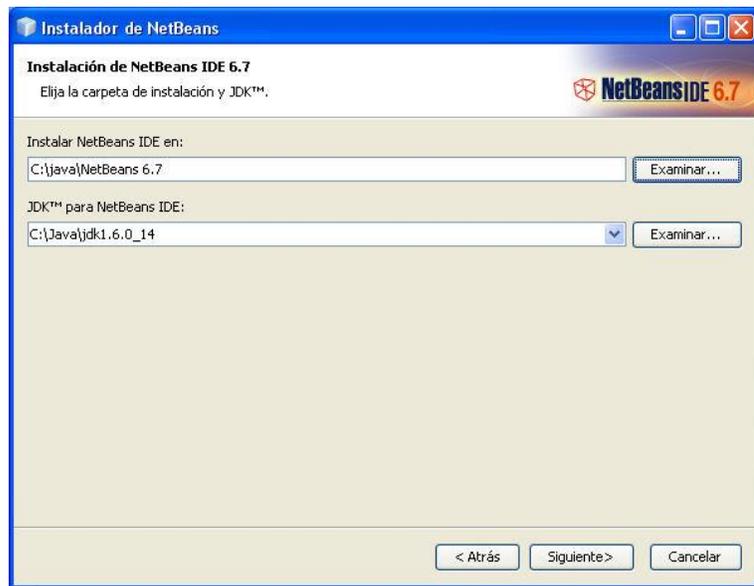


Figura B3. Subdirectorio de instalación Netbeans.

A continuación fijamos el directorio y subdirectorios donde deseamos instalar (Figura B4).

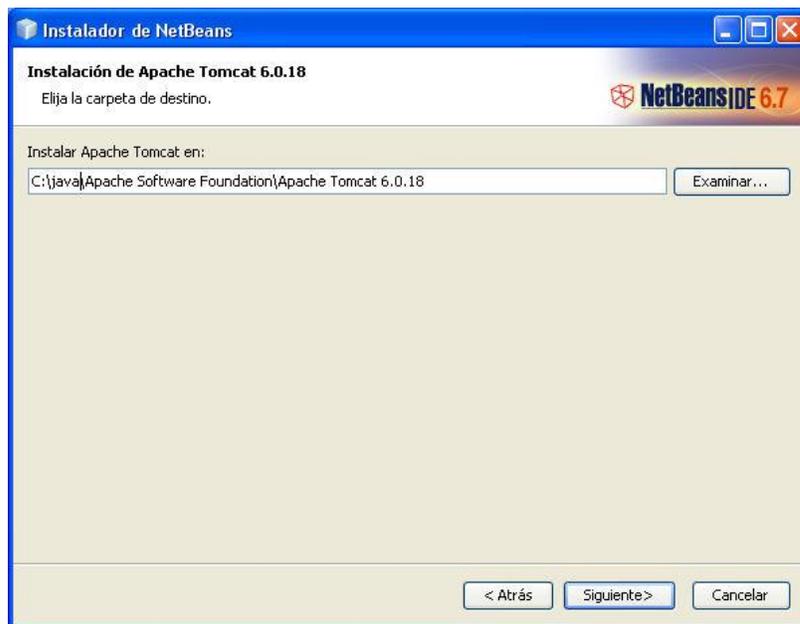


Figura B4. Subdirectorio de instalación Apache Tomcat.

Fijamos ahora el directorio y subdirectorio de Apache Tomcat, donde será instalada la aplicación Web MobFlow (Figuras B3 y B4).

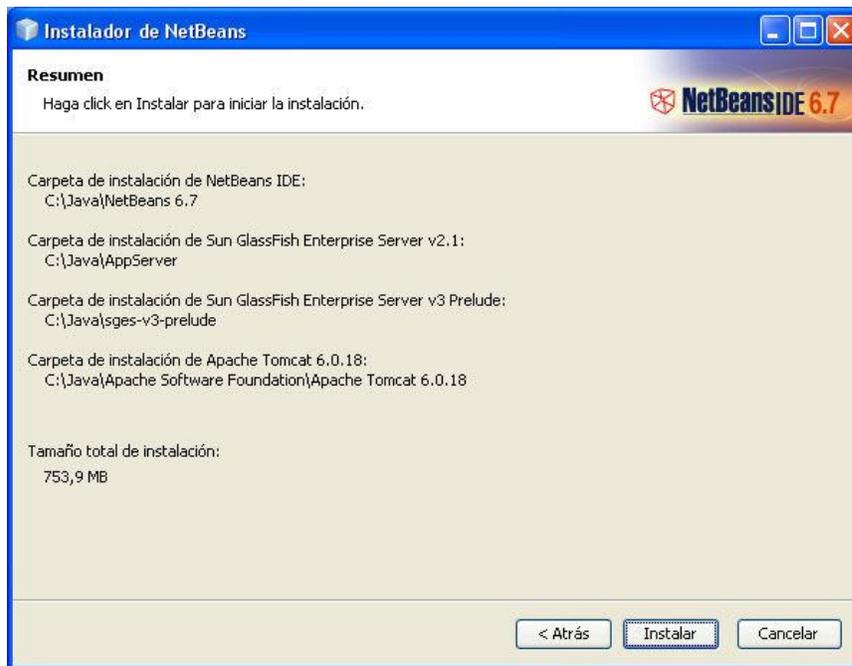


Figura B5. Subdirectorio de instalación Apache Tomcat.

En esta interfaz muestra un resumen de los componentes a instalar donde finalmente se da clic en Instalar (Figura B5).

▪ Instalación y Configuración de Apache Axis

Como se menciona anteriormente utilizaremos el componente Axis, para el llamado y publicación de los servicios Web que utilizamos con la plataforma. Para instalar el componente, primero que todo, se descarga la versión binaria del instalador Axis2 para el sistema Operativo destino. Ya obtenido el instalador, se procede de la siguiente forma:

- ✓ Se descomprime la versión zip de la distribución WAR, y el fichero axis2.war en la carpeta webapps del Apache Tomcat, como se muestra en el ejemplo dado a continuación: C:\TutorialWS\apache-tomcat-6.0.18\webapps
- ✓ Se añade a la variable PATH del sistema la ruta del instalador, como se muestra a continuación: C:\TutorialWS\axis2-1.4.1\bin
- ✓ Se descomprime la distribución binaria en una carpeta, y se establece la variable AXIS2_HOME a la ruta absoluta del directorio descomprimido, como se muestra en el siguiente ejemplo: AXIS2_HOME=C:\TutorialWS\axis2-1.4.1
- ✓ Para comprobar que el servidor y el motor de axis se han instalado correctamente, se arranca el servidor ejecutando CATALINA_HOME\bin\startup.bat, la cual muestra una ventana como la que se muestra a continuación.

▪ Instalación de eclipse

Antes de instalar este ID de desarrollo, es necesario tener instalado el JDK 1.3 o superior y un mínimo de memoria de 256 MB. Para la instalación, primero que todo, se descarga la versión de eclipse para el sistema Operativo destino. Ya obtenido el instalador se procede de la siguiente forma:

- ✓ Se descomprime el fichero eclipse-SDK-3.0-win32.zip que se descargo en el directorio c:\eclipse al que llamamos eclipse_home.
- ✓ Para arrancar la plataforma eclipse se ejecuta eclipse.exe que se encuentra dentro eclipse_home.
- ✓ Para configurar el servidor Tomcat con el entorno de desarrollo, se debe ir al menú Window - > Preferences, se selecciona la pestaña Tomcat, y se procede escoger la versión del servidor, y la ubicación donde se instalado el servidor Tomcat.

▪ Sistema Gestor de Base de Datos

Para este caso se utilizo el gestor de base de datos Postgres en la versión 8.3. Primero que todo, se descarga el instalador y a continuación se siguen los siguientes pasos:

- ✓ Seleccionar el lenguaje en el cual se instalara Postgres 8.3 Figura B6.

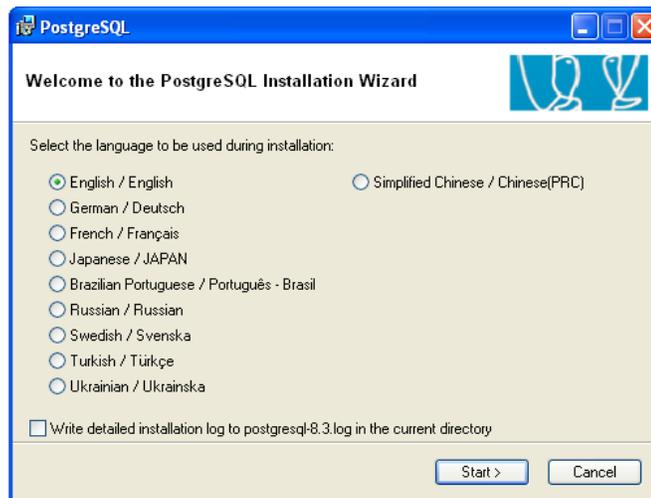


Figura B6. Selección del lenguaje en el que se instalara Postgres 8.3

- ✓ En la pantalla siguiente se aprecian opciones adicionales a la instalación normal del producto. Para este caso, damos clic en siguiente Figura B8.

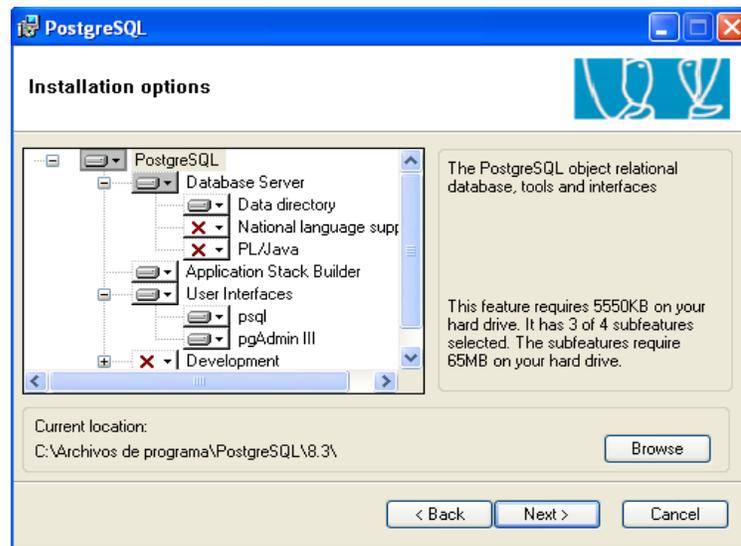


Figura B8. Opciones de instalación

- ✓ A continuación se presenta la interfaz de configuración del servicio donde se puede crear una cuenta que ejecuta el servidor Postgres. Posteriormente, asignamos un password, lo confirmamos y pasamos al siguiente paso Figura B9.

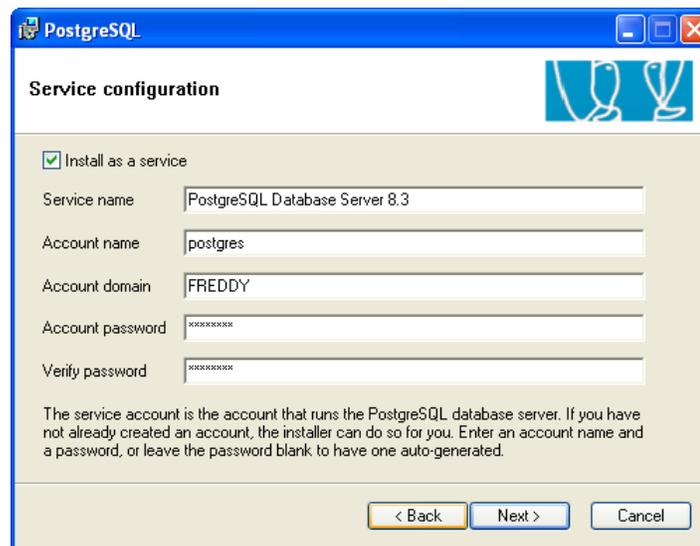


Figura B9. Opciones de instalación

- ✓ En la Figura B10, se configura el puerto en el que quiere ejecutarse Postgres, así como la codificación, localidad y los datos del súper usuario que administrara la aplicación.



Figura B10. Inicializar el cluster de la base de datos Postgres.

- ✓ Por último se da clic en siguiente hasta finalizar la instalación.

▪ Despliegue de la plataforma MobFlow

✓ Creación de la bases de datos

La aplicación Web MobFlow, requiere almacenar su información en una base de datos. Por ello es necesaria la creación de la siguiente base de datos sobre el servidor Postgres anteriormente descrito:

- *DBproceso*: base de datos utilizada por la aplicación, para el almacenamiento de la información de los procesos de negocio y los usuarios del sistema.
- Para el caso de estudio es necesario crear una base de datos para al almacenar los datos de las ventas, la cual debe ser llamada TiendaElectrodomesticos.

▪ Despliegue de las Aplicaciones

Para realizar esta labor, se recomienda utilizar el IDE Netbeans 6.7.1 que viene instalado con Apache Tomcat, con el fin de facilitar las tareas que implica el despliegue.

Por otro lado, para lograr instalar con éxito las aplicaciones y debido a las dependencias que tienen, es necesario instalar siguiendo el orden presentado a continuación:

- Instalar en primer lugar la aplicación Web MobFlow ya que no tiene dependencia alguna, al contrario la aplicación móvil depende de ella.
- En segundo lugar, se instala la aplicación Cliente Móvil MobFlow, ya que hacen uso de los servicios desplegados por la plataforma Web. Para lo cual se copia en el dispositivo sobre el cual se va a ejecutar MOBFLOW/MobileApplication.jar

Configuración

Aplicación Web: La carpeta mobflow contiene el proyecto con la aplicación Web de la plataforma, la cual debe cargarse en el servidor Web Apache Tomcat, para ello es recomendable utilizar el ID eclipse o Netbeans ya que contienen las herramientas que permiten facilitar esta labor. Adicional a la instalación del proyecto es necesario copiar la carpeta recursos en la unidad C del equipo. Por último se debe copiar el archivo cabeceraBpelSliver.xml en la carpeta bin de Apache Tomcat. Las carpetas y archivos requeridos vienen en el CD de instalación.

Caso de estudio: La carpeta caso de estudio contiene los servicios web (.jws) creados para el caso de estudio, estos deben ser copiados dentro de la carpeta Axis ubicada dentro del directorio donde se instaló el Apache Tomcat.

B2. Manual de Usuario

Una vez instaladas todas las aplicaciones se puede usar la aplicación. A continuación se explicarán los pasos básicos para entender el funcionamiento del sistema, tanto para el usuario administrador, como para el usuario de consulta de la plataforma Web y posteriormente se describirá el manual de usuario para la aplicación móvil.

Manual para el usuario administrador.

Para acceder a la aplicación Web es necesario autenticarse, por lo tanto, la primera pantalla que verá es una interfaz donde debe ingresar el nombre de usuario admin opción (1) y contraseña admin. Opción (2) de la figura B11, luego presione el botón (3) para el ingreso.



Figura B11. Iniciar sesión.

Una vez autenticado, será dirigido a la página principal donde puede ver un menú en la parte superior izquierda de la aplicación como se muestra en la figura B12. En esta interfaz se presentan seis secciones claramente definidas:

- (1) Sección subir proceso, donde se presenta el formulario para que los usuarios pueden subir el proceso de negocio BPEL así como el WSDL y la información del mismo.
- (2) SubProcesos, donde se hace el particionamiento del proceso de negocio en subprocesos BPEL y se vinculan los actores y dispositivos por cada subproceso.
- (3) Visualizar proceso, donde el usuario puede visualizar el proceso de negocio en una interfaz gráfica.
- (4) Monitoreo, donde el usuario puede consultar y monitorear la ejecución del proceso de negocio.

(5) Usuarios, donde el usuario Administrador puede registrar diferentes usuarios que tiene acceso a la aplicación Web

(6) Actores, donde el usuario Administrador puede registrar diferentes actores que tiene acceso a la aplicación Web desde los sistemas móviles de información

(7) Roles, donde el usuario Administrador puede consultar y editar los diferentes roles manejados por los procesos.

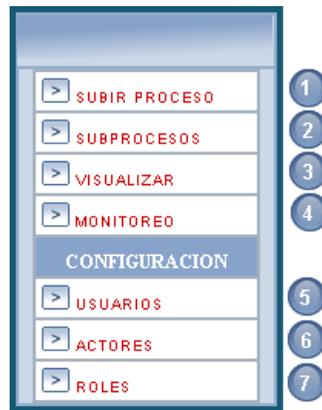


Figura B12. Menú principal.

Al oprimir la opción (1) del menu principal (figura B12) se despliega una interfaz que permite cargar los procesos de negocio representados por un documento BPEL así como el WSDL correspondiente, junto con la información del mismo como el nombre y la descripción del proceso de negocio.

Para realizar la carga del proceso se busca el documento BPEL y WSDL en la opción examinar de la opción 3 y 4 en la figura B13, se llenan los campos nombre (1) , descripción (2) y se le da clic en el botón cargar (5), en ese momento el documento BPEL y WSDL es subido a la plataforma donde se analiza si el documento BPEL está bien formado, lo cual carga los documento y la información de los mismos en la base de datos y muestra una tabla con la información del proceso de negocio, en caso contrario retorna un mensaje de error avisando que existen errores en el documento BPEL razón por la cual no se pudo completar la transacción.

Cargar Proceso de Negocio		
Nombre	1	<input type="text"/>
Descripcion	2	<input type="text"/>
Documento WSDL	3	<input type="button" value="Seleccionar archivo"/> No se h... archivo
Documento BPEL	4	<input type="button" value="Seleccionar archivo"/> No se h... archivo
	5	<input type="button" value="Enviar"/>

PROCESO DE NEGOCIO			
NOMBRE	DESCRIPCION	ARCHIVO	ELIMINAR
MisVentas	proceso de ventas	MisVentas.bpel	<input type="button" value="X"/>

Figura B13. Pagina cargar proceso de negocio.

Una vez cargado el proceso de ventas, se puede acceder a la interfaz gráfica de usuario que permite realizar el particionamiento del proceso, esta interfaz es presentada en la figura B13, donde se selecciona el botón particionar (1). Al ejecutar el particionamiento se obtienen los subprocesos, como se muestra en la figura B14.

PROCESO DE NEGOCIO			
NOMBRE	DESCRIPCION	ARCHIVO	PARTICIONAR
MisVentas	proceso de ventas	MisVentas.bpel	<input type="button" value="PARTICIONAR"/> 1

SUBPROCESOS DE NEGOCIO	
NOMBRE	ACTOR Y DISPOSITIVO
ServicioInventario	 → 
ServicioVentaCredito	 → 
RegistroVenta	 → 
DatosVenta	 → 

Figura B14. Subprocesos resultantes del Particionamiento.

Posterior al particionamiento de negocio se necesita crear los actores, que ejecutaran cada proceso para eso la plataforma cuenta con una interfaz que me permite crear y gestionar toda la información de los actores del proceso la cual se muestra en la figura B15.

Nuevo

LISTA DE ACTORES						
NOMBRES	APELLIDOS	CEDULA	NICK	PASSWORD	EDIT	ELIMINAR
henry william	dorado gomez	5617264	hdorado	123		
gustavo adolfo	aponza	5423135453	gustavo	123		

Figura B15. Interfaz gestión de actores.

Particionado y creados los actores del proceso, la plataforma Web permite proceder a vincular los actores con los subprocesos que ejecutarán en sus dispositivos móviles, para ello se selecciona la opción (2) de la figura B14 que carga la interfaz presentada en la figura 44.

Al cargar la interfaz se despliegan los actores asociados con el rol del subproceso (seleccionado en la anterior interfaz), estos actores se despliegan en la opción (1) como se muestra en un ejemplo en la figura B15, de la cual se escoge el actor para vincular con el subproceso lo que despliega sus datos automáticamente. Luego mediante la opción (2) se ingreso el número de celular del Actor y con el botón Adicionar (3) se hace vinculación. Este proceso se realiza de igual manera para cada uno de los subproceso generados.

Realizado el proceso de vinculación los actores pueden descargar sus procesos a los dispositivos móviles.

Vincular Actor	
Rol del Actor:	ServicioInventario
Actor:	1 <input type="text" value="gustavo adolfo"/>
Nombres:	<input type="text" value="gustavo adolfo"/>
Apellidos:	<input type="text" value="aponza"/>
Cedula:	<input type="text" value="5423135453"/>
Celular:	2 <input type="text" value="300159889"/>
	3 <input type="button" value="Adicionar"/>

Figura B15. Interfaz que permite Vincular a los actores del proceso.

Para la descarga de los subproceso se ejecuta la aplicación Cliente Móvil MobFlow (Figura B16), del menú principal se escoge la opción 1, esta opción carga la interfaz gráfica presentada en la figura B17, la cual permite la descarga de los subprocesos correspondientes a cada actor. Para ello se llenan los campos Nombre de Usuario (1), Contraseña (2) y se presiona el botón Conectar (3) (Para cada usuario se ejecuta la aplicación Cliente Móvil en un dispositivo diferente).

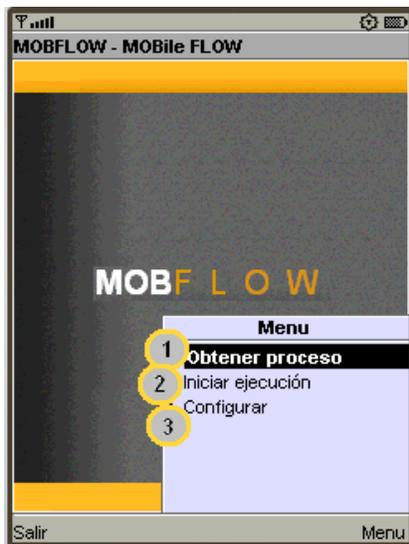


Figura B16. Menú principal aplicación cliente Móvil MobFlow



Figura B17. Interfaz gráfica para el registro desde el móvil a la plataforma.

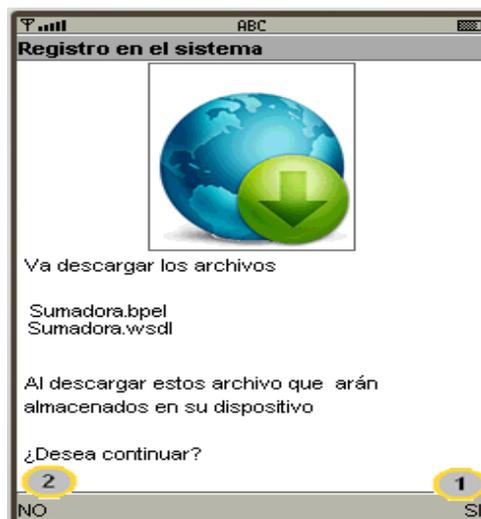


Figura B18. Interfaz para descarga de procesos.

Luego de que el usuario ingresa sus datos (Nombre de usuario, contraseña), la aplicación móvil le presenta el nombre del proceso asignado (BPEL, WSDL) con opciones para la descarga o no en el dispositivo (Figura B18). Si desea descargar ese proceso el usuario debe presionar el botón SI (1), así la aplicación móvil inicia la descarga y almacenamiento de los archivos. AL presionar el botón NO (2) la descarga no ocurre y vuelve nuevamente a presentar la pantalla de la interfaz inicial.

Cuando el usuario desea iniciar la ejecución de su subprocesso presiona la opción Ejecutar (2) del menú principal (Figura B16), de aquí el sistema presenta una interfaz para lanzar el proceso descargado y ejecutarlo en el motor BPEL, como la presentada en la figura B19, por lo cual para iniciar la ejecución del

proceso el usuario debe presionar el botón *Lanzar Proceso* (1) en cual genera la interfaz presentada en la figura B20, en caso contrario presiona volver (2) y presenta la interfaz del menú principal (Figura B16). La interfaz de ejecución (figura B20.) se genera dependiendo de las tareas propias del actor las cuales están dadas por el subproceso descargado, el usuario debe realizar el ingreso de datos según sus tareas, cuando finalice presione el botón *Send* (2) se procesa esta información, en caso de que requiera de mas tareas la aplicación genera las interfaces según lo definido el subproceso descargado, en caso contrario presenta una pantalla con mensaje de ejecución exitosa o errónea según sea el caso.



Figura B19. Interfaz lanzar ejecución del Proceso



Figura B20. Interfaz para ejecución de procesos.

Para eliminar un proceso de negocio el usuario debe presionar el botón configuración (3) de la figura B16 al presionarlo le presenta la interfaz presentada en la figura B21, en la cual el usuario debe presionar el botón *Borrar proceso de negocio* (1) para realizar la eliminación o si no presiona el botón (2) para volver a la interfaz de la figura B16.

Si presiona el botón (1) de la figura B21 se carga la interfaz presentada en la figura B22, en la cual al presionar el botón *BORRAR PROCESO* (1) se confirma la eliminación del proceso y se vuelve a la interfaz de la figura B16.

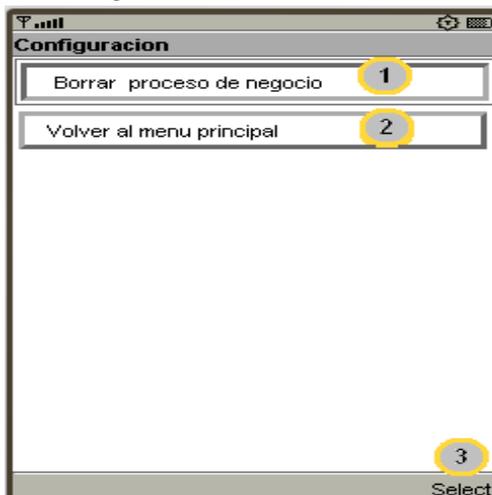


Figura B21. Interfaz eliminar un Proceso



Figura B22. Interfaz para confirmar eliminación.

A medida que los actores ejecutan los subprocessos en los dispositivos móviles, se puede monitorear el estado de la ejecución, mediante una interfaz como la presentada en la figura B23, en esta interfaz muestra la información de la ejecución del proceso de negocio, en donde se muestra la información de los subprocessos, actores, y las actividades ejecutadas por los subprocessos generados, además el icono del dispositivo móvil señala el ultimo subprocessos en ejecutarse.

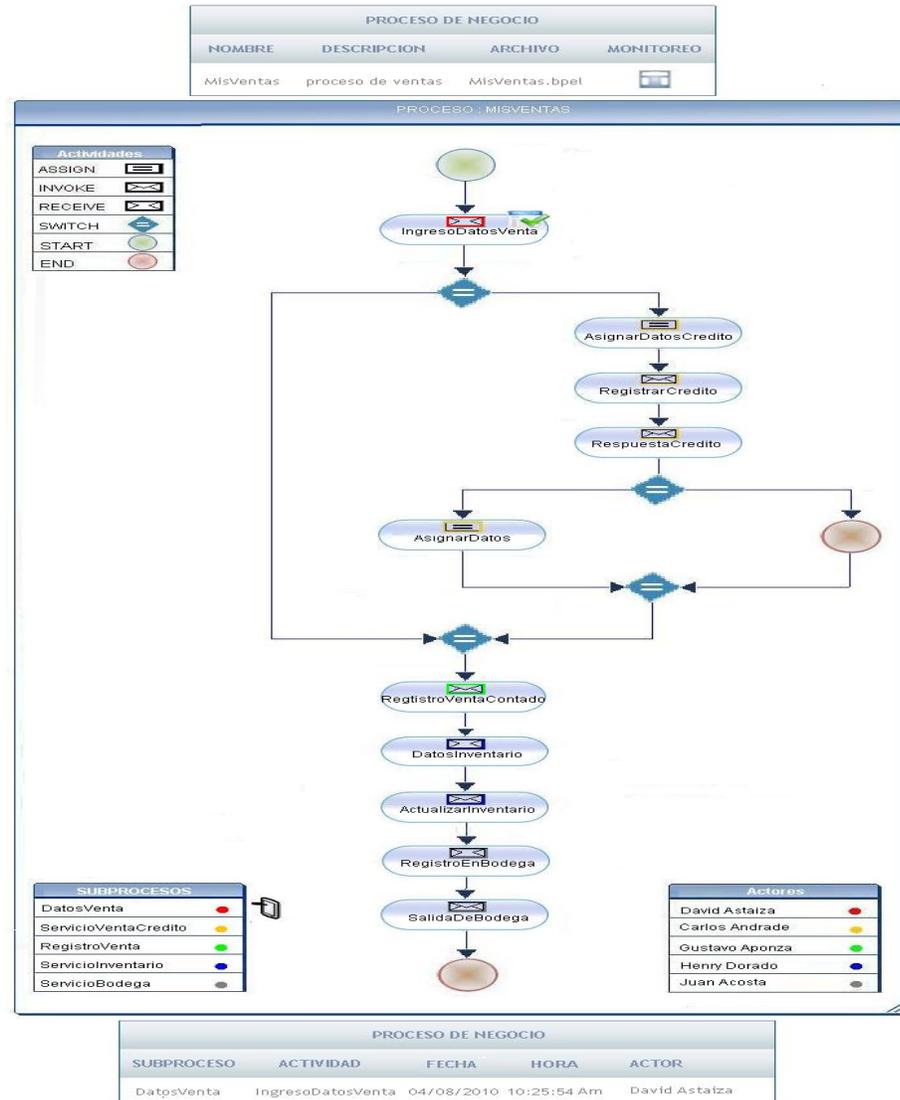


Figura B23. Interfaz gráfica para el monitoreo

Manual para el usuario de consulta

El usuario de consulta puede observar el estado de ejecución de un proceso de negocio particular para ello es necesario autenticarse, por lo tanto, la primera pantalla que vera es una interfaz como se muestra en la figura B11, posteriormente este usuario accede directamente a la interfaz de monitoreo figura B23, en donde puede Monitoreo los procesos que se encuentran en ejecución.

ANEXO C. FORMATO DE LOS PROCESOS BPEL Y WSDL SOPORTADOS POR LA PLATAFORMA MOBFLOW

Archivo BPEL

```
1<process name="BPELElectrodomesticosCE"
2  targetNamespace="http://xmlns.oracle.com/BPELElectrodomesticosCE"
3  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
4  xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
5  xmlns:ns1="http://www.globalcompany.com/ns/sales"
6  xmlns:ns2="http://localhost:8090/axis/RegistroVenta.jws"
7  xmlns:ns3="http://localhost:8090/axis/ServicioCredito.jws"
8  xmlns:ns4="http://localhost:8090/axis/ServicioInventario.jws"
9  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
10  xmlns:as="http://xmlns.oracle.com/BPELElectrodomesticosCE"
11  xmlns:ora="http://schemas.oracle.com/xpath/extension">
12<partnerLinks>
13  <partnerLink myRole="Vendedor" name="DatosVenta" partnerLinkType="ns1:TiendaElectrodomesticosCE" />
14  <partnerLink myRole="Cajero" name="RegistroVenta" partnerLinkType="ns2:RegistroVenta_PL" />
15  <partnerLink myRole="Analista_Credito" name="ServicioVentaCredito" partnerLinkType="ns3:ServicioCredito_PL" />
16  <partnerLink myRole="Encargado_Inventario" name="ServicioInventario"
    partnerLinkType="ns4:ServicioInventario_PL"/>
17</partnerLinks>
18<variables>
19  <variable messageType="as:DetallesVenta" name="inputVariable" />
20  <variable messageType="as:ResponseMessage" name="outputVariable" />
21  <variable messageType="as:RegistroVentaResponse" name="InvokeRegistroVentaVariableSalida" />
22  <variable messageType="as:RegistroCredito" name="InvokeServicioCreditoVariableEntrada" />
23  <variable messageType="as:ResponseCredito" name="ReceiveRespuestaCreditoVariableEntrada" />
24  <variable messageType="as:RegistroVentaResponse" name="ActualizarInventarioVariableEntrada" />
25</variables>
26<sequence name="main">
27  <receive name="entrada_Datos_de_Entrada" partnerLink="DatosVenta" portType="as:TiendaElectrodomesticosCE"
    operation="iniciarVenta" variable="inputVariable" createInstance="yes"/>
28  <switch name="TipoVenta">
29    <case condition="bpws:getVariableData('inputVariable','tipoVenta') = 'contado' ">
30      <sequence name="SequenceCase">
31        <invoke name="InvokeRegistroVentaContado" partnerLink="RegistroVenta" portType="as:RegistroVenta"
          operation="RegistrarVenta" variable="inputVariable" closeConnection="no" />
32      </sequence>
33    </case>
34    <otherwise>
35      <sequence name="SequenceAssingCredito">
36        <assign name="AssignDatosCredito">
37          <copy>
38            <from part="IDVentas" variable="inputVariable"/>
39            <to part="idcredito" variable="InvokeServicioCreditoVariableEntrada"/>
40          </copy>
41          <copy>
42            <from part="CodigoEan" variable="inputVariable"/>
43            <to part="codigoean" variable="InvokeServicioCreditoVariableEntrada"/>
44          </copy>
45          <copy>
46            <from part="Identificacion" variable="inputVariable"/>
46            <to part="id" variable="InvokeServicioCreditoVariableEntrada"/>
47          </copy>
48          <copy>
49            <from part="Cedula" variable="inputVariable"/>
50            <to part="cc" variable="InvokeServicioCreditoVariableEntrada"/>
51          </copy>
52          <copy>
53            <from part="tipoVenta" variable="inputVariable"/>
54            <to part="tipoVenta" variable="InvokeServicioCreditoVariableEntrada"/>
55          </copy>
56        </assign>

```

```

57     <copy>
58         <from part="Valor" variable="inputVariable"/>
59         <to part="valor" variable="InvokeServicioCreditoVariableEntrada"/>
60     </copy>
61     <copy>
62         <from part="Cuotas" variable="inputVariable"/>
63         <to part="cuotas" variable="InvokeServicioCreditoVariableEntrada"/>
64     </copy>
65 </assign>
66 <invoke name="InvokeRegistroVenta" partnerLink="ServicioVentaCredito" portType="as:ServicioCredito"
    operation="RegistrarCredito" variable="InvokeServicioCreditoVariableEntrada"/>
67 <receive name="Entrada_RespuestaCredito" partnerLink="ServicioVentaCredito" portType="as:ServicioCredito"
    operation="ResponderCredito" variable="ReceiveRespuestaCreditoVariableEntrada" createInstance="no"/>
68 <switch name="SwitchCredito">
69     <case condition="bpws:getVariableData('ReceiveRespuestaCreditoVariableEntrada','RespuestaCredito') = 'aprobado'">
70         <sequence name="Sequencecc2_1">
71             <assign name="AssignDatosVenta2">
72                 <copy>
73                     <from part="IDVentas" variable="inputVariable"/>
74                     <to part="IDVentas" variable="InvokeRegistroVentaVariableEntrada"/>
75                 </copy>
76                 <copy>
77                     <from part="CodigoEan" variable="inputVariable"/>
78                     <to part="CodigoEan" variable="InvokeRegistroVentaVariableEntrada"/>
79                 </copy>
80                 <copy>
81                     <from part="Identificacion" variable="inputVariable"/>
82                     <to part="Identificacion" variable="InvokeRegistroVentaVariableEntrada"/>
83                 </copy>
84                 <copy>
85                     <from part="Cedula" variable="inputVariable"/>
86                     <to part="Cedula" variable="InvokeRegistroVentaVariableEntrada"/>
87                 </copy>
88                 <copy>
89                     <from part="TipoVenta" variable="inputVariable"/>
90                     <to part="TipoVenta" variable="InvokeRegistroVentaVariableEntrada"/>
91                 </copy>
92                 <copy>
93                     <from part="Valor" variable="inputVariable"/>
94                     <to part="Valor" variable="InvokeRegistroVentaVariableEntrada"/>
95                 </copy>
96                 <copy>
97                     <from part="Cuotas" variable="inputVariable"/>
98                     <to part="Cuotas" variable="InvokeRegistroVentaVariableEntrada"/>
99                 </copy>
100            </assign>
101            <invoke name="InvokeRegistroVenta" partnerLink="RegistroVenta" portType="as:RegistroVenta"
    operation="RegistrarVenta" variable="inputVariable" closeConnection="no" />
102        </sequence>
103    </case>
104    <otherwise>
105        <sequence name="SequenceoTerminate">
106            <terminate name="TerminateProceso">
107            </terminate>
108        </sequence>
109    </otherwise>
110 </switch>
111 </sequence>
112 </otherwise>
113 </switch>

```

```

114 <receive name="ReceiveRegistroVenta" partnerLink="RegistroVenta" portType="as:RegistroVenta"
      operation="RegistrarVenta" variable="InvokeRegistroVentaVariableSalida" createInstance="no"/>
115 <invoke name="ActualizarInventario" partnerLink="ServicioInventario" portType="as:ServicioInventario"
      operation="ActualizarInventario" inputVariable="InvokeRegistroVentaVariableSalida"/>
116 </sequence>
117 </process>

```

Figura C1. Formato de archivo BPEL soportado por la plataforma y correspondiente a parte del proceso BPEL del caso de estudio

Archivo WSDL

```

1 <definitions name="TiendaElectrodomesticosCE"
2   targetNamespace="http://xmlns.oracle.com/TiendaElectrodomesticosCE/"
3   xmlns:ns1="http://www.globalcompany.com/ns/sales/"
4   xmlns:plnk="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
5   xmlns:tns="http://xmlns.oracle.com/TiendaElectrodomesticosCE/"
6   xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
7   xmlns:xsd="http://www.w3.org/2001/XMLSchema"
8   xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
9   xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
10  xmlns="http://schemas.xmlsoap.org/wsdl/">
11 <wsdl:message name="DetallesVenta">
12   <wsdl:part name="IDVentas" type="xsd:integer" />
13   <wsdl:part name="CodigoEan" type="xsd:string" />
14   <wsdl:part name="Identificacion" type="xsd:integer" />
15   <wsdl:part name="Cedula" type="xsd:integer" />
16   <wsdl:part name="tipoVenta" type="xsd:string" />
17   <wsdl:part name="Valor" type="xsd:sinteger" />
18   <wsdl:part name="Cuotas" type="xsd:integer" />
19 </wsdl:message>
20 <wsdl:message name="RegistroCredito">
21   <wsdl:part name="idcredito" type="xsd:integer" />
22   <wsdl:part name="codigoean" type="xsd:string" />
23   <wsdl:part name="id" type="xsd:integer" />
24   <wsdl:part name="cc" type="xsd:integer" />
25   <wsdl:part name="tipoVenta" type="xsd:integer" />
26   <wsdl:part name="valor" type="xsd:integer" />
27   <wsdl:part name="cuotas" type="xsd:integer" />
28 </wsdl:message>
29 <wsdl:message name="ResponseCredito">
30   <wsdl:part name="RespuestaCredito" type="xsd:string" />
31 </wsdl:message>
32 <wsdl:message name="RegistroVentaResponse">
33   <wsdl:part name="codigoProducto" type="xsd:string" />
34 </wsdl:message>
35 <wsdl:message name="ResponseMessage">
36   <wsdl:part name="OperExitosa" type="xsd:string" />
37 </wsdl:message>
38 <wsdl:portType name="TiendaElectrodomesticosCE">
39   <wsdl:operation name="iniciarVenta">
40     <wsdl:input message="tns:DetallesVenta"/>
41     <wsdl:output message="tns:ResponseMessage"/>
42   </wsdl:operation>
43 </wsdl:portType>
44 <wsdl:portType name="RegistroVenta">
45   <wsdl:operation name="RegistrarVenta">
46     <wsdl:input message="tns:DetallesVenta"/>
47     <wsdl:output message="tns:RegistroVentaResponse"/>
48   </wsdl:operation>
49 </wsdl:portType>
50 <wsdl:portType name="ServicioVentaCredito">
51   <wsdl:operation name="RegistrarCredito">
52     <wsdl:input message="tns:RegistroCredito"/>
53     <wsdl:output message="tns:ResponseCredito"/>
54   </wsdl:operation>

```

```

55 <wsdl:operation name="ResponderCredito">
56     <wsdl:input message="tns:ResponseCredito"/>
57     <wsdl:output message="tns:ResponseCredito"/>
58 </wsdl:operation>
59 </wsdl:portType>
60 <wsdl:portType name="ServicioInventario">
61     <wsdl:operation name="ActualizarInventario" parameterOrder="codigoProducto">
62         <wsdl:input message="tns:RegistroVentaResponse"/>
63         <wsdl:output message="tns:ResponseMessage"/>
64     </wsdl:operation>
65 </wsdl:portType>

70 <plnk:partnerLinkType name="TiendaElectrodomesticosCE">
71     <plnk:role name="Vendedor">
72         <plnk:portType name="tns:TiendaElectrodomesticosCE"/>
73     </plnk:role>
74 </plnk:partnerLinkType>
75 <plnk:partnerLinkType name="RegistroVenta_PL">
76     <plnk:role name="Cajero">
77         <plnk:portType name="tns:RegistroVenta"/>
78     </plnk:role>
79 </plnk:partnerLinkType>
80 <plnk:partnerLinkType name="ServicioCredito_PL">
81     <plnk:role name="Analista_Credito">
82         <plnk:portType name="tns:ServicioVentaCredito"/>
83     </plnk:role>
84 </plnk:partnerLinkType>

85 <plnk:partnerLinkType name="ServicioInventario_PL">
86     <plnk:role name="Encargado_Inventario">
87         <plnk:portType name="tns:ServicioInventario"/>
88     </plnk:role>
89 </plnk:partnerLinkType>

90 <wsdl:binding name="RegistroVentaSoapBinding" type="tns:RegistroVenta">
91     <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
92     <wsdl:operation name="RegistrarVenta">
93         <soap:operation soapAction="" />
94         <wsdl:input name="RegistrarVentaRequest">
95             <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace"
96                 use="encoded" />
97         </wsdl:input>
98         <wsdl:output name="RegistrarVentaResponse">
99             <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
100                 namespace="http://localhost:8090/axis/RegistroVenta.jws" use="encoded" />
101         </wsdl:output>
102     </wsdl:operation>
103 </wsdl:binding>

102 <wsdl:binding name="ServicioVentaCreditoSoapBinding" type="tns:ServicioVentaCredito">
103     <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
104     <wsdl:operation name="RegistrarCredito">
105         <soap:operation soapAction="" />
106         <wsdl:input name="RegistrarCreditoRequest">
107             <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" namespace="http://DefaultNamespace"
108                 use="encoded" />
109         </wsdl:input>
110         <wsdl:output name="RegistrarCreditoResponse">
111             <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
112                 namespace="http://localhost:8090/axis/ServicioCredito.jws" use="encoded" />
113         </wsdl:output>
114     </wsdl:operation>
115     <wsdl:operation name="ResponderCredito">
116         <soap:operation soapAction="" />
117         <wsdl:input name="ResponderCreditoRequest">
118             <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
119                 namespace="http://DefaultNamespace" use="encoded" />
120         </wsdl:input>
121         <wsdl:output name="ResponderCreditoResponse">
122             <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
123                 namespace="http://localhost:8090/axis/ServicioCredito.jws" use="encoded" />
124         </wsdl:output>
125     </wsdl:operation>
126 </wsdl:binding>

```

```

123 <wsdl:binding name="ServicioInventarioSoapBinding" type="tns:ServicioInventario">
124   <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http" />
125   <wsdl:operation name="ActualizarInventario">
126     <soap:operation soapAction="" />
127     <wsdl:input name="ActualizarInventarioRequest">
128       <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
129         namespace="http://DefaultNamespace" use="encoded" />
130     </wsdl:input>
131     <wsdl:output name="ActualizarInventarioResponse">
132       <soap:body encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
133         namespace="http://localhost:8090/axis/ServicioInventario.jws" use="encoded" />
134     </wsdl:output>
135   </wsdl:operation>
136 </wsdl:binding>

137 <wsdl:service name="RegistroVentaService">
138   <wsdl:port binding="tns:RegistroVentaSoapBinding" name="RegistroVenta">
139     <soap:address location="http://localhost:8090/axis/RegistroVenta.jws" />
140   </wsdl:port>
141 </wsdl:service>

142 <wsdl:service name="ServicioInventarioService">
143   <wsdl:port binding="tns:ServicioInventarioSoapBinding" name="ServicioInventario">
144     <soap:address location="http://localhost:8090/axis/ServicioInventario.jws" />
145   </wsdl:port>
146 </wsdl:service>

147 <wsdl:service name="ServicioVentaCreditoService">
148   <wsdl:port binding="tns:ServicioVentaCreditoSoapBinding" name="ServicioVentaCredito">
149     <soap:address location="http://localhost:8090/axis/ServicioCredito.jws" />
150   </wsdl:port>
151 </wsdl:service>
152 </definitions>

```

Figura C2. Formato de archivo WSDL soportado por la plataforma y correspondiente al proceso BPEL de la C1

Figura

En la Figura C1 y Figura C2 se presenta un proceso BPEL junto con su WSDL que realiza el caso de estudio presentado en este proyecto. A continuación se presentan las consideraciones que se deben tener para formar un documento BPEL y WSDL que tenga el formato requerido por la plataforma a partir de los presentados en las figuras ya nombradas.

Archivo BPEL

- Los espacios de nombre (*namespace*) presentados en las líneas 5, 6, 7 y 8 deben ir asociados a los diferentes *partnerLinks* definidos en el proceso, de manera tal que estos *namespace* indiquen la *URL* donde se encuentra desplegado el servicio Web asociado a determinado *partnerLink*, en caso de que el *partnerLink* no tenga asociado Web no hay problema, pero debe tener un *namespace* asociado. La asociación del *namespace* con el *partnerLink* se realiza mediante el atributo *partnerLinkType* en la etiqueta donde se define el *partnerLink* (presentado mas adelante).
- El *namespace* "xmlns:as" (línea10) es requerido de manera obligatoria en la definición del proceso para el correcto funcionamiento de los subprocesso y lo asignado a esté *namespace* debe coincidir exactamente con el *namespace* "xmlns:tns" del archivo WSDL del proceso(Línea 5 archivo WSDL). Es decir, si en el BPEL se define el namespace xmlns:as de la siguiente manera "xmlns:as=http://mobflow.com" en el archivo WSDL debe definirse como "xmlns:tns=http://mobflow.com".
- Luego de los *namespace* se deben definir los diferentes *partnerLinks* del proceso.

- En la definición de los *partnerLinks* debe seguir el orden de los atributos como los presentado en las líneas de 12 a 17, es decir en la etiqueta *<partnerLink>* se define primero el atributo “*myRole*”, luego el atributo “*name*” y finalmente el atributo “*partnerLinkType*”.
- Es importante recordar que con el atributo “*myRole*” la plataforma MobFlow obtiene los diferentes roles del proceso y estos son los que se vinculan con los actores de la empresa para su posterior distribución, por lo cual es importante tener cuidado con los valores asignados.
- El atributo *partnerLinkType* debe definirse con el espacio de nombres (*namespace*) correspondiente a donde está desplegado el servicio Web asociado a esté. Por ejemplo el *partnerLink RegistroVenta* está ligado al ns2 (que indica donde esta desplegado el servicio Web asociado) por lo tanto el atributo *partnerLinkType* para el *partnerLink RegistroVenta* se define *partnerLinkType="ns2:RegistroVenta_PL"*.
- Las variables deben ir especificadas como esquemas XML en el archivo WSDL (Ejemplo Líneas 11 a 37
- las definidas en el archivo BPEL (ejemplo: líneas 18 a 25 figura 1) se establecen según las especificadas en el WSDL. Estas se declaran primero el atributo “*messageType*” (que define el tipo de variable según el WSDL) y luego con el atributo “*name*”.
- El valor asignado al atributo *messageType* debe ir relacionado con el *namespace xmlns:as* como se muestra en las líneas del 19 a 24 de la Figura C1. Proceso BPEL con formato soportado por la plataforma (BPEL prototipo Caso de estudio) para todas las variables BPEL declaradas. Por ejemplo *messageType="as:DetallesVenta"*.
- Solo se debe relacionar una variable con un *messageType*, es decir en la declaración de variables no pueden haber dos con el mismo *messageType*.
- La definición de las etiquetas para las actividades básicas deben ir en el
- En la definición de las etiquetas para las actividades básicas *<receive>*, *<invoke>* y *<reply>*, los atributos deben ir en el siguiente orden: “*name*”, “*partnerLink*”, “*portType*”, “*operation*” y “*variable*” (Ejemplo línea 27).
- El valor asignado al atributo “*portType*” en estas actividades debe estar relacionado con el *namespace “xmlns:as”*, por ejemplo: *portType="as:TiendaElectrodomesticos"* (Ejemplo Línea 27).
- No deben existir actividades con el mismo nombre
- Las actividades realizadas dentro de una actividad estructurada deben estar entre una etiqueta *<sequence>* (Ejemplo actividad dentro del switch Línea 30).
- Si la ejecución de un servicio Web mediante una etiqueta *<invoke>* es de naturaleza síncrona, debajo de la actividad que realiza la invocación debe ir una actividad *<receive>* para recibir el resultado de dicha invocación (Ejemplo líneas 66 y 67).
- Las etiquetas que generan interfaz gráfica para el ingreso de datos en el sistema móvil de información deben tener asignado el atributo *name* un nombre que empiece con la palabra entrada por ejemplo *name=entrada_XXX*. Estas deben ser actividades *receive* (Ejemplo línea 27).

- Es importante tener en consideración que los nombres de las actividades asignado mediante el atributo *name*, son los que aparecen en las imágenes del flujo del proceso, ya sea para visualización o monitoreo.

Archivo WSDL

- En la definición de lo espacios de nombre (*namespace*) en el archivo WSDL debe ir un “*xmlns:tns*”, con una asignación similar al “*xmlns:as*” del archivo BPEL como ya se había explicado (Ejemplo línea 5).
- En primer lugar se deben establecer los *wSDL:message* que corresponden a las variables del proceso BPEL, y también se relacionan con las entradas y salidas de los servicios Web socios del proceso (Ejemplo línea 11).
- Luego de la definición de los *wSDL:message* se debe realizar la definición de los *portType*, que van ligados a los definidos en los servicios Web socios del proceso. Dentro de los *portType* se establecen las operaciones *operation* relacionadas que son las disponibles dentro de determinado servicio Web. En la definición de las etiquetas básicas en el archivo BPEL estas deben estar relacionada con los *PortType* y *Operations* aquí definidas.
- Los datos de entrada y salidas de las operaciones *operation* se definen asociándolos con los *wSDL:message* el dato asignado al atributo de entrada y de salida, deben ir asociados al *xmlns:tns*. Por ejemplo `<wSDL:input message = “tns:Ventas”>`. Toda operación igualmente debe tener definida una entrada y una salida (Ejemplo Línea 38).
- Los *partnerLinkTypes* establecen los roles del proceso según el *portType* estos deben estar relacionados con los definidos en las etiquetas de los *partnerLinks* del archivo BPEL (Ejemplo Línea 70).
- Dentro del *partnerLinkType* se establece un rol mediante la etiqueta `<plnk:role>` y dentro de este último se define un *portType* relacionado a el rol. Este *portType* debe ser definido asociándolo al “*xmlns:tns*”. Por ejemplo `<plnk:portType name = tns:PortVenta>`(Ejemplo Línea 72).
- El resto de etiquetas del archivo WSDL no requieren consideraciones específicas.

ACTIVIDADES BPEL SOPORTADAS

Actividades Básicas y Formato requerido

<ul style="list-style-type: none"> • <receive> 	<pre><receive partnerLink="ncname" portType="as:qname" operation="ncname" variable="ncname"? createInstance="yes no"? standard-attributes> standard-elements <correlations>? <correlation set="ncname" initiate="yes no"?>+ </correlations> </receive></pre>
<ul style="list-style-type: none"> • <invoke> 	<pre><invoke partnerLink="ncname" portType="as:qname" operation="ncname" inputVariable="ncname"? outputVariable="ncname"? standard-attributes> standard-elements <correlations>? <correlation set="ncname" initiate="yes no"? pattern="in out out-in"/>+ </correlations> </invoke></pre>
<ul style="list-style-type: none"> • <reply> 	<pre><reply partnerLink="ncname" portType="as:qname" operation="ncname" variable="ncname"? faultName="qname"? standard-attributes> standard-elements <correlations>? <correlation set="ncname" initiate="yes no"?>+ </correlations> </reply></pre>
<ul style="list-style-type: none"> • <assing> <ul style="list-style-type: none"> ○ <copy> <ul style="list-style-type: none"> ▪ <from> ▪ <to> 	<pre><assign standard-attributes> standard-elements <copy>+ <from-spec> <to-spec> </copy> </assign></pre>
<ul style="list-style-type: none"> • <empty> 	<pre><empty standard-attributes> standard-elements </empty></pre>

Tabla C1. Actividades Básicas y Formato requerido

Actividades Estructuradas y Formato requerido

<ul style="list-style-type: none"> • <sequence> 	<pre><sequence standard-attributes> standard-elements activity+ </sequence></pre>
<ul style="list-style-type: none"> • <switch> <ul style="list-style-type: none"> ○ <case> ○ <otherwise> 	<pre><switch standard-attributes> standard-elements <case condition="bool-expr">+ activity </case> <otherwise?> activity </otherwise> </switch></pre>

<ul style="list-style-type: none"> • <pick> <ul style="list-style-type: none"> ○ <onmessage> 	<pre><pick createInstance="yes no"? standard-attributes> standard-elements <onMessage partnerLink="NCName" portType="QName"? operation="NCName" variable="BPEL VariableName"? messageExchange="NCName"?>+ <correlations?> <correlation set="NCName" initiate="yes join no"? />+ </correlations> <fromParts?> <fromPart part="NCName" toVariable="BPEL VariableName" />+ </fromParts> activity </pick></pre>
<ul style="list-style-type: none"> • <flow> 	<pre><flow standard-attributes> standard-elements <links?> <link name="ncname">+ </links> activity+ </flow></pre>
<ul style="list-style-type: none"> • <while> 	<pre><while condition="bool-expr" standard-attributes> standard-elements activity </while></pre>

Tabla C2. Actividades Estructuradas y Formato requerido

Alcances (Scopes) y Formato requerido

<ul style="list-style-type: none"> • <scope> 	<pre><scope> activity+ </scope></pre>
<ul style="list-style-type: none"> • <eventHandlers> <ul style="list-style-type: none"> ○ <onmessage> 	<pre><eventHandlers> <onMessage operation="NCName" portType="as:QName" partnerLink="NCName" variable="BPEL VariableName"? <correlations?> <correlation set="NCName" initiate="yes join no"? />+ </correlations> activity+ </onMessage> </eventHandlers></pre>

Tabla C3. Alcances y Formato requerido

ANEXO D. EJEMPLOS DETALLADOS PARA LOS ALGORITMOS PROPUESTOS

```

1 <process name="Operaciones" targetNamespace="http://xmlns.oracle.com/ Operaciones "
2   xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
3   xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
4   xmlns:ns1="http://localhost:8090/axis/Operadora.jws" xmlns:ns3="http://localhost:8090/axis/Restadora.jws"
5   xmlns:ns2="http://localhost:8090/axis/Sumadora.jws" xmlns:as="http://xmlns.oracle.com/Promedio2Nums">
6
7   <partnerLinks>
8     <partnerLink myRole="Operar" name="Operacion" partnerLinkType="ns1:Operacion_PL"/>
9     <partnerLink myRole="Sumar" name="Sumadora" partnerLinkType="ns2:Sumadora_PL"/>
10    <partnerLink myRole="Restar" name=" Sustractora" partnerLinkType="ns3:Restadora_PL"/>
11  </partnerLinks>
12
13  <variables>
14    <variable messageType="as:Input" name="Entrada" />
15  </variables>
16
17  <sequence name="main">
18    <receive name="EntradaDatos" partnerLink=" Operacion " portType="as:Opera" operation="operar" variable="Entrada"/>
19    <switch name="DatoEntrada">
20      <case condition="bpws:getVariableData('inputVariable','num1') > bpws:getVariableData('inputVariable','num2') ">
21        <invoke name="Suma" partnerLink="Sumadora" portType="as:Suma" operation="sumar" variable="Entrada"/>
22      </case>
23      <otherwise>
24        <invoke name="Resta" partnerLink="Sustractora" portType="as:Resta" operation="restar" variable="Entrada"/>
25      </otherwise>
26    </switch>
27  </sequence>
28 </process>

```

FiguraD1. Ejemplo de un proceso BPEL que realiza operaciones de suma o resta según los datos de entrada Operaciones.bpel

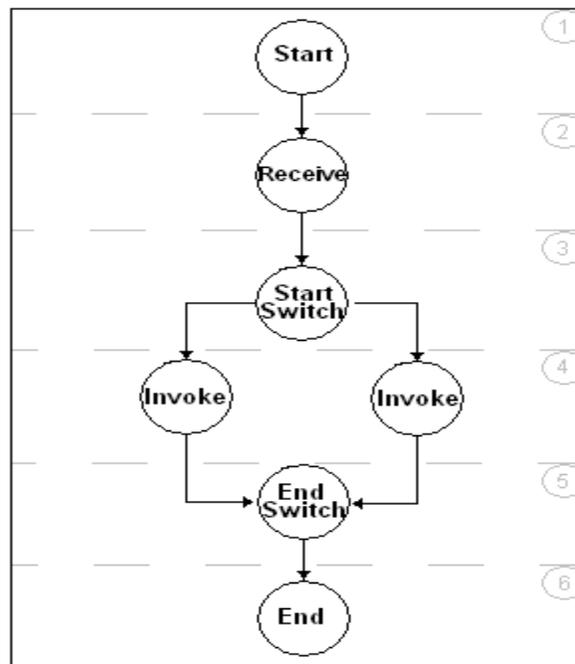


Figura D2. Grafo que representa el proceso BPEL presentado en la D1

Algoritmo 1: Inicio del Particionamiento

```
INPUT: BPELXML
OUTPUTS: SubBPELXML1, Sub BPELXML2...SubBPELXMLN
1 BEGIN
2 Obtener todos los partnerLinks del proceso XMLBPEL
3 FOR Cada partnerLink Obtenido del proceso XMLBPEL
4     Obtener el atributo nombre del partnerLinki
5     Crear objeto XML (SubBPELXMLi) para contener actividades del subproceso
6     Convertir BPELXML a Modelo de Grafos
7     Ubicar el primer nodo en el grafo con atributo partnerLink igual nombre partnerLinki
   (Representa el primer nodo del subproceso a generar)
8     IF NO es el primer subproceso a crear THEN
9         Insertar actividad receive de sincronización en SubXMLBPELi con los siguientes datos
           TipoActividad = Receive
           Nombre Actividad = sincronizacion
           PartnerLink = Obtener atributo PartnerLink Nodo actual
           PortType = Obtener atributo PartnerLink Nodo actual
           Operación = Obtener atributo PartnerLink Nodo actual
           Variable = Obtener atributo Variable Nodo actual
           (Si existen más atributos obtenerlos del nodo y ponerlo con la etiqueta)
10    END IF
11    SubBPELXMLi ← Llamar Reglas Particionamiento
12 END FOR
13 END
```

Algoritmo 2: Reglas de Particionamiento

```
INPUTS: GBPEL, Nodoi...j, SubBPELXML1...n.
OUTPUT: SubBPELXML
1 BEGIN
2 IF Tipo actividad que representa Nodoj es una actividad tipo End THEN
3     Finalizar → ir a END
4 ELSE
5     IF Tipo actividad que representa Nodoj es una actividad básica THEN
6         IF Atributo partnerLink Nodoj = Nombre subproceso a particionar Ó Nodoj no tiene atributo partnerLink THEN
7             Obtener información correspondiente a etiqueta del Nodoj (nodo actual)
8             Insertar información obtenida en SubBPELXMLi
9             Ubicar siguiente nodo
10            Llamar a Reglas Particionamiento con nodo ubicado en línea anterior
              (Los otros parámetros iguales)
11        ELSE
12            Insertar actividad invoke de sincronización en SubBPELXMLi con los siguientes datos:
              TipoActividad = Invoke
              Nombre Actividad = sincronizacion
              PartnerLink = Obtener nombre del PartnerLink del Nodo
              PortType = Obtener atributo PartnerLink Nodo actual
              Operación = Obtener atributo PartnerLink Nodo actual
13        Finalizar → ir a END
14    END IF
15 ELSE
16    SWITCH Tipo actividad que representa Nodoj
17        CASE "Switch"
18            Crear objeto XML para contener las actividades del Switch (XMLSwitch)
19            Recorrer desde StarSwitch hasta EndSwitch y obtener actividades, aplicando las reglas definidas del Switch
20            Insertar actividades en objetoXML Switch
21            Insertar de XMLSwitch al objeto XML SubBPELXMLi
22        CASE "While"
23            ...
24        CASE "Pick"
25            ...
26        CASE "Flow"
27            ...
```

```
25     END SWITCH
26     END IF
27     Ubicar siguiente nodo
28     Llamar a Reglas Particionamiento con nodo ubicado (Los otros parámetros iguales)
29     END IF
30 END
```

Teniendo como entrada esté proceso al algoritmo 1, los partnerLink obtenidos al efectuar la sentencia presentada en la línea 2, son los correspondientes a las líneas 10 a 12 de la figura 1 y sus nombres son *Operacion*, *Sumadora* y *Sustractora*.

El proceso BPEL presentado en la figura D1, al realizar el proceso de transformación se obtiene un grafo con la estructura similar al presentado en la figura D2.

Teniendo como entrada el proceso BPEL de la figura D1, en la primera iteración del ciclo FOR del algoritmo 1, se obtiene el subproceso correspondiente a las actividades del partnerLink *Operacion*, el primer nodo ubicado por la sentencia presentada en la línea 7 en esta iteración es el nodo *receive*, ya que este tiene el atributo partnerLink con el nombre igual a *Operacion* (nombre del partnerLink sobre el cual se está realizando el particionamiento).

Debido a que es el primer subproceso a crear no se le inserta actividad *receive* de sincronización, es decir no entra en la condición expuesta en la línea 9 del algoritmo y pasa directamente a la línea 18 que hace la invocación al algoritmo 2.

La entrada en este caso para el algoritmo 2, es el grafo presentado en la figura D2, el nodo *receive* (nodo actual en el recorrido del grafo) y el objeto XML que contiene las actividades del partnerLink *Operacion* (Hasta el momento no se le han insertado actividades).

Teniendo esta entrada la condición de la línea 2 de este algoritmo no se cumple debido a que el nodo no representa una actividad de finalización, por lo cual va a línea 5 condición que si cumple al igual que la siguiente condición de la línea 6. Caso en el cual la información que se obtiene del nodo mediante sus atributos es: tipo actividad: *receive*, nombre actividad: *EntradaDatos*, partnerLink: *Operación*, *portType*: *as:Opera*, *operation*: *operar* y *variable*: *Entrada*. Con esto se crea una etiqueta para la actividad y se inserta en el objeto XML.

El siguiente nodo en ser ubicado es el nodo *StartSwitch* (línea 9), el cual como ya se mencionó representa la actividad estructurada *Switch* del proceso BPEL y la siguiente línea (10) hace un llamado recursivo nuevamente al algoritmo 2 con las siguientes entradas, el grafo presentado en la figura D2, el nodo *StartSwitch* (nodo actual en el recorrido del grafo) y el objeto XML que contiene las actividades del partnerLink *Operacion*.

Por la evaluación de las condiciones, con esta entrada se ejecuta la línea 16 de la sentencia *Switch* y para este caso se ejecuta el caso de la línea 18. En este caso se crea un objeto XML para que contenga las actividades de propias del *Switch* y se hace el recorrido desde el *StartSwitch* al *EndSwitch*, por cada nodo encontrado se obtiene información de sus atributos de actividades y se aplican las reglas definidas para los nodos contenidos dentro de la actividad estructurada *Switch*. Los nodos que se encuentran dentro del *StartSwitch* y *EndSwitch* son nodos que representan actividades *invoke* con el nombre del atributo partnerLink diferente a *Operacion*, por consiguiente se crean actividades *invoke* de sincronización y se insertan en el objeto XML creado para contener las actividades del *Switch*.

Luego la información obtenida del objeto XML del *Switch* se inserta en el objeto XML que contiene las actividades propias del partnerLink *Operacion* y se ubica el siguiente nodo para nuevamente hacer un llamado recursivo. Debido a que el siguiente nodo representa una actividad de finalización termina la ejecución del algoritmo 2 y se obtiene un objeto XML con las actividades requeridas para la ejecución y

sincronización de las tareas del partnerLink *Operacion*. La información contenida en este objeto es presentada en la figura D3.

```

1 <sequence name="main">
2   <receive name="EntradaDatos" partnerLink="Operadora" portType="as:Opera" operation="operar" variable="Entrada"/>
3   <switch name="DatoEntrada">
4     <case condition="bpws:getVariableData('inputVariable','num1') > bpws:getVariableData('inputVariable','num2') ">
5       <invoke name="sincronizacion" partnerLink="Sumadora" portType="as:Suma" operation="sumar" variable="Entrada"/>
6     </case>
7     <otherwise>
8       <invoke name=" sincronizacion " partnerLink="Sustractora" portType="as:Resta" operation="restar" variable="Entrada"/>
9     </otherwise>
10  </switch>
11 </sequence>

```

Figura D3. Información contenida en el Objeto XML (SubBPEL_{XML1}) resultante de la primera iteración del algoritmo1

En la segunda iteración del ciclo FOR del algoritmo 1, efectuada por la línea 3, el nombre del partnerLink que se obtiene es *Sumadora*, igualmente se crea un objeto XML para contener las actividades propias de este partnerLink (SubBPEL_{XML2}). El nodo ubicado en el grafo por la línea 7 es el nodo invoke de la derecha en la figura D2. Para este caso sí se crea un actividad receive de sincronización. En la entrada para el algoritmo 2 se tiene grafo presentado en la figura D2, el nodo invoke de dicho grafo y el objeto XML que contiene las actividades para el partnerLink *Sumadora*, por las condiciones planteadas en el algoritmo 2 con estas entrada se ejecuta en primer lugar la sentencia de la línea 7 donde se obtiene la información del nodo y se inserta en SubBPEL_{XML2}. Luego se ubica el siguiente nodo y se realiza el llamado recursivo, como el siguiente nodo es una actividad de finalización termina el algoritmo para este partnerLink. Obteniendo el Objeto XML presentado en la figura D4.

```

<sequence name="main">
  <receive name=" sincronizacion" partnerLink=" Sumadora " portType="as: Suma " operation=" sumar " variable="Entrada"/>
  <invoke name=" Suma " partnerLink="Sumadora" portType="as:Suma" operation="sumar" variable="Entrada"/>
</sequence>

```

Figura D4. Información contenida en el Objeto XML (SubBPEL_{XML2}) resultante de la segunda iteración del algoritmo1

Finalmente obtenidos todos los subprocesos correspondientes en los objetos XML, se procede a leer el documento BPEL original y se obtiene la información de la etiqueta raíz del proceso *<process>* con sus correspondientes espacios de nombre XML (XML namespaces), y se inserta en cada uno de los objetos XML creados con la etiqueta de cierre *</process>*. Igualmente se obtiene la información de las etiquetas partnerLinks y variables, las cuales se insertan en cada uno de estos objetos XML obtenidos en el lugar correspondiente dentro de la etiqueta raíz.

Al finalizar la ejecución del algoritmo y realizando los aspectos que finalizan el algoritmo se obtiene como resultado los subprocesos BPEL presentados en las figuras D5, D6 y D7.

```

<process name="Promedio2Nums" targetNamespace="http://xmlns.oracle.com/Promedio2Nums"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:ns1="http://localhost:8090/axis/Operadora.jws" xmlns:ns3="http://localhost:8090/axis/Restadora.jws"
  xmlns:ns2="http://localhost:8090/axis/Sumadora.jws" xmlns:as="http://xmlns.oracle.com/Promedio2Nums">
  <partnerLinks>
    <partnerLink myRole="Operar" name="Operacion" partnerLinkType="ns1:Operacion_PL"/>
    <partnerLink myRole="Sumar" name="Sumadora" partnerLinkType="ns2:Sumadora_PL"/>
    <partnerLink myRole="Restar" name="Sustractora" partnerLinkType="ns3:Restadora_PL"/>
  </partnerLinks>
  <variables>
    <variable messageType="as:Request" name="Entrada" />
  </variables>
  <sequence name="main">
    <receive name="EntradaDatos" partnerLink="Operadora" portType="as:Opera" operation="operar" variable="Entrada"/>
    <switch name="DatoEntrada">
      <case condition="bpws:getVariableData('inputVariable','num1') > bpws:getVariableData('inputVariable','num1') ">
        <invoke name="sincronizacion" partnerLink="Sumadora" portType="as:Suma" operation="sumar" variable="Entrada"/>
      </case>
      <otherwise>
        <invoke name="sincronizacion" partnerLink="Sustractora" portType="as:Resta" operation="restar" variable="Entrada"/>
      </otherwise>
    </switch>
  </sequence>
</process>

```

Figura D5. SubProceso Operacion.bpel

```

<process name="Promedio2Nums" targetNamespace="http://xmlns.oracle.com/Promedio2Nums"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:ns1="http://localhost:8090/axis/Operadora.jws" xmlns:ns3="http://localhost:8090/axis/Restadora.jws"
  xmlns:ns2="http://localhost:8090/axis/Sumadora.jws" xmlns:as="http://xmlns.oracle.com/Promedio2Nums">
  <partnerLinks>
    <partnerLink myRole="Operar" name="Operacion" partnerLinkType="ns1:Operacion_PL"/>
    <partnerLink myRole="Sumar" name="Sumadora" partnerLinkType="ns2:Sumadora_PL"/>
    <partnerLink myRole="Restar" name="Sustractora" partnerLinkType="ns3:Restadora_PL"/>
  </partnerLinks>
  <variables>
    <variable messageType="as:Request" name="Entrada" />
  </variables>
  <sequence name="main">
    <receive name="sincronizacion" partnerLink="Sumadora" portType="as:Suma" operation="sumar" variable="Entrada"/>
    <invoke name="Suma" partnerLink="Sumadora" portType="as:Suma" operation="sumar" variable="Entrada"/>
  </sequence>
</process>

```

Figura D6. SubProceso Sumadora.bpel

```

<process name="Promedio2Nums" targetNamespace="http://xmlns.oracle.com/Promedio2Nums"
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-process/" xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:bpws="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
  xmlns:ns1="http://localhost:8090/axis/Operadora.jws" xmlns:ns3="http://localhost:8090/axis/Restadora.jws"
  xmlns:ns2="http://localhost:8090/axis/Sumadora.jws" xmlns:as="http://xmlns.oracle.com/Promedio2Nums">
  <partnerLinks>
    <partnerLink myRole="Operar" name="Operacion" partnerLinkType="ns1:Operacion_PL"/>
    <partnerLink myRole="Sumar" name="Sumadora" partnerLinkType="ns2:Sumadora_PL"/>
    <partnerLink myRole="Restar" name="Sustractora" partnerLinkType="ns3:Restadora_PL"/>
  </partnerLinks>
  <variables>
    <variable messageType="as:Request" name="Entrada" />
  </variables>
  <sequence name="main">
    <receive name="sincronizacion" partnerLink="Sustractora" portType="as:Resta" operation="restar" variable="Entrada"/>
    <invoke name="Resta" partnerLink="Sustractora" portType="as:Resta" operation="restar" variable="Entrada"/>
  </sequence>
</process>

```

Figura D7. SubProceso Sustractora.bpel

Algoritmo 3: Ejecución actividades sincronización

```
INPUT: BPEL_XML, WSDL_XML
OUTPUTS: String Mensaje
1 BEGIN
2 Inicializar variables: VariableRequerida (Objeto XML) ← NULO.
                       MensajeSalida (String) ← NULO.

3 Obtener la primera actividad del proceso BPEL_XML
4 IF (Tipo de primera Actividad del proceso BPEL_XML = receive Y Atributo Nombre primera Actividad del proceso
   BPEL_XML = sincronizacion)
5     Obtener atributo variable de primera Actividad del proceso BPEL_XML
6     Enviar petición a plataforma Web solicitando variables de sincronización agregando como parámetro en la petición
   atributo variable obtenido en el paso anterior
7     Obtener respuesta de plataforma Web
8     IF (Respuesta obtenida NO es mensaje de ERROR)
9         Asignar a VariableRequerida mensaje obtenido de plataforma Web
           VariableRequerida ← Respuesta obtenida de plataforma Web
10    ELSE
11        Asignar a MensajeSalida valor de "Error de ejecución"
           MensajeSalida ← "Error de Ejecución"
12        Imprimir en pantalla MensajeSalida
13        Finalizar Algoritmo. Ir a línea 24
14    END IF
15 END IF
16 Ejecutar actividades del proceso BPEL_XML junto con WSDL_XML en motor BPEL teniendo en consideración que:
    ○ Si fue valida la condición de la sentencia 4, ejecutar Actividad 2 del proceso BPEL_XML asignando lo obtenido en
      VariableRequerida. a los datos de variable de esta actividad.
    ○ Si existen procesos de asignación de datos a variable en la ejecución de actividades del proceso BPEL_XML registrarlos
      en la plataforma Web.
    ○ Al ejecutar una actividad invoke o receive (no de sincronización) o reply que requiera invocación de un Servicio Web
      enviar petición a plataforma Web solicitando ejecución del servicio, obteniendo los siguiente parámetros para el
      envió en la petición.
      ○ URL donde está desplegado el Servicio Web
      ○ Atributo operation de la etiqueta
      ○ Parametros obtenidos de los datos asignados a la variable relacionada con la actividad invoke actual.
    ○ Si actividad es invoke de sincronización ir a línea 17.

17 IF (Tipo Actividad del proceso BPEL_XML = invoke Y Atributo nombre de Actividad = sincronizacion)
18     Obtener atributo PartnerLink de ActividadActual
19     Enviar petición a plataforma Web solicitando número de dispositivo de siguiente en la ejecución agregando parámetro
   PartnerLink
20     Número ← Obtener Respuesta de plataforma Web
21     Enviar SMS a Número Obtenido indicando que debe iniciar la ejecución de su subproceso
22 END IF
23 Finalizada la ejecución del proceso presentar mensaje de ejecución exitosa
   MensajeSalida ← "Ejecución Exitosa"
   Imprimir en pantalla MensajeSalida
24 END
```

Si se tiene como entrada al algoritmo 3 el subproceso BPEL presentado en la figura D5 Operaciones.bpel, al obtener la primera actividad (`<receive name="EntradaDatos" partnerLink="Operadora" portType="as:Opera" operation="operar" variable="Entrada"/>`) (línea 3), ya que no es una actividad de sincronización (Línea 4), es manejada por el motor BPEL, en el caso de esta actividad se realiza un proceso de asignación a la variable "Entrada", según los datos introducidos por el usuario, los cuales son enviados al Servidor Web para su registro. Para este caso vamos a suponer que se ingresaron los valores de 5 y 6 a los campos requeridos por este subproceso, por lo cual la información respecto a la asignación de variables es como la presentada en la figura D8.

```

<n0: operar xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:n0="http://xmlns.oracle.com/Operaciones/">
  <num1 xsi:type="xsd:double">5</num1>
  <num2 xsi:type="xsd:double">6</num2>
</n0: operar>

```

Figura D8. Datos asignados por el ingreso de datos

La información registrada en la plataforma es el nombre de variable, en este caso *Entrada* y sus datos asignados que son los presentados en la figura D8.

Luego se continúa con la ejecución del subproceso, debido a la actividad *<Switch>* y por los datos de entrada se ejecuta la actividad (*<invoke name="sincronizacion" partnerLink="Sustractora" portType="as:Resta" operation="restar" variable="Entrada"/>*).

Ya que esta actividad cumple la condición de la línea 18, al ejecutarse la línea 19 se obtiene el atributo *partnerLink* asociado, en este caso el valor asignado a este atributo es el *Sustractora*, con este dato se envía una petición al servidor web MobFlow para obtener el número de teléfono asociado al *partnerLink Sumadora*. Obtenido el teléfono se le envía un mensaje de texto al número obtenido.

Finalmente ya que la ejecución fue exitosa se le asigna el valor *"Finalización exitosa"* a la variable *mensajeSalida*, para que sea impresa en pantalla.

Siguiendo el flujo, en la ejecución del subproceso presentado en la figura D7 *Sustractora.bpel* teniéndolo como entrada en el algoritmo 3, se tiene que la primera actividad definida en este subproceso es una actividad *receive* de sincronización (*<receive name="sincronizacion" partnerLink="Sumadora" portType="as:Suma" operation="sumar" variable="Entrada"/>*), por lo cual se cumple la condición planteada en la línea 4. Por la línea 5 del algoritmo el atributo *variable* asignado a esta actividad es *"Entrada"*, por lo cual se le envía una petición al servidor Web solicitándole la información de asignación del último registro con nombre igual a *"Entrada"* (Dato enviado). Para este caso en la ejecución del anterior subproceso *Operaciones.bpel* se había registrado esta información en el servidor Web (sería el último registro existente), por lo cual la plataforma devuelve datos iguales a los presentados en la figura D8. Obtenidos estos datos (Línea 7) y ya que la respuesta obtenida no fue un mensaje de error (cumple condición de la Línea 8), esta respuesta es asignada a la variable *variableRequerida*.

Luego se continúa con la ejecución del subproceso (Línea 16), y considerando que la siguiente actividad (*<invoke name="Resta" partnerLink="Sustractora" portType="as:Resta" operation="restar" variable="Entrada"/>*) requiere la invocación de un servicio Web según las consideraciones propuestas (Línea 16) se tendría lo siguiente:

- Como fue válida la condición de la línea 4, al atributo *variable* de esta actividad se le asigna lo contenido en la *variableRequerida*, que es igual a lo presentado en la línea figura D8.
- Como es una actividad *invoke* que requiere invocación de un Servicio Web se le envía una petición a plataforma Web solicitando ejecución del servicio, los parámetros obtenidos según el proceso BPEL son los siguientes:
 - **URL** donde está desplegado el Servicio Web = *http://localhost:8090/axis/Restadora.jws* debido a que el *partnerLink* asociado esta actividad está relacionado con el *xmlns:ns3* de la cabecera del proceso.

- El atributo *operation* de la etiqueta es “*restar*”
- Los parámetros obtenidos de los datos asignados a la variable relacionada con la actividad invoke actual son los presentados en la figura D8

Finalmente debido a que el proceso no tiene actividad invoke de sincronización y no se presentaron errores en la ejecución de le asigna el valor de “*Ejecución Exitosa*” a la variable *MensajeSalida* para que sea presentada en pantalla.

Algoritmo 4: Visualización del proceso

```

INPUT: BPELXML, LISTASUBPROCESOS, LISTAACTORES, LISTAACTIVIDADES, boolean: BanderaMonitoreo
OUTPUTS: IMG_BPEL
1 BEGIN
2   Crear imagen en blanco IMG_BPEL
3   Convertir BPELXML a Modelo de Grafos se obtiene GBPEL
4   IF BanderaMonitoreo = TRUE THEN
5     Asignar de manera aleatoria un color a cada actor y subproceso relacionado, contenidos en LISTAACTORES y
      LISTASUBPROCESOS respectivamente
6     Asignar un color a cada Nodo del grafo de acuerdo al color del subprocesa a que pertenece
7     Dibujar una tabla en IMG_BPEL con los subprocesos contenidos en LISTASUBPROCESOS y con el color
      asignado a cada subproceso.
8     Dibujar una tabla en IMG_BPEL con los actores contenidos en LISTAACTORES y con el color
      asignado a cada actor.
9     Señalar el subproceso que ejecutó la última actividad según lo indicado en LISTAACTIVIDADES
10  END IF
11  Reorganizar los arcos de GBPEL para eliminar los ciclos
12  Crear una estructura de datos en forma de arreglos bidimensionales (MatrizACTIVIDADES), para contener las actividades
      del proceso.
13  Recorrer GBPEL por niveles y agregar cada nodo a una fila de MatrizACTIVIDADES según el nivel correspondiente,
      manteniendo la relación de los nodo padre-hijo.
      Por ejemplo los nodos del nivel 1, en la fila 1 de MatrizACTIVIDADES, nodos del nivel 2, en la fila 2 de MatrizACTIVIDADES...
14  A cada nodo de la matriz MatrizACTIVIDADES, asignarle una coordenadas (X, Y) según su nivel y posición(fila,columna),
      Teniendo en consideración el tamaño de la imagen creada IMG_BPEL.
15  FOR entero i ← 1 hasta i = numero de Filas de MatrizACTIVIDADES DO
16    FOR entero j ← 1 hasta j = numero de Columnas de MatrizACTIVIDADES DO
17      Obtener el nodo con la información de la actividad, que representa nodo contenido en la MatrizACTIVIDADES
        en la fila i columna j
18      Obtener el icono de la actividad correspondiente al nodo en notación BPMN e insertar un símbolo para las
        actividades BPEL
19      IF BanderaMonitoreo = TRUE THEN
20        IF La actividad obtenida en el nodo se encuentra en la LISTAACTIVIDADES THEN
21          Marcar el icono de la actividad como ejecutada.
22        END IF
23        Marcar el icono de la actividad con el color de correspondiente al subproceso que pertenece
24      END IF
25      Obtener posición (X, Y) asignada a nodo contenido en MatrizACTIVIDADES en la fila i columna j.
26      Pintar icono obtenido con nombre de actividad (si se requiere) en posición (X, Y) de IMG_BPEL.
27      Obtener la posición de los nodos hijos asociados al nodo contenido en MatrizACTIVIDADES
28      Crear las líneas de los arcos correspondientes entre nodo Padre e Hijo, evitando la superposición entre las líneas de
        los arcos.
29      Pintar las líneas en IMG_BPEL
30    END FOR
31  END FOR

```

Si se tiene como entrada al algoritmo 4 el proceso BPEL (BPELXML) de la figura D1, la lista de subprocesos (LISTASUBPROCESOS) con los siguientes datos LISTASUBPROCESOS = (Operacion, Sumadora, Sustractora), la lista de actores (LISTAACTORES) con los siguientes datos LISTAACTORES = (Actor1, Actor2,

Actor3) , la lista de actividades (LISTA_{ACTIVIDADES}) con la siguiente información LISTA_{ACTIVIDADES} = (`<receive name="EntradaDatos" partnerLink="Operadora" portType="as:Opera" operation="operar" variable="Entrada"/>`) y finalmente con la *BanderaMonitoreo* = TRUE, se ejecutaran los siguientes pasos del algoritmo.

Al ejecutarse la línea 2 se crea una imagen en blanco se crea una imagen en blanco donde se pintara el proceso BPEL, luego en la línea 3 se convierte el proceso BPEL a un modelo de grafos, por lo cual se obtiene un grafo como el presentado en la figura D2.

Ya que la variable de entrada *BanderaMonitoreo* tiene el valor de TRUE se cumple la condición planteada en la línea 4. En la línea 5 y 6 se asigna de manera aleatoria un color a cada Actor y subprocesso contenidos en la LISTA_{ACTORES} y LISTA_{SUBPROCESOS} respectivamente, para este caso suponemos que para la LISTA_{ACTORES} al *Actor1* se le asigno Azul, al *Actor2* se le asigno Amarillo y al *Actor3* se le asigno Verde, por lo cual el subprocesso *Operación* queda con color Azul, el subprocesso *Sumadora* queda con color Amarillo y el subprocesso *Sustractora* queda con Verde. Esta información es presentada en unas tablas (Líneas 7 y 8) las cuales se pintan sobre la imagen como se presentan en las figura D9 y D10.

SUBPROCESOS	
Operacion	●
Sumadora	●
Sustractora	●

Figura D9. Tabla Subprocesos

Actores	
Actor1	●
Actor2	●
Actor3	●

Figura D10. Tabla Actores

Al ejecutarse la línea 9 se consulta la LISTA_{ACTIVIDADES} y como la única actividad registrada en ella es la actividad (`<receive name="EntradaDatos"...`) la cual está asociada al subprocesso *Operación* se señala como se presenta en la figura D11.

SUBPROCESOS	
Operacion	●
Sumadora	●
Sustractora	●

Figura D11. Tabla subprocessos señalada

Luego debido a que el grafo no tiene ciclos no es necesario la ejecución de la línea 11, entonces a continuación se crea un matriz para contener las actividades del proceso línea 12 en este caso se crea una matriz de (6X2) y según lo establecido en la línea 13 utilizando el grafo de la figura D2 se tendría una matriz similar a la presentada en la figura D12.

Start (<code><process></code> ó <code><sequence main></code>)	
Receive (<code><receive name="EntradaDatos"...</code>)	
Start Switch (<code><switch name="DatoEntrada"></code>)	
Invoke (<code><invoke name="Suma"...</code>)	Invoke (<code><invoke name="Resta"...</code>)
End Switch (<code></switch></code>)	
End (<code></process></code> ó <code></sequence></code>)	

Figura D12. Matriz de niveles

Luego en la línea 14 se le asignan posiciones a los diferentes nodos contenidos en la **MatrizACTIVIDADES** según su nivel, para el caso de una imagen de 640X480 se podrían asignar los siguientes resultados:

- Fila 1 (una sola columna con información) → (256, 1)
- Fila 2 (una sola columna con información) → (256, 70)
- Fila 3 (una sola columna con información) → (256, 150)
- Fila 4 (dos columnas con información) → (128, 230), (128, 230)
- Fila 5 (una sola columna con información) → (256, 310)
- Fila 6 (una sola columna con información) → (256, 390)

En las siguientes Líneas (Línea 15 y 16) se realizan ciclos para recorrer cada uno de los campos de la matriz, en la primera iteración se obtiene lo asignado a la fila 1, columna 1 de la matriz, en este caso se obtiene la información del nodo *Start*, para este se obtiene la representación del nodo en notación BPMN (para este caso se utiliza la tabla 1) y para el símbolo la tabla 2. Como la *BanderaMonitoreo* es igual a *TRUE* y la actividad se encuentra en la **LISTAACTIVIDADES** se marca la actividad como ejecutada utilizando la imagen de la tabla 3.

Es decir primero se obtiene el icono que representa la actividad en BPMN, luego si es una actividad básica se coloca encima del icono obtenido el símbolo de la actividad BPEL y finalmente si es una actividad que ha sido ejecutada se le coloca encima también la imagen de actividad ejecutada, todo esto conforma el icono de la actividad.

Después de tener completo el icono de la actividad, se obtiene el color asignado al subproceso relacionado con esa actividad, y se pinta de ese color el símbolo BPEL que representa la actividad.

A continuación considerando las posiciones (X, Y) asignadas al nodo contenido en esa posición de la matriz (posición 1,1 de la matriz), se pinta el icono en esa posición sobre la imagen, para este caso sería la posición (256, 1) con lo que se tiene que la imagen queda centrada y en la posición más superior, esto sucede en la ejecución de las líneas 25 y 26. Los hijos de este nodo corresponden a los nodos ubicados en la filas debajo de la actual, para el caso de la primera iteración de las líneas 15 y 16 se tiene que el hijo de la actividad *Start* (<process>) es el nodo de la fila 2 *Receive* (<receive name="EntradaDatos"...>) cuya posición asignada es (256, 70) y como es un solo hijo no hay problemas de superposición de líneas por lo que se crea una línea directa, para finalmente ser pintada en la imagen.

Luego en la siguiente iteración se ubica en la posición (2,1) (ya que la posición (1, 2) es vacía), se obtienen el icono y el símbolo de la actividad *Receive* (<receive name="EntradaDatos"...>), en esta iteración se cumple la condición de la línea 19, pero no la de la línea 20 por lo cual únicamente se marca la actividad con el color amarillo (asignado previamente a este subproceso) debido a la ejecución de la línea 23, de aquí obtiene por la línea 25 la posición asignada para este caso sería (256, 70) para pinta el icono obtenido en la imagen en esa posición (líneas 26 y 27). Finalmente se obtiene la posición de los nodos hijos posición (3,1) de la matriz, para pintar la línea sobre la imagen (Líneas 28 y 29). Así continua para las siguientes iteraciones de los ciclos obteniendo la imagen presentada en la figura D13.

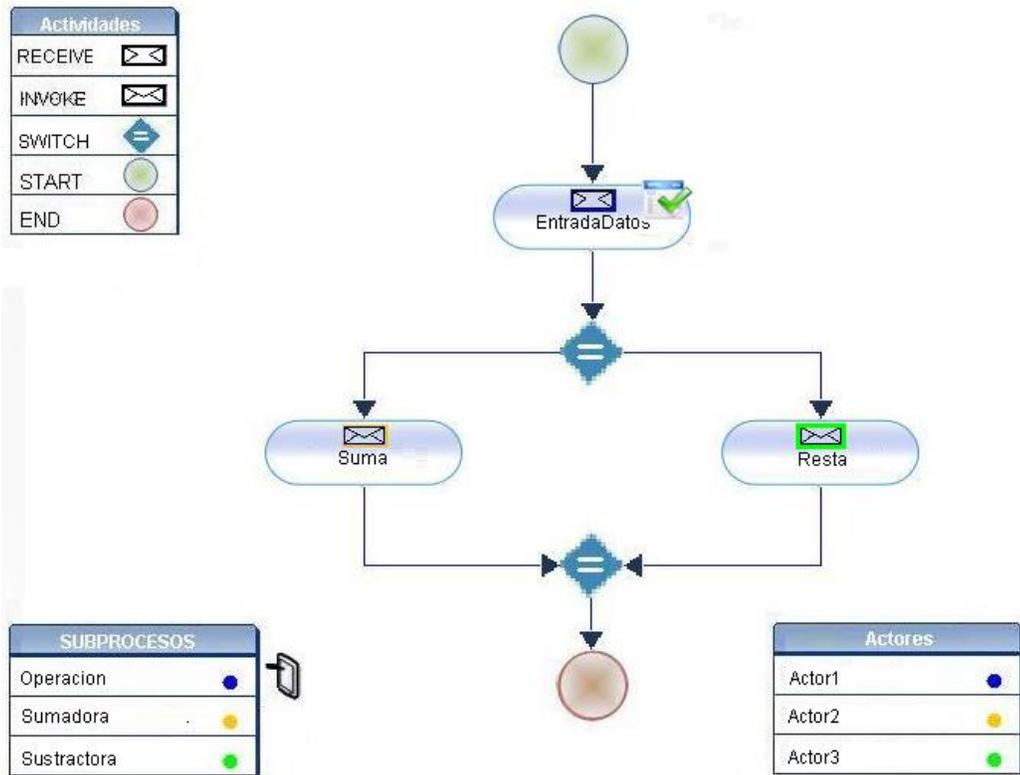


Figura D13. Imagen obtenida del proceso de monitoreo.

ACTIVIDADES REPRESENTADAS	ICONO REPRESENTACIÓN BPMN UTILIZADA
<ul style="list-style-type: none"> • <i><receive></i>, <i><invoke></i>, <i><reply></i> y <i><assing></i> 	
<ul style="list-style-type: none"> • <i><switch></i> 	
<ul style="list-style-type: none"> • <i><pick></i> 	
<ul style="list-style-type: none"> • <i><flow></i> 	
<ul style="list-style-type: none"> • <i><while></i>: <i>dado igualmente por los arcos que representan el bucle</i> 	
<ul style="list-style-type: none"> • <i>Start (<process>)</i> 	
<ul style="list-style-type: none"> • <i>End (</process>)</i> 	
<ul style="list-style-type: none"> • <i>Flujo del proceso</i> 	

Tabla D1. Representación BPMN de las actividades

ACTIVIDADES REPRESENTADAS	SÍMBOLO REPRESENTACIÓN BPEL UTILIZADA
<ul style="list-style-type: none"> • <i><receive></i> 	
<ul style="list-style-type: none"> • <i><invoke></i> 	
<ul style="list-style-type: none"> • <i><reply></i> 	
<ul style="list-style-type: none"> • <i><assing></i> 	

Tabla D2. Símbolos utilizados para representación BPEL de las actividades básicas

OTROS SÍMBOLOS	REPRESENTACIÓN GRÁFICA SÍMBOLO
<ul style="list-style-type: none"> • <i>Actividades ejecutadas</i> 	

Tabla D3. Otros símbolos utilizados para representación del flujo BPEL monitoreado

ANEXO E. TABLAS DE RESULTADOS DE LA PRUEBA DE RENDIMIENTO Y ESTABILIDAD

Tablas Resultados Pruebas Para el Particionamiento y Sincronización

Prueba de rendimiento PR1:

Numero de Actividades	Tiempo (ms)
10	718
50	792
150	1570
300	1960
500	2902
1000	5483
1500	8451
2000	10615

Tabla E1. Tabla de resultados prueba de rendimiento PR1

Tablas Resultados Pruebas Para la Distribución

Prueba de rendimiento DR1

- **NOKIA N93:** 190ms, 215ms, 175ms, 179ms, 201ms
- **Pocket PC DELL AXIM x51v:** 209ms, 180ms, 203ms, 149ms, 168ms
- **Emulador:** 102ms, 78ms, 89ms, 93ms, 97ms

GPRS Nokia N93: 3762ms, 1894ms, 2712ms, 2225ms, 2476ms

Prueba de rendimiento DR2

Tamaño Archivo (KB)	N93 (ms)	Pocket PC Dell (ms)	Emulador (ms)
1,1	377	489	290
1,6	549	712	422
2,2	755	979	581
2,8	960	1247	739
2,9	995	1291	766
3,1	1063	1380	819
3,5	1201	1559	924
3,8	1304	1692	1004
4,2	1441	1870	1109
4,3	1475	1915	1136
5	1715	2227	1321
6,5	2230	2895	1717

6,9	2367	3073	1823
7,3	2505	3251	1928
7,5	2573	3340	1981
7,96	2731	3545	2103
8,01	2748	3568	2116
8,52	2923	3795	2251
8,83	3030	3933	2333
9,01	3092	4013	2380
9,12	3129	4062	2409
9,54	3274	3192	1948
9,88	4338	4401	2123
10,01	4741	4665	2645
11,33	5204	5200	2931
11,59	5323	5319	2998
12,09	5553	5548	3127
12,63	5801	5796	3267
13,69	6287	6283,	3541
14,09	6471	6466	3644
14,59	6701	6696	3774
15,63	7179	7173	4043
16,25	7463	7458	4203
16,96	7789	7783	4387
17,56	8065	8059	4542
18,63	8556	8550	4819
19,23	8832	8825	4974
20,05	9002	9202	5186

Tabla E2. Tabla de resultados prueba de rendimiento DR2

Tamaño Archivo (KB)	WI-FI (ms)	GPRS (ms)
1,1	377	4386
1,1	377	4385
1,6	549	4527
2,2	755	4759
2,8	960	4961
2,9	995	4994
3,1	1063	5052
3,5	1201	5202
3,8	1304	5305
4,2	1441	5428
4,3	1475	5466

5	1715	5705
6,5	2230	6220
6,9	2367	6357
7,3	2505	6495
7,5	2573	6463
7,96	2731	6722
8,01	2748	6739
8,52	2923	6914
8,83	3030	6910
9,01	3092	6972
9,12	3129	7009
9,54	3274	7155
9,88	4338	8219
10,01	4741	8122
11,33	5204	8204
11,59	5323	8282
12,09	5553	8522
12,63	5801	8691
13,69	6287	9170
14,09	6471	9362
14,59	6701	9596
15,63	7179	10068
16,25	7463	10343
16,96	7789	10569
17,56	8065	11065
18,63	8556	11679
19,23	8832	11832
20,05	9002	12002

Tabla E3. Tabla de resultados prueba de rendimiento DR2 con el acceso desde dos tipos de redes

Prueba de estabilidad DE1

A continuación se presentan las tablas para esta prueba según el número de usuarios con acceso simultaneo

5 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)	BYTES
1	174	146
2	163	146
3	176	146
4	180	146
5	187	146

Tabla E4. Tiempos de respuesta a los usuarios cuando acceden 5 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	5	176	163	187	7.87	0%	5.0/ seg	0.72

Tabla E5. Resumen resultados acceso 5 usuarios.

10 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	461
2	1069
3	712
4	1001
5	939
6	801
7	1220
8	1469
9	727
10	677

Tabla E6 Tiempos de respuesta a los usuarios cuando acceden 10 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	10	907	461	1469	279.89	0%	6.3/seg	0.86

TablaE7. Resumen resultados acceso 10 usuarios.

20 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	533
2	746
3	653
4	751
5	1127
6	1330
7	1287
8	1351
9	1321
10	1347
11	1352
12	1320
13	1494
14	1273
15	1443
16	1293
17	1468
18	1369
19	1282
20	1271

Tabla E8. Tiempos de respuesta a los usuarios cuando acceden 20 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	20	1200	533	1494	278.37	0%	8.7/seg	1.17

Tabla E9. Resumen resultados acceso 20 usuarios.

30 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	504
2	633
3	817
4	845
5	1008
6	1160
7	1318
8	1464
9	1590
10	1734
11	1769
12	1871
13	1846
14	1861
15	1932
16	1929
17	1917
18	1921
19	1921
20	1884
21	1977
22	1993
23	2020
24	2003
25	2110
26	2052
27	2033
28	2028
29	2054
30	2039

Tabla E10. Tiempos de respuesta a los usuarios cuando acceden 30 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	30	1629	504	2110	493.55	0%	9.8/seg	1.41

Tabla E11. Resumen resultados acceso 30 usuarios.

40 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	2539
2	2568
3	2521
4	2470
5	2636
6	2680
7	2704
8	2427
9	2508
10	2680
11	2603
12	2454
13	2755
14	2768
15	2895
16	2890
17	2867
18	2891
19	2906
20	2937
21	2973
22	2910
23	2984
24	2889
25	2976
26	2978
27	3020
28	3052
29	3033
30	3034
31	3029
32	3045
33	3080
34	3068
35	2962
36	3040
37	3039
38	3149
39	3177
40	3253

Tabla E12. Tiempos de respuesta a los usuarios cuando acceden 40 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	40	2859	2427	3253	218.91	0%	9.7/ seg	0.35

Tabla E13. Resumen resultados acceso 40 usuarios.

50 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	725
2	1117
3	1120
4	1144
5	1317
6	1480
7	1752
8	1858
9	1924
10	2017
11	2423
12	2604
13	2757
14	3043
15	3078
16	3141
17	3170
18	3185
19	3181
20	3219
21	3227
22	3206
23	3318
24	3289
25	3327
26	3362
27	3357
28	3381
29	3416
30	3470
31	3456
32	3496
33	3495
34	3551
35	3601
36	3611
37	3694
38	3688
39	3744
40	3772
41	3762
42	3805
43	3895
44	3881
45	3936
46	3902
47	3931
48	4047

49	4030
50	4035

Tabla E14. Tiempos de respuesta a los usuarios cuando acceden 50 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	50	3058	725	4047	894.34	0%	9.9/ seg	1.37

Tabla E15. Resumen resultados acceso 50 usuarios.

60 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	1300
2	1303
3	1479
4	1567
5	1826
6	1819
7	2122
8	2297
9	2413
10	2623
11	2781
12	2927
13	3050
14	3236
15	3345
16	3457
17	3610
18	3682
19	3664
20	3720
21	3697
22	3739
23	3750
24	3765
25	3778
26	3782
27	3853
28	3884
29	3894
30	3937
31	3955
32	3954
33	3997
34	4043
35	4026
36	4048
37	4166
38	4179
39	4172

40	4238
41	4280
42	4308
43	4365
44	4389
45	4388
46	4418
47	4409
48	4485
49	4468
50	4450
51	4569
52	4646
53	4608
54	4636
55	4558
56	4601
57	4634
58	4647
59	4671
60	4635

Tabla E16. Tiempos de respuesta a los usuarios cuando acceden 60 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	60	3686	1300	4671	935.76	0%	10.5/ seg	1.50

Tabla E17. Resumen resultados acceso 60 usuarios.

70 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	1131
2	1421
3	1778
4	1789
5	2046
6	2246
7	2630
8	2701
9	3084
10	3322
11	3505
12	3992
13	4101
14	4088
15	4186
16	4180
17	4228
18	4230
19	4282
20	4353

21	4398
22	4430
23	4469
24	4503
25	4519
26	4542
27	4548
28	4619
29	4637
30	4637
31	4696
32	4685
33	4746
34	4810
35	4803
36	4859
37	4907
38	4909
39	4950
40	5008
41	5041
42	5048
43	5097
44	5123
45	5167
46	5239
47	5246
48	5282
49	5348
50	5343
51	5390
52	5445
53	5520
54	5577
55	5656
56	5692
57	5718
58	5736
59	5765
60	5796
61	5836
62	5892
63	5943
64	5976
65	6003
66	5995
67	6059
68	6073
69	6097
70	6108

Tabla E18. Tiempos de respuesta a los usuarios cuando acceden 70 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	70	4645	1131	6108	1189.39	0%	9.8/ seg	1.40

Tabla E19. Resumen resultados acceso 70 usuarios.

80 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	2192
2	2327
3	2383
4	2451
5	2978
6	3345
7	3474
8	3592
9	4013
10	4130
11	4695
12	4714
13	4738
14	4738
15	4786
16	4809
17	4819
18	4917
19	4955
20	5000
21	4999
22	5078
23	5099
24	5140
25	5193
26	5219
27	5249
28	5283
29	5293
30	5335
31	5401
32	5434
33	5445
34	5484
35	5532
36	5556
37	5605
38	5620
39	5670
40	5664

41	5726
42	5722
43	5812
44	5806
45	5921
46	5885
47	5959
48	5972
49	5928
50	6045
51	6003
52	6069
53	6104
54	6232
55	6239
56	6294
57	6316
58	6348
59	6393
60	6427
61	6425
62	6475
63	6473
64	6516
65	6566
66	6588
67	6621
68	6643
69	6695
70	6712
71	6742
72	6746
73	6802
74	6813
75	6879
76	6888
77	6885
78	6925
79	7032
80	6989

Tabla E20. Tiempos de respuesta a los usuarios cuando acceden 80 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	80	5493	2192	7032	1207.48	0%	10/ seg	1.34

Tabla E21. Resumen resultados acceso 80 usuarios.

90 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	5284
2	5251
3	5398
4	5407
5	5354
6	5379
7	5448
8	5395
9	5521
10	5904
11	5931
12	5916
13	5969
14	6030
15	6058
16	6150
17	6146
18	6147
19	6274
20	6320
21	6330
22	6370
23	6468
24	6450
25	6449
26	6530
27	6576
28	6631
29	6651
30	6697
31	6719
32	6765
33	6834
34	6845
35	6868
36	6907
37	6947
38	6990
39	7030
40	7071
41	7101
42	7159
43	7186
44	7220
45	7273
46	7275
47	7326
48	7373

49	7381
50	7444
51	7425
52	7541
53	7539
54	7561
55	7643
56	7638
57	7619
58	7706
59	7746
60	7741
61	7838
62	7861
63	7864
64	7920
65	7973
66	7975
67	8065
68	8067
69	8160
70	8173
71	8244
72	8241
73	8337
74	8332
75	8375
76	8432
77	8486
78	8490
79	8486
80	8561
81	8612
82	8552
83	8560
84	8679
85	8693
86	8764
87	8784
88	8830
89	8850
90	8870

Tabla E22. Tiempos de respuesta a los usuarios cuando acceden 90 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	90	7215	5251	8870	1031.38	0%	9/ seg	1.30

Tabla E23. Resumen resultados acceso 90 usuarios.

100 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	3252
2	3921
3	5883
4	6075
5	6208
6	6254
7	6210
8	6272
9	6347
10	6385
11	6380
12	6495
13	6484
14	6519
15	6577
16	6775
17	6633
18	6801
19	6760
20	6901
21	6827
22	6864
23	6913
24	6975
25	6984
26	7001
27	7054
28	7090
29	7129
30	7169
31	7207
32	7262
33	7298
34	7331
35	7375
36	7413
37	7489
38	7522
39	7553
40	7624
41	7638
42	7720
43	7766
44	7785
45	7850
46	7858
47	7918
48	7947

49	8000
50	8067
51	8161
52	8157
53	8205
54	8251
55	8293
56	8329
57	8392
58	8417
59	8478
60	8547
61	8572
62	8608
63	8603
64	8608
65	8603
66	8634
67	8681
68	8727
69	8818
70	8793
71	8857
72	8877
73	8881
74	8960
75	8942
76	9048
77	9036
78	9127
79	9123
80	9140
81	9188
82	9172
83	9302
84	9329
85	9445
86	9449
87	9481
88	9502
89	9515
90	9567
91	9577
92	9601
93	9661
94	9782
95	9834
96	9820
97	9906
98	9987
99	9972

100	10024
------------	--------------

Tabla E24. Tiempos de respuesta a los usuarios cuando acceden 100 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	100	8002	3232	9987	1315.60	0%	9/ seg	1.29

Tabla E25. Resumen resultados acceso 100 usuarios.

Tablas Resultados Pruebas Para la Sincronización

Prueba de rendimiento ER1

- **NOKIA N93:** 3002ms, 2713ms, 2682ms, 3092ms, 2625ms.
- **Pocket PC DELL AXIM x51v:** 2767ms, 2106ms, 3101ms, 2312ms, 2445ms.
- **Emulador:** 1813ms, 1403ms, 957ms, 1466ms, 1631ms.

GPRS Nokia N93: 5270ms, 3911ms, 4326ms, 4019ms, 3728ms.

Prueba de rendimiento ER2

- **NOKIA N93:** 5312ms, 4139ms, 3510ms, 4237ms, 4448ms.
- **Pocket PC DELL AXIM x51v:** 4139ms, 5111ms, 3798ms, 4481ms, 3470ms.
- **Emulador:** 2845ms, 1611ms, 1921ms, 2769ms, 2022ms.

GPRS Nokia N93: 8479ms, 7241ms, 7321ms, 8083ms, 7190ms.

Prueba de estabilidad EE1

5 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	460
2	387
3	336
4	366
5	253

Tabla E26. Tiempos de respuesta a los usuarios cuando acceden 5 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	5	360	253	460	67.51	0%	4.6/ seg	2.45

TablaE27. Resumen resultados acceso 5 usuarios.

10 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	378
2	592
3	544
4	555
5	590
6	797
7	1141
8	936
9	1157
10	1089

Tabla E28. Tiempos de respuesta a los usuarios cuando acceden 10 usuarios al tiempo para obtener su variables de sincronización.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	10	777	378	1157	270.53	0%	5.1/seg	2.71

Tabla E29. Resumen resultados acceso 10 usuarios.

20 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	811
2	812
3	965
4	1125
5	1348
6	1713
7	1832
8	1641
9	1786
10	1970
11	2073
12	2033
13	2191
14	2177
15	2038
16	1980
17	2122
18	1979
19	2047
20	1983

Tabla E30. Tiempos de respuesta a los usuarios cuando acceden 20 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	20	1731	811	2191	448.50	0%	6.5/seg	3.47

Tabla E31. Resumen resultados acceso 20 usuarios.

30 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	861
2	1397
3	2060
4	2285
5	2506
6	2748
7	2909
8	2786
9	2883
10	3076
11	3245
12	3306
13	3335
14	3381
15	3343
16	3417
17	3294
18	3448
19	3465
20	3308
21	3476
22	3417
23	3420
24	3445
25	3452
26	3734
27	3429
28	3431
29	3537
30	3597

Tabla E32. Tiempos de respuesta a los usuarios cuando acceden 30 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	30	3066	861	3734	648.64	0%	6.6/seg	3.52

Tabla E33. Resumen resultados acceso 30 usuarios.

40 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	2782
2	2779
3	2835
4	2861
5	2853
6	3445
7	4295
8	4300

9	4464
10	4281
11	4478
12	4583
13	4543
14	4511
15	4686
16	4659
17	4796
18	4828
19	4867
20	4906
21	4961
22	4941
23	4929
24	4979
25	4984
26	4972
27	5038
28	5024
29	5039
30	5058
31	5039
32	5055
33	5088
34	5075
35	4088
36	5079
37	5191
38	5196
39	5141
40	5200

Tabla E34. Tiempos de respuesta a los usuarios cuando acceden 40 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	40	4568	2779	5200	739.27	0%	6.5/ seg	3.44

Tabla E35. Resumen resultados acceso 40 usuarios.

50 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	990
2	1082
3	1327
4	2921
5	3568
6	3677
7	3721

8	3515
9	3795
10	4016
11	4479
12	5430
13	5534
14	5711
15	5629
16	5717
17	5818
18	5708
19	5834
20	5885
21	5709
22	6038
23	5994
24	6065
25	5955
26	5969
27	6100
28	6025
29	6514
30	6219
31	6183
32	6429
33	6327
34	6520
35	6067
36	6135
37	6152
38	6237
39	6317
40	6350
41	6353
42	6440
43	6460
44	6513
45	6546
46	6662
47	6724
48	6745
49	6783
50	6799

Tabla E36. Tiempos de respuesta a los usuarios cuando acceden 50 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	50	5473	990	6799	1461.14	0%	6.4/ seg	3.40

Tabla E37. Resumen resultados acceso 50 usuarios.

60 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	1268
2	1246
3	1406
4	1793
5	1808
6	2963
7	2896
8	2992
9	3105
10	3123
11	3100
12	4323
13	4349
14	4603
15	5730
16	5312
17	5365
18	6194
19	6249
20	6332
21	6398
22	6388
23	6403
24	6487
25	6553
26	6418
27	6843
28	6413
29	6369
30	6572
31	6807
32	6977
33	6960
34	7157
35	7288
36	7384
37	7464
38	7576
39	7697
40	7697
41	7827
42	8501
43	8739
44	8505
45	8339
46	8345
47	8313
48	8884
49	8244
50	8518

51	8630
52	8428
53	8317
54	8420
55	8753
56	8430
57	8631
58	8879
59	9214
60	8404

Tabla E38. Tiempos de respuesta a los usuarios cuando acceden 60 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	60	6371	1246	9241	2247.48	0%	6.4/ seg	3.39

Tabla E39. Resumen resultados acceso 60 usuarios.

70 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	1268
2	1246
3	1406
4	1793
5	1808
6	2963
7	2896
8	2992
9	3105
10	3123
11	3100
12	4323
13	4349
14	4603
15	5730
16	5312
17	5365
18	6194
19	6249
20	6332
21	6398
22	6388
23	6403
24	6487
25	6553
26	6418
27	6843

28	6413
29	6369
30	6572
31	6807
32	6977
33	6960
34	7157
35	7288
36	7384
37	7464
38	7576
39	7697
40	7697
41	7827
42	8501
43	8739
44	8505
45	8339
46	8345
47	8313
48	8884
49	8244
50	8518
51	8630
52	8428
53	8317
54	8420
55	8753
56	8430
57	8631
58	8879
59	9214
60	8404
61	9264
62	9352
63	9449
64	9591
65	9562
66	9684
67	9723
68	9774
69	9882
70	10024

Tabla E40. Tiempos de respuesta a los usuarios cuando acceden 70 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	70	8342	1268	10024	1945.20	0%	4.9/seg	2.62

Tabla E41. Resumen resultados acceso 60 usuarios

Prueba de estabilidad EE2

5 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	1326
2	1322
3	1165
4	1289
5	1123

Tabla E42. Tiempos de respuesta a los usuarios cuando acceden 5 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	5	1245	1123	1326	84.51	0%	2.6/seg	0.68

Tabla E43. Resumen resultados acceso 5 usuarios.

10 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	1707
2	2570
3	2700
4	2799
5	2616
6	2595
7	3123
8	2957
9	2869
10	3082

Tabla E44. Tiempos de respuesta a los usuarios cuando acceden 10 usuarios al tiempo para obtener su variables de sincronización.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	10	2701	1707	3123	380.82	0%	2.7/seg	0.70

Tabla E45. Resumen resultados acceso 10 usuarios.

20 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	4461
2	5139
3	5180
4	5402

5	5313
6	5634
7	6055
8	6573
9	6214
10	6532
11	6737
12	6687
13	6832
14	6997
15	7156
16	6661
17	6458
18	6507
19	6763
20	6560

Tabla E46. Tiempos de respuesta a los usuarios cuando acceden 20 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	20	6203	4661	7156	701.53	0%	2.7/seg	0.71

Tabla E47. Resumen resultados acceso 20 usuarios.

30 USUARIOS

USUARIO	TIEMPO DE RESPUESTA (MS)
1	4461
2	5139
3	5180
4	5402
5	5313
6	5634
7	6055
8	6573
9	6214
10	6532
11	6737
12	6687
13	6832
14	6997
15	7156
16	6661
17	6458
18	6507
19	6763
20	6560
21	8510
22	9026

23	9277
24	8540
25	9375
26	9594
27	9487
28	9327
29	9638
30	10014

Tabla E48. Tiempos de respuesta a los usuarios cuando acceden 30 usuarios al tiempo para obtener su proceso de negocio.

Petición HTTP	Cantidad Peticiones	Promedio (Media)	Min	Max	Desviación Estándar	%Error	Rendimiento	Kb/Seg
TOTAL	30	9524	4461	10014	1350.26	0%	2.7/seg	0.71

Tabla E49. Resumen resultados acceso 30 usuarios.

Prueba de rendimiento MR1:

Actividades Básicas	Tiempo (ms)
10	401
20	429
50	478
150	726
250	1613
500	8594
600	14355

Tabla E50. Tabla de resultados prueba de rendimiento PR1

ANEXO F. PRUEBAS CON EMULADOR DE SUN JAVA WIRELESS TOOLKIT

Al igual que las pruebas realizadas sobre los dispositivos móviles reales, presentadas en el capítulo 5, se realizaron las mismas pruebas utilizando el emulador de Sun Java (TM) Wireless Toolkit 2.5.1 for CLDC **¡Error! No se encuentra el origen de la referencia.** con el dispositivo DefaultColorPhone (Emulador de 240x320, color).

El emulador se ejecutó sobre la misma máquina donde corre el servidor Web de manera que la conexión HTTP entre el emulador y el servidor se realiza de manera transparente. A continuación se presentan las pruebas realizadas y los resultados obtenidos con la emulación, junto con las gráficas de estos resultados. Además se presentan las gráficas comparativas de los resultados de la emulación con los resultados de los dispositivos reales.

PLAN DE PRUEBAS	
<u>Distribución de los subprocesos</u>	<ul style="list-style-type: none"> • Prueba de rendimiento DR1: Determina el tiempo que tarda la plataforma en validar y dar respuesta al usuario, después de que esté realiza el proceso de inicio de sesión desde su sistema móvil de información. Esta prueba se ejecuta utilizando diferentes dispositivos con el fin de verificar la influencia que tienen las características físicas de cada dispositivo en los tiempos de respuesta. • Prueba de rendimiento DR2: Determina el tiempo que demora un usuario del sistema móvil de información en descargar un proceso de negocio, después de haber validado sus datos y aceptado la descarga. Esta prueba se ejecuta utilizando diferentes dispositivos con el fin de verificar la influencia que tienen las características físicas de cada dispositivo en los tiempos de respuesta.
<u>Ejecución sincronizada</u>	<ul style="list-style-type: none"> • Prueba de rendimiento ER1: Determina el tiempo que tarda la plataforma en obtener y enviar las variables de inicio de ejecución, requeridas cuando se ejecuta una actividad <i>receive</i> de sincronización desde el motor BPEL del dispositivo móvil. Para esta prueba se mide el tiempo desde que el usuario inicia la ejecución del subproceso hasta que el servidor retorna la información de las variables requeridas y son mostradas en la pantalla del dispositivo móvil. Se ejecuta utilizando diferentes dispositivos con el fin de verificar la influencia que tienen las características físicas de cada dispositivo en los tiempos de respuesta. • Prueba de rendimiento ER2: Determina el tiempo que demora la plataforma en obtener y enviar el resultado de la invocación de un servicio Web al sistema móvil de información, cuando esté requiere de dicha invocación para la ejecución de su proceso de negocio. Para esta prueba se mide el tiempo desde que un cliente móvil hace la petición de invocación de un servicio Web al servidor, hasta cuando el móvil obtiene el resultado de la invocación y lo procesa para

<u>Ejecución sincronizada</u>	<p>continuar con la ejecución.</p> <p>Esta prueba se ejecuta utilizando diferentes dispositivos con el fin de verificar la influencia que tienen las características físicas de cada dispositivo en los tiempos de respuesta.</p>
-------------------------------	---

Tabla F1. Plan de pruebas.

Especificaciones Técnicas del Servidor Central y de los Dispositivos móviles usados en las pruebas.

- **Servidor Central.**

Procesador	RAM	Disco Duro	Sistema Operativo
Genuine Intel(R) T2080 @ 1.73GHZ x2	1 GB	120 GB	Microsoft Windows XP

Tabla F2. Especificaciones técnicas del Servidor Central de MobFlow.

- **Dispositivos Móviles**

Dispositivo	Procesador	RAM	ROM	Tamaño de pantalla	Sistema Operativo
Pocket PC DELL AXIM x51v	Intel PXA270, 520 MHz	64MB	256MB	480 X 640 Pixeles.	Microsoft Windows Mobile 5.0
Nokia N93	Dual ARM 11 332 MHz	64MB	50 MB	128 X 160 Pixeles.	Symbian 9.1

Tabla F3. Especificaciones técnicas de los dispositivos usados para las pruebas.

- Emulador de Sun Java (TM) Wireless Toolkit 2.5.1 for CLDC **¡Error! No se encuentra el origen de la referencia.** con el dispositivo DefaultColorPhone (Emulador de 240x320, color).

Resultados pruebas para la Distribución
<ul style="list-style-type: none"> • Prueba de rendimiento DR1: En la figura F1 se presenta la gráfica de resultados de las pruebas de rendimiento para la validación y obtención de respuesta en el proceso de registro de usuario requerido en la distribución, utilizando el Emulador, el cual se ejecuta sobre la misma máquina donde corre el servidor Web de manera que la conexión HTTP se realiza de manera transparente. <p>La figura F2 presenta la comparativa de los resultados obtenidos con los dispositivos móviles reales y el emulador, donde se observa que el emulador presenta tiempos significativamente menores a los dispositivos móviles reales debido a la manera como se realiza la conexión, lo cual permite establecer que los tiempos de respuesta van ligados a la manera como se conecta el dispositivo con el servidor.</p>

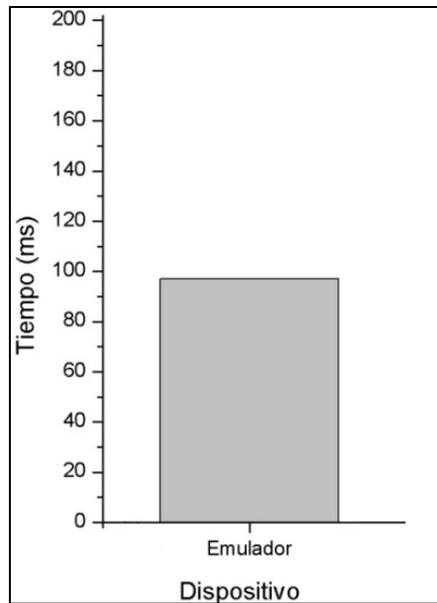


Figura F1. Rendimiento de la plataforma en el proceso de descarga y almacenamiento de procesos de negocio utilizando el emulador de Sun.

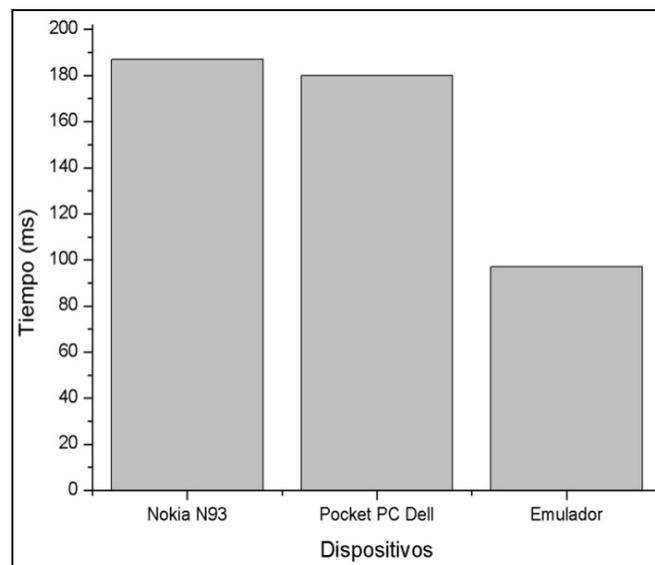


Figura F2. Rendimiento de la plataforma en el proceso de descarga y almacenamiento de procesos de negocio utilizando diferentes dispositivos.

- Prueba de rendimiento DR2:** En la figura F3 se muestra la gráfica de resultados de las pruebas de rendimiento para la descarga y almacenamiento de subprocesos de negocio (archivo BPEL y WSDL). Para esta prueba se generaron varios subprocesos con diferentes tamaños que varían desde 1KB hasta 20 KB, utilizando el Emulador, el cual se ejecuta sobre la misma máquina donde corre el servidor Web de manera que la conexión HTTP se realiza de manera transparente.

Como se visualiza en la figura F3, los diferentes tiempos de descarga varían según el tamaño del archivo, entre más grande es el archivo más demora la descarga.

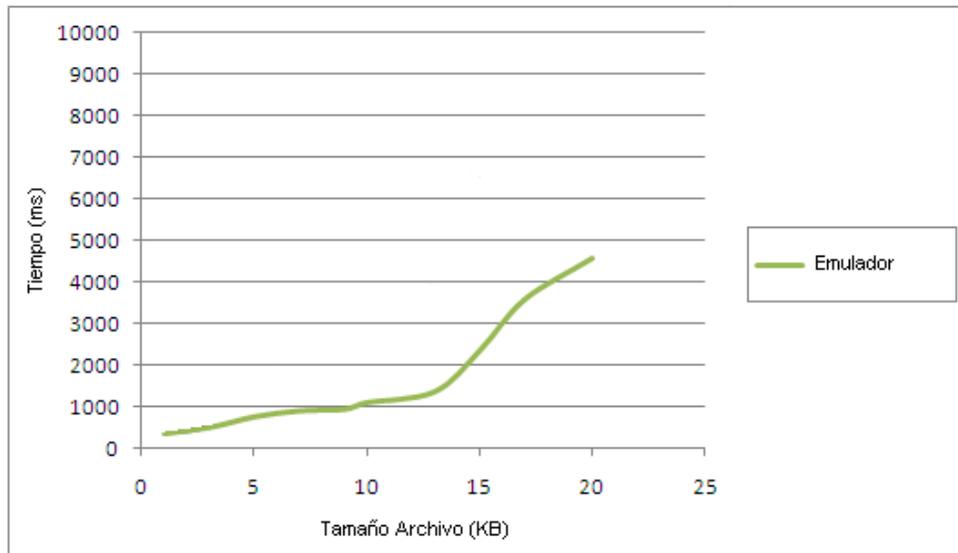


Figura F3. Rendimiento de la plataforma en el proceso de descarga con el acceso desde el emulador.

La figura F4 presenta los resultados comparativos de la prueba utilizando diferentes tipos de dispositivos, donde se observa que los tiempos con el emulador son bastante reducidos en comparación con los dispositivos reales, debido a la manera como se realiza la conexión con el servidor web.

Igualmente de esto se puede concluir que los tiempos del proceso de descarga de archivos a los sistemas móviles de información, están ligados al tamaño de los archivos y a la conexión, más que a las capacidades de los dispositivos móviles usados en la prueba.

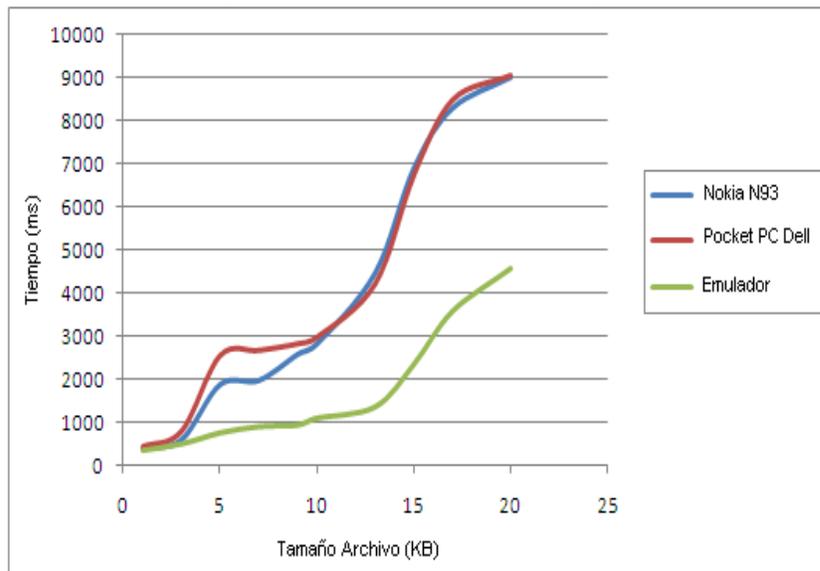


Figura F4. Rendimiento de la plataforma en el proceso de descarga con el acceso desde diferentes tipos de dispositivos

Tabla F4. Análisis de resultados pruebas para la funcionalidad de distribución

Resultados pruebas para la ejecución sincronizada

- **Prueba de rendimiento ER1:** En la figura F5 se visualiza la gráfica que presenta los resultados de las pruebas de rendimiento, donde se estipulan los tiempos de respuesta promedio que tarda la plataforma en obtener y presentar las variables de sincronización en el sistema móvil de información, proceso requerido para llevar a cabo una ejecución sincronizada del proceso. Los resultados presentados en la figura son los obtenidos con el emulador, el cual se conecta al Servidor MobFlow como se ha descrito anteriormente. En este se descargaron y ejecutaron los subprocesos de negocio que requerían variables de sincronización para iniciar su ejecución, luego se midió el tiempo de respuesta desde el momento de hacer la petición al servidor hasta obtener la información de las variables en el emulador.

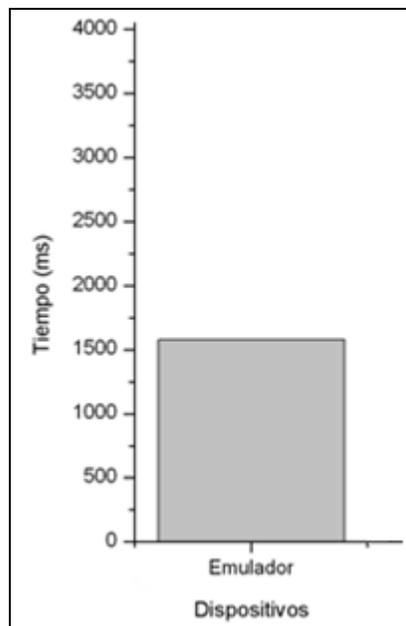


Figura F5. Rendimiento de la plataforma en el proceso de obtención de variables de sincronización con el acceso desde el emulador.

En la figura F6 se presenta la gráfica comparativa de rendimiento de la plataforma en el proceso de obtención de variables de sincronización con el acceso desde los tres dispositivos, se observan tiempos de respuesta aceptables al obtener las variables de sincronización (figura F6), las variaciones de tiempo entre el Nokia N93 y la Pocket PC Dell son mínimas, a diferencia de los tiempos obtenidos con el emulador, el cual presenta tiempos significativamente menores a los dispositivos móviles reales debido a la manera como se realiza la conexión. Por otra parte es importante tener en consideración que la información de las variables son pequeños documentos XML los cuales no superan el tamaño de 1KB generalmente, por lo que podemos concluir que este proceso llevado a cabo por la plataforma es óptimo para gran parte de los procesos de negocio que pueden ser ejecutados.

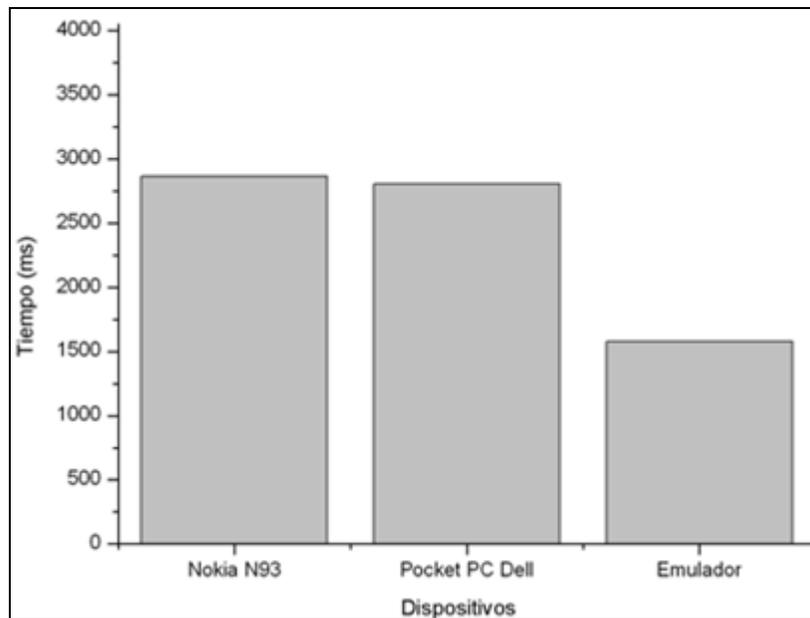


Figura F6. Gráfica de rendimiento del proceso de obtención de variables de sincronización.

- Prueba de rendimiento ER2:** El proceso de invocación de un servicio Web, cuyo resultado es requerido para la ejecución de un subproceso de negocio desde el sistema móvil de información, posee una mayor complejidad debido a la cantidad de funciones involucradas. La figura F7 presenta los resultados de las pruebas de rendimiento que determinan el tiempo que demora la plataforma en obtener y enviar el resultado de la invocación de un servicio Web al sistema móvil de información, utilizando como dispositivo de referencia el emulador.

Para esta prueba se descargaron y ejecutaron subprocesos de negocio que requerían invocación de servicios Web, en los cuales se midió el tiempo desde la petición para la invocación de los servicios hasta obtener su respuesta.

En la figura F8 se presenta la gráfica comparativa de los resultados utilizando los dos dispositivos móviles reales y el emulador, la evaluación de estos resultados muestra un rendimiento aceptable de la plataforma sobre los dispositivos usados para la prueba. Igual a las pruebas anteriores se observan tiempos de respuesta bastante inferiores con el emulador en comparación con los dispositivos móviles reales, esto se da debido a la manera como realiza la conexión el emulador con la plataforma.

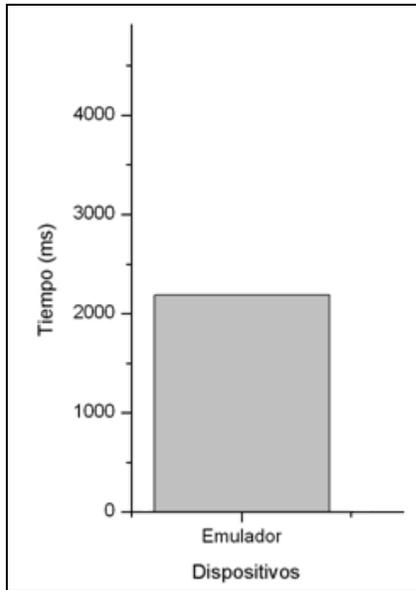


Figura F7. Gráfica de rendimiento del proceso de obtención de respuesta en la invocación de un servicio Web requerido para la ejecución de un subproceso.

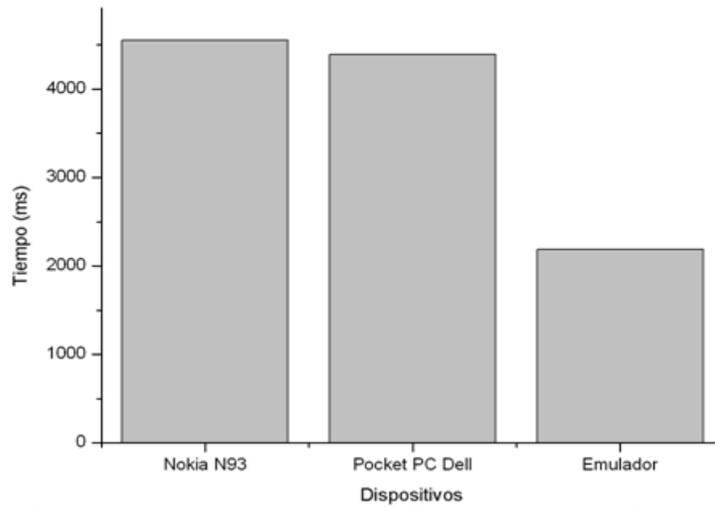


Figura F8. Gráfica de rendimiento del proceso de obtención de respuesta en la invocación de un servicio Web requerido para la ejecución de un subproceso.

Tabla F5. Análisis de resultados pruebas para la funcionalidad de ejecución distribuida