

**ANEXO B MODELADO DE LA APLICACIÓN  
BT\_FILETRANSFER**

## TABLA DE CONTENIDO

<b>ANEXO B MODELADO DE LA APLICACIÓN BT_FILETRANSFER</b>	<b>2</b>
<b>B.1 Análisis del dominio</b>	<b>3</b>
B.1.1 Descripción de Problema	3
B.1.2 Propósito del Sistema a Desarrollar	3
B.1.3 Modelo de Casos de Uso	4
B.1.4 Arbol de Funciones	5
B.1.5 Análisis de Riesgos y Plan del Software	5
B.1.5.1 Identificación de Riesgos y Estrategias de Solución	5
B.1.5.2 Recursos	6
<b>B.2 Análisis del software BT_File Transfer</b>	<b>8</b>
B.2.1 Casos de Uso Elementales Extendidos	8
B.2.1.1 Establecimiento de la conexión	8
B.2.1.2 Transferir de Archivos	15
B.2.2 Paquetes de Análisis	19
B.2.2.1 Paquete Interfaz de Aplicación BT_FileTransfer	20
B.2.2.2 Paquete Control de Aplicación BT_FileTransfe	23
B.2.2.3 Paquete Entidad de Aplicación BT_FileTransfer	33
B.2.3 Diagrama de Clases de Análisis	36
B.2.3.1 Establecimiento de la Conexión	36
B.2.3.2 Transferir de Archivos	37
B.2.4 Diagramas De Interacción Del Sistema	38
B.2.4.1 Establecimiento de la Conexión	38
B.2.4.2 Transferir Archivo	54

## ***B.1 ANALISIS DEL DOMINIO***

### **B.1.1 Descripción de Problema**

Un usuario de un equipo portátil desea copiar un archivo que se encuentra en su equipo; ambos tienen la posibilidad de comunicarse por Bluetooth.

También se pretende que en la aplicación involucrada en la transferencia se puedan apreciar los procedimientos de conexión relacionados con Bluetooth.

### **B.1.2 Propósito del Sistema a Desarrollar**

Una aplicación que le permita al usuario escoger un archivo local y transferirlo hacia una ubicación especificada en el otro dispositivo.

El usuario activa la aplicación, la cual muestra los dispositivos disponibles para transferirles archivos.

El usuario selecciona cuál de los dispositivos va a recibir el archivo. Luego se muestra al usuario la estructura de archivos y directorios local con el fin de seleccionar el archivo que se va a enviar.

En el dispositivo que recibe el archivo, puede seleccionarse la ruta en la cual el archivo que llega se va a almacenar, de no escogerse esta ruta, se almacenará en una ruta por defecto.

Cuando el usuario selecciona el archivo, presiona un botón para iniciar la transferencia. Cuando se completa la transferencia, se informa de esto al usuario en ambos equipos.

Cuando se completa la operación de copiado, se notifica en ambos equipos, con lo cual concluye la operación.

### B.1.3 Modelo de Casos de Uso

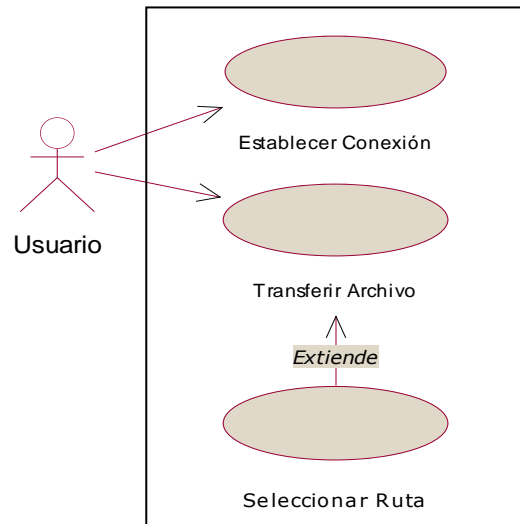


Figura B.1 Diagrama de Casos de Uso de la Aplicación

#### **Caso de Uso:** Establecer Conexión

##### **Descripción:**

Este caso de uso tiene por objetivo establecer la conexión física entre dos dispositivos, de manera que estos puedan reconocerse y hacer uso de los servicios que cada dispositivo ofrece. Con el fin de que un cliente pueda conectarse al servicio que ofrece un servidor, es preciso que este último registre sus servicios en una base de datos local. Así, cuando el cliente desea usar servicios, primero debe buscarlos en las bases de datos de los dispositivos con los que logra establecer comunicación, registrarlos en una base de datos local y posteriormente conectarse hacia ellos. Una vez se acepta la solicitud de conexión, puede empezar a usarse el servicio.

#### **Caso de Uso:** Transferir Archivos

##### **Descripción:**

El usuario, teniendo ya una conexión establecida con el dispositivo destino, puede realizar la transferencia de archivos. Con oprimir un botón de búsqueda el usuario elige el archivo de su propio sistema de archivos, cuando ha elegido este, pulsa el botón de enviar para dar

inicio a la transferencia. Durante la transferencia del archivo el usuario ve el progreso de la transferencia al mirar el tamaño total del archivo y el total transferido hasta el momento. Al terminar la transferencia, se le notifica que esta se ha realizado con éxito y puede volver a enviarse otro archivo. En el lado del dispositivo que recibe el archivo, se puede escoger la ruta en la que se escribirá el archivo por medio del caso de uso de Seleccionar Ruta.

#### **B.1.4 Arbol de Funciones**

1. Funciones del usuario
  - 1.1. Buscar Dispositivos
  - 1.2. Seleccionar dispositivo
    - 1.2.1. Refrescar lista de dispositivos
  - 1.3. Buscar Servicios
    - 1.3.1. Seleccionar Servicio
    - 1.3.2. Conectar a Servicio
    - 1.3.3. Refrescar Lista de Servicios
  - 1.4. Seleccionar archivo
  - 1.5. Enviar archivo
  - 1.6. Seleccionar Ruta

#### **B.1.5 Análisis de Riesgos y Plan del Software**

##### **B.1.5.1 Identificación de Riesgos y Estrategias de Solución**

<b>Riesgo</b>	<b>Estrategia</b>
La no existencia de aplicaciones y material de referencia, hace que el proceso de desarrollo se haga difícil.	Establecimiento de contactos directamente con las empresas desarrolladoras con el fin de obtener mayor información, inscripción a foros.
Las herramientas de desarrollo aun no da soporte completo al la Pila de Protocolos, lo cual hace que en el desarrollo tenga que implementarse los protocolos que hagan falta, agregando tiempo al desarrollo.	Adquirir, dentro de lo posible, herramientas cada vez más actuales.
La ausencia de recurso humano	Contactos con empresas a través de

capacitado en ésta temática.	Internet y otros grupos de desarrollo afines en Universidades y entidades de Educación.
------------------------------	---

Teniendo en cuenta el análisis de riesgos, se considera que se está en capacidad de superarlos gracias a las estrategias planteadas y que de acuerdo a esto, sólo será necesaria una iteración para el desarrollo de la aplicación, y todos los casos de uso se implementarán en su totalidad.

### **B.1.5.2 Recursos**

Los recursos a utilizar en el desarrollo son los siguientes:

#### *B.1.5.2.1 Recursos Humanos.*

Un ingeniero en Electrónica y Telecomunicaciones, director del proyecto, con experiencia en el campo de los sistemas telemáticos. Estará a cargo de coordinar el grupo de trabajo con disponibilidad de dos horas semanales durante el tiempo que dure el desarrollo del proyecto.

Dos estudiantes de ingeniería Electrónica y Telecomunicaciones, encargados de desarrollar el proyecto como trabajo de grado, con una disponibilidad de 30 horas semanales.

#### *B.1.5.2.2 Recursos Técnicos.*

##### *B.1.5.2.2.1 Recursos Hardware.*

- 2 computadores personales.
  - 866 MHz
  - 128 Megas en RAM
  - Disco Duro de 10 GB
- 1 impresora HP Deskjet
- Kit de desarrollo Bluetooth Ericsson.

##### *B.1.5.2.2.2 Recursos Software*

- Sistema operativo Windows98.
- Aplicaciones Microsoft Office97:
- Software de libre distribución
- Visual C++

#### *B.1.5.2.2.3 Recursos Bibliográficos.*

- Revistas relacionadas con el tema
- Libros y manuales de documentación del kit de desarrollo.
- Información obtenida a través de Internet
- Monografías: Trabajos de grado que contengan algún tipo de información que pueda servir para la elaboración de nuestro desarrollo.
- Informes y recomendaciones UIT y del Bluetooth SIG.

#### *B.1.5.2.3 Recursos en Comunicaciones*

- Servicio de acceso a Internet al interior de la universidad para realizar investigación y obtención de documentación y desarrollo de algunas aplicaciones.
- Acceso a la LAN de la universidad.
- Servicio de e-mail, FTP, etc.

#### *B.1.5.2.4 Recursos Físicos*

Para el desarrollo del proyecto, es necesario disponer de una infraestructura física dentro de la Facultad dos PCs con acceso a Internet con el fin de obtener información sobre los conceptos a incluir en la parte teórica y ayudas para el desarrollo de las aplicaciones software y hardware. Además teniendo en cuenta que se plantea el desarrollo de un prototipo hardware, es conveniente que se realice en un solo lugar para evitar averías y desajustes en su implementación.

#### *B.1.5.2.5 Recursos Varios*

- 1 caja de discos floppy de 3 ½, alta densidad.
- 2 cartuchos de tinta para impresora HP 690C, uno blanco y negro y otro de color.
- 4 resmas de papel Bond.
- Papel continuo de 3 copias
- Elementos de escritorio.

## B.2 ANÁLISIS DEL SOFTWARE BT\_FILE TRANSFER

### B.2.1 Casos de Uso Elementales Extendidos

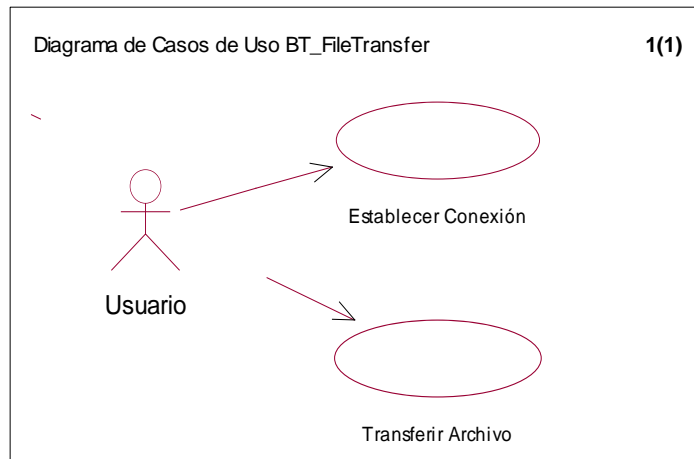


Figura B.2 Casos De Uso Elementales Extendidos

#### B.2.1.1 Establecimiento de la conexión

**Iniciador:** Usuario

**Propósito:** Establecer la concesión entre dos dispositivos. Registrar servicios para que otros dispositivos puedan encontrarlos y registrarlos para si mismos y acceder a ellos.

**Resumen:** El usuario inicia la aplicación (se establecen parámetros especiales para llevar a acabo la conexión de dispositivos) y procede a ejecutar una búsqueda de dispositivos en un entorno próximo. En caso de que se encuentre uno o mas dispositivos, la aplicación indicará al usuario cual es el nombre y la dirección del dispositivo que se descubrió.

El usuario selecciona uno de los dispositivos y presiona un botón para iniciar la búsqueda de servicios en el dispositivo remoto. Si el dispositivo remoto ha registrado servicios con anterioridad, estos podrán descubrirse y el usuario del dispositivo local sabrá a que tipo de servicios tiene acceso y si desea emplearlos, podrá registrarlos localmente para poder



conectarse con ellos. Para conectarse, el usuario presiona un botón que ejecuta una solicitud de conexión al servicio; si la solicitud se acepta, el dispositivo remoto responde afirmativamente y el servicio está listo para usarse. Cuando el usuario ya no desea continuar utilizando el servicio, presiona un botón de desconexión que permite desregistrar el servicio en la base de datos local y activar nuevamente la aplicación para conectarse con otro dispositivo u otro servicio.

**Tipo:** Primario

**Precondiciones:** Ninguna

**Referencias Cruzadas:**

**Funciones:** 1.1, 1.2, 1.2.1, 1.3, 1.3.1, 1.3.2, 1.3.3.

**Casos de Uso:** Ninguno

**Escenarios:**

**Escenario Uno**

*Flujo 1:* Escenario Búsqueda de Dispositivo y Servicios.

<b>Acción de Usuario</b>	<b>Respuesta del Sistema</b>
Ejecuta la aplicación	Despliega ventana de dialogo principal, con las opciones para seleccionar la interfaz del dispositivo, activas (USB, Serial)
Presiona una de las opciones de interfaz (USB, Serial)del modulo Bluetooth	Establece parámetros de configuración para conexión de una unidad Bluetooth e inicializa componentes del API: Configuración del puerto, Modo de Encriptación y Autenticación, Tiempo máximo de establecimiento de la conexión, Tiempo máximo de respuesta a paginación, Clase de Dispositivo, Nombre del Dispositivo, Modo de Escaneo. Habilita el botón para búsqueda de dispositivos. Habilita el botón de Registro de Servicio (Necesario para el servidor)
Presiona el botón para	Ejecuta la búsqueda de dispositivos en el entorno y

búsqueda de dispositivos.	permanece en estado de espera de respuestas. En caso de que un dispositivo remoto conteste la solicitud, se ejecuta una solicitud para conocer el nombre del dispositivo remoto; si este contesta, se actualiza la lista de dispositivos con el nombre del dispositivo.
Selecciona un dispositivo presente en la lista de dispositivos	Ejecuta una solicitud para establecer una conexión de datos con el dispositivo remoto. En caso de que la solicitud se acepte, el sistema presenta al usuario el nombre y la dirección del dispositivo remoto con el cual se ha establecido la conexión. De igual forma se almacenan datos propios de la conexión y se habilita el botón de Búsqueda de Servicios.
Presiona el botón de Búsqueda de Servicios	Se establece una conexión para búsqueda de servicios y se pregunta al dispositivo remoto si tiene disponibles servicios que cumplan con un patrón de búsqueda definido para cada servicio (En este caso se refiere al un servicio que ofrezca la posibilidad de realizar una Transferencia de Archivos). En caso de que el dispositivo remoto cuente con servicios que coinciden con el patrón de búsqueda, responderá devolviendo los identificadores de estos servicios que se han almacenado con anterioridad en su base de datos. Para cada uno de los identificadores de servicio devueltos por el dispositivo remoto, se indaga también acerca de sus características de servicio; estas características corresponden al Nombre del servicio, los protocolos que emplea para poder ejecutarse y los atributos especiales de estos protocolos de acuerdo al servicio en cuestión. Una vez se reciben estos datos del servicio, se almacena esta información, se muestra al usuario el nombre del servicio por el que se interrogó y se habilita el botón de seleccionar servicios.
Selecciona el servicio (Transferencia de Archivos) con el que se desea	Se cierra la conexión para búsqueda de servicios y sus atributos y se procede a almacenar el servicio seleccionado en el dispositivo local. Para tal efecto se

conectar y presiona el botón de Seleccionar Servicios	establece una conexión para almacenamiento de servicios con la base de datos local. Al momento de establecer la conexión, se genera en la base de datos local, una identificación única para el nuevo servicio a almacenar. Con este identificador se proceden a almacenar los demás atributos del servicio que se recolectaron a través de la indagación anterior. Una vez se confirma que el almacenamiento ha tenido éxito, se habilita un botón de conexión al servicio.
Presiona el botón de conexión al servicio.	Ejecuta una solicitud para conectarse al servicio deseado, indicando los parámetros que se necesitan para establecer esta conexión (Dependientes del servicio y de los protocolos que los soportan). Si se recibe una respuesta positiva a la solicitud, el dispositivo remoto se encuentra listo para iniciar con la prestación del servicio de transferencia de Archivos y el cliente puede iniciar la transacción. Se habilita el botón Búsqueda de Archivos
Presiona el botón de Desconexión	Se cierra la conexión con el servicio y así mismo se remueve el registro correspondiente en la base de datos. Se habilitan los botones para realizar nuevamente búsqueda de dispositivos o búsquedas de servicios.
Presiona el Botón de Cerrar	Se cierra la conexión con el servicio y así mismo se remueve el registro correspondiente en la base de datos, si no se ha realizado esta operación anteriormente. Finalmente se destruye el dialogo de aplicación.

### ***Diagrama de Flujo***

Flujo Principa

*Ver página siguiente...*

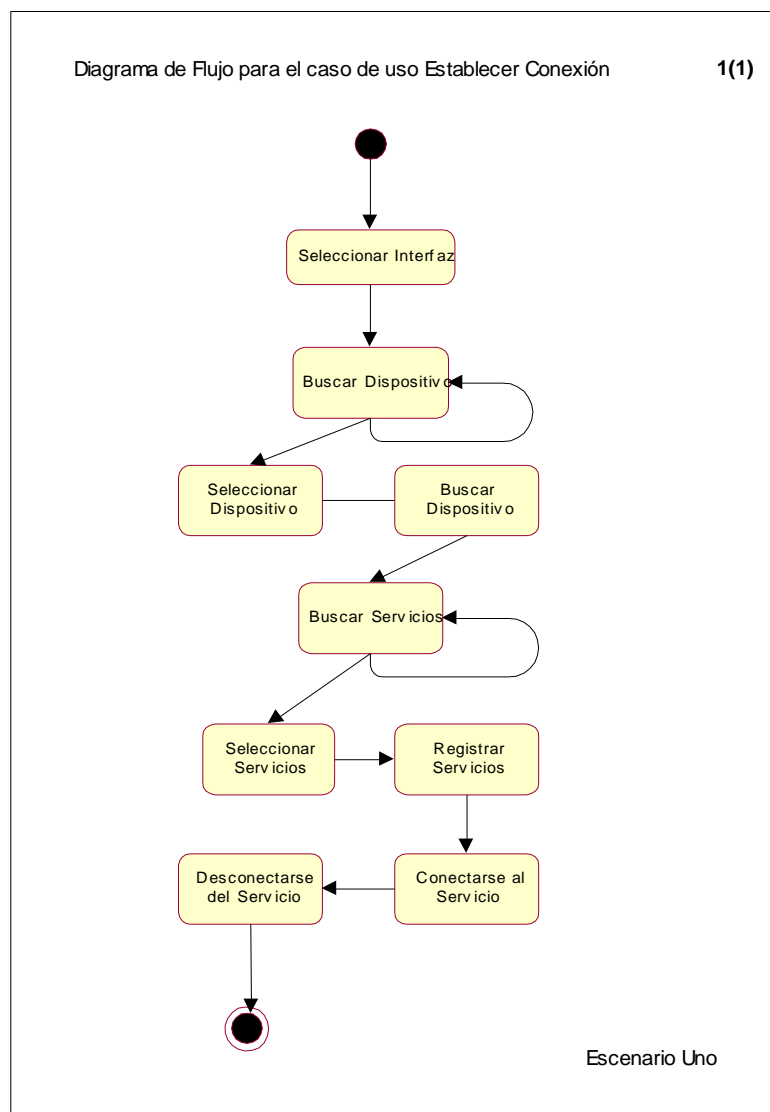


Figura B.3 Diagrama de Flujo para el caso de uso Establecimiento de la Conexión Escenario 1

**Escenario Dos**

Flujo 1: Registro de Servicios en el Servidor

Acción de Usuario	Respuesta del Sistema
-------------------	-----------------------

Ejecuta la aplicación	Despliega ventana de dialogo principal, con las opciones para seleccionar la interfaz del dispositivo, activas (USB, Serial)
Presiona una de las opciones de interfaz (USB, Serial)del modulo Bluetooth	Establece parámetros de configuración para conexión de una unidad Bluetooth e inicializa componentes del API: Configuración del puerto, Modo de Encriptación y Autenticación, Tiempo máximo de establecimiento de la conexión, Tiempo máximo de respuesta a paginación, Clase de Dispositivo, Nombre del Dispositivo, Modo de Escaneo. Habilita el botón para búsqueda de dispositivos. Habilita el botón de Registro de Servicio (Necesario para el servidor).
Presiona el Botón de Registro del Servicio.	Se establece una conexión para almacenamiento de servicios en el dispositivo local, con el fin de que otros dispositivos puedan encontrar los servicios que se está en capacidad de suministrar. Para servicio que se registra (En este caso un servicio para transferencia de archivos) deben registrarse sus atributos específicos; aquellos por los cuales será interrogado en caso de que otro dispositivo desee acceder a este servicio. Estos Atributos son: Nombre del servicio, los protocolos que emplea para poder ejecutarse y los atributos especiales de estos protocolos de acuerdo con el servicio en cuestión. Si la operación de almacenamiento en la base de datos es exitosa, se habilita el botón de para desregistrar el servicio (Desactivar Servicio), se deshabilita el botón de registro del servicio (Activar Servicio) y los botones para hacer búsqueda de dispositivos; esto con el fin de que el servidor mantenga este papel en el proceso de transferencia de archivos.
Presiona el Botón de Desactivar el servicio	Se registra el servicio de la base de datos (para descubrimiento de servicios). Si la operación de remoción en la base de datos es exitosa, se deshabilita el botón de para Registrar el servicio (Activar Servicio), se habilita el botón de desregistro del servicio (Desactivar Servicio) y

	los botones para hacer búsqueda de dispositivos; esto con el fin de que el servidor pueda tomar el papel de cliente en el proceso de transferencia de archivos.
Presiona el Botón de Cerrar	Se cierra la conexión con el servicio y así mismo se remueve el registro correspondiente en la base de datos, si no se ha realizado esta operación anteriormente. Finalmente se destruye el dialogo de aplicación.

### Diagrama de Flujo

#### Flujo Principal

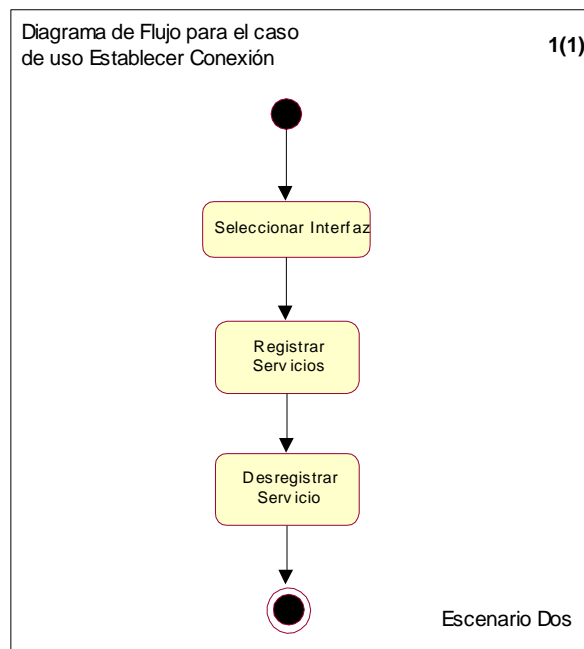


Figura B.4 Diagrama de Flujo para el caso de uso Establecimiento de la Conexión Escenario 2

### Excepciones

Perdida del enlace. En este caso la aplicación pasara al estado de búsqueda de dispositivos y deshabilitará los botones relacionados con la operaciones de prestación de servicio.

### Restricciones:

Ninguna

**Interfaces:**

**B.2.1.2 Transferir de Archivos**

**Iniciador:** Usuario

**Propósito:** Enviar archivos desde el dispositivo local hacia el dispositivo con el cual se ha establecido una conexión (con el caso de uso Establecer Conexión).

**Resumen:** El usuario busca el archivo dentro de su propio sistema de archivos, para ello oprime el botón de “Examinar” y con ello aparece un cuadro de dialogo donde explora la estructura de directorios local en busca del archivo deseado para enviar. Al seleccionar el archivo deseado, el usuario hace click en el botón de “abrir” y con ello se prepara el archivo para el envío. Si selecciona “Cancelar” no se selecciona ningún archivo.

Cuando el archivo esta preparado para el envío, el usuario puede ver información diversa del dispositivo como su nombre y tamaño; luego al pulsar el botón “Enviar” el archivo comienza a enviarse. En la ventana principal aparece una zona de despliegue de texto donde se muestra al usuario en tiempo real el estado de la transferencia, diciendo la cantidad de información del archivo enviada y la cantidad de información total del archivo. En el dispositivo que recibe el archivo también se despliega esta misma información.

El archivo debe guardarse en la ruta especificada por el usuario previamente mediante el caso de uso de Establecer Ruta.

Una vez terminada la transferencia de manera exitosa, se notifica esto por medio de cajas de texto en ambos dispositivos; el usuario de los dispositivos hace click en aceptar para confirmar la transferencia.

**Tipo:** Primario

**Precondiciones:** Se debe haber establecido una conexión con el dispositivo que recibirá el archivo, esta conexión es una conexión al servicio específico del otro dispositivo (establecida con el caso de uso “Establecer Conexión”). El servicio al cual se está conectado es el de Transferencia de Archivos.

**Referencias cruzadas:**

**Funciones:** 1.4 - Seleccionar Archivo y 1.5 - Enviar Archivo.

**Casos de uso:** Establecer Conexión.

**Escenarios:**

**Escenario Uno**

*Flujo 1: Envío del Archivo*

<b>Acción de Usuario</b>	<b>Respuesta del Sistema</b>
El usuario oprime el botón de “Examinar” de la ventana principal.	Aparece el cuadro de dialogo de selección de archivos con la estructura de directorios local, esta ventana no permite utilizar la ventana principal mientras esté activa.
Selecciona un archivo para enviar y presiona “Aceptar” (ver flujo 4)	Se almacena la ruta del archivo seleccionado para poder leerlo. El Cuadro de diálogo de selección de archivos desaparece y el nombre del archivo queda desplegado en la ventana principal. Se habilita el Botón de “Enviar”.

**Flujo 2: (Normal)**

<b>Acción de Usuario</b>	<b>Respuesta del Sistema</b>
El usuario Oprime el Botón de “Enviar” (ver flujo 5)	Se deshabilita el botón de “Examinar” y se deshabilita el botón de “Enviar”. El sistema envía la información del archivo relevante (como nombre y tamaño) en la primera trama de intercambio de objetos; luego secuencialmente envía tramas con contenido leído del archivo, recibiendo confirmaciones de llegada cada vez. Respectivamente en el dispositivo que recibe el



	archivo se va escribiendo el archivo a trozos hasta que llega el último, momento en el cual se cierra el archivo y se guarda. En la ventana principal aparece la información enviada y la información total que debe ser transferida. Al terminar la transferencia se muestra una caja de mensaje que indica la transferencia exitosa del archivo.
El usuario hace Click en “aceptar” de la caja de mensaje.	Se borra la información de transferencia de la ventana principal. Queda listo para una nueva transferencia. En el dispositivo que recibe el archivo (servidor) se realiza lo mismo. El caso de uso termina

### Flujo 3: Selección de la ruta de llegada del archivo

<b>Acción de Usuario</b>	<b>Respuesta del Sistema</b>
El usuario oprime el botón “Ruta” de la ventana principal	Presenta un cuadro de diálogo con la estructura de directorios del dispositivo local con el fin de localizar el directorio destino de los archivos que lleguen

### Flujo 4:

<b>Acción de Usuario</b>	<b>Respuesta del Sistema</b>
El usuario oprime el botón “Cancelar” al elegir el archivo.	El sistema no selecciona ningún archivo, deja deshabilitado el botón de “Enviar” y habilitado el botón de “Examinar” para realizar de nuevo la búsqueda.

### Flujo 5:

<b>Acción de Usuario</b>	<b>Respuesta del Sistema</b>
El usuario oprime el botón “Examinar” de nuevo después de elegir un archivo.	El sistema muestra el cuadro de diálogo de selección del archivo para elegir de nuevo.

### Flujo 6:

<b>Acción de Usuario</b>	<b>Respuesta del Sistema</b>
El usuario oprime el botón "Examinar" de nuevo después de haber enviado algún otro archivo.	El sistema muestra de nuevo el cuadro de diálogo de selección del archivo para realizar el próximo envío.

***Diagrama de Flujo***

Flujo principal

*Ver página siguiente...*

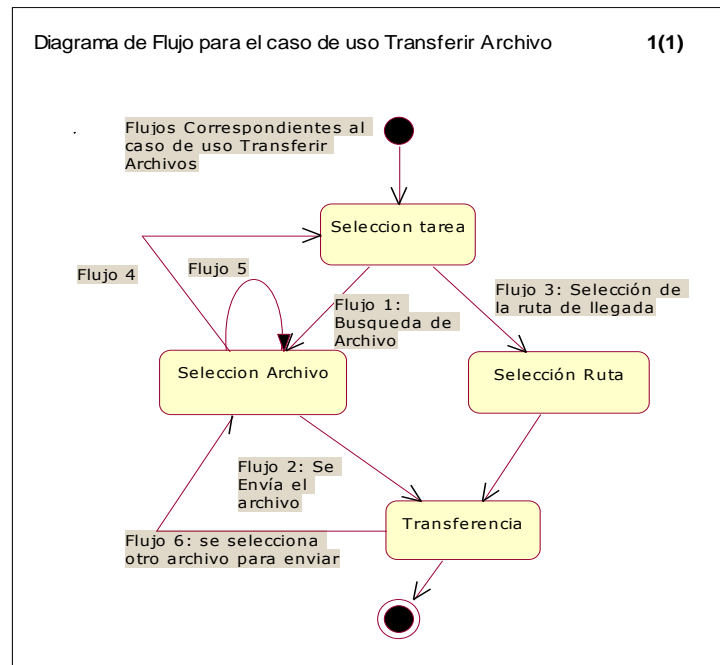


Figura B.5 Diagrama de Flujo para el caso de uso Transferencia de Archivos

**Restricciones de Tiempo:** Ninguna.

### B.2.2 Paquetes de Análisis

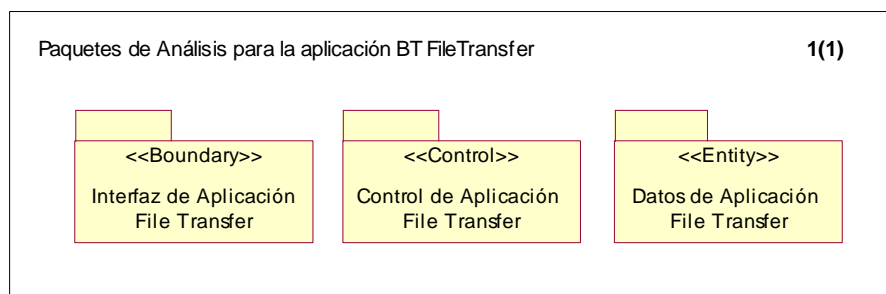


Figura B.6 Diagrama de Paquetes de Análisis para la aplicación BT FileTransfer

### B.2.2.1 Paquete Interfaz de Aplicación BT\_FileTransfer

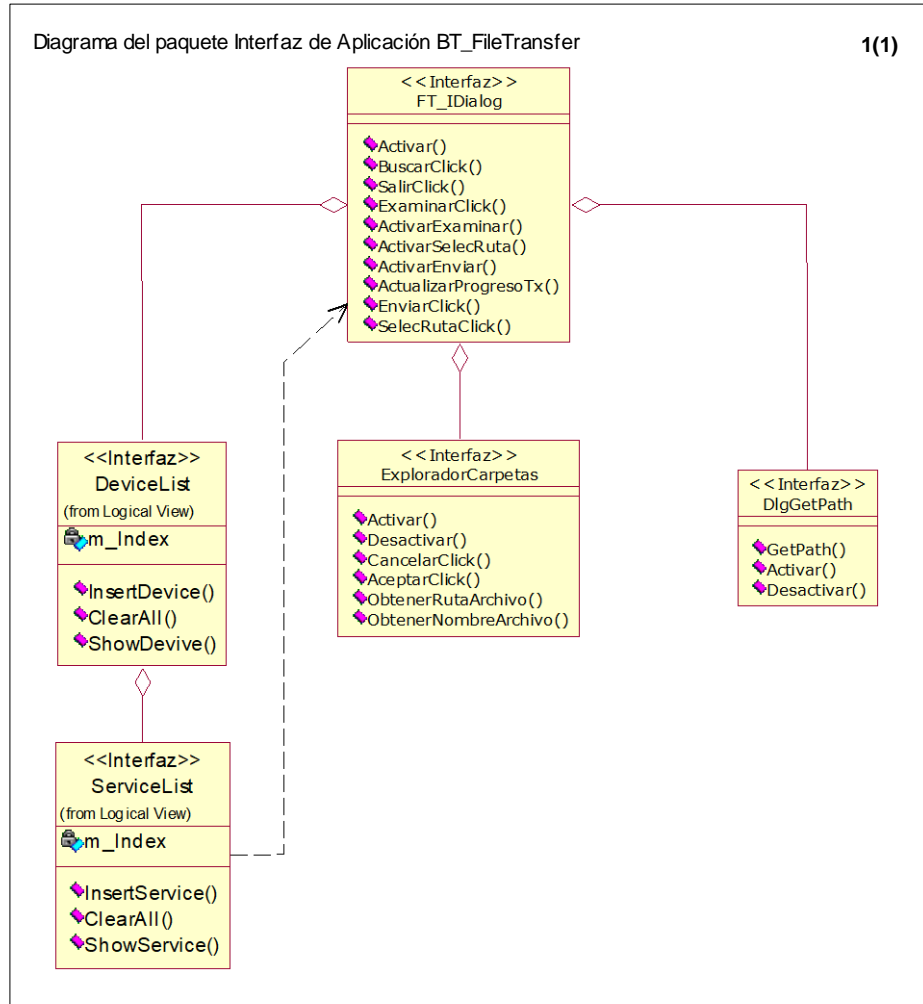


Figura B.7 Diagrama del Paquete Interfaz de Aplicación BT\_FileTransfer

#### B.2.2.1.1 Clase FT\_IDialog

##### Operaciones

**SeleccionarInterfazClick():** Función que permite indicar a la aplicación, el tipo de interfaz o conectar del módulo Bluetooth, esta interfaz puede se Serial RS 232 o USB.

**BuscarDisClick():** Función asociada al botón Buscar Dispositivo. Desencadena el proceso de búsqueda de dispositivo en un entorno próximo.

**SeleccionarDisClick():** Función asociada al botón Seleccionar Dispositivo. Permite especificar cual es el dispositivo con el que se desea establecer la comunicación, en caso de que se encuentre mas de uno en el entorno próximo.

**BuscarServClick():** Función asociada al botón Buscar Servicio. Desencadena el proceso de búsqueda de servicios y sus atributos en el dispositivo seleccionado con la función anterior.

**SeleccionarServClick():** Función asociada al botón Seleccionar Servicio. Permite especificar cual es el servicio al que se quiere tener acceso, en caso de que se encuentre mas de uno en el dispositivo con el que se ha establecido conexión. Esta función desencadena un proceso de registro del servicio en el dispositivo local, con el fin de que este último pueda hacer uso del servicio que encontró.

**ConectarServClick():** Función asociada al botón Conectar. Desencadena el proceso para establecer una conexión con el servicio en el dispositivo remoto.

**ActivarServClick():** Función asociada al botón Activar Servicio. Desencadena el proceso para registrar un servicio en un dispositivo que desea establecerse como servidor, con el fin de que otros dispositivos puedan encontrarlo (al servicio) y hacer uso de el.

**DesactivarServClick():** Función asociada al botón Desactivar Servicio. Desencadena el proceso para remover el registro de un servicio en un dispositivo que ha actuado como servidor, con el fin de que este pueda actuar como cliente en la aplicación.

**ExaminarClick():** Captura el Click del botón "Examinar", debe invocar en la clase de dialogo la función que muestra el cuando de selección del archivo a enviar.

**EnviarClick():** desencadena el envío del archivo en la clase de dialogo de transferencia de archivos

**SeleccRutaClick():** Ocurre al pulsar el botón de Seleccionar Ruta.

**CerrarClick():** Función asociada al botón Cerrar. Desencadena el proceso de finalización y cierre de la aplicación.

**ShowDevice():** Función ordena a la lista de mostrar cuales son los dispositivos encontrados en el entorno.

**ShowService():** Función que ordena a la lista de servicios, mostrar cuales son los servicios encontrados en in dispositivo remoto específico.

#### *B.2.2.1.2 Clase DeviceList*

##### **Operaciones**

**InsertDevice():** Función asociada a la lista de dispositivo, por medio de la cual se agregan dispositivos a la lista, sin que esto implique presentarlos en la interfaz.

**ClearAll():** Función para eliminar todos los registros que la lista de dispositivos ha almacenado.

**ShowDevive():** Orden que se aplica a la lista de dispositivos para presentarlos en la ventana o interfaz de usuario principal.

#### *B.2.2.1.3 Clase ServiceList*

##### **Operaciones**

**Insert Service():** Función asociada a la lista de servicios, por medio de la cual se agregan servicios a la lista, sin que esto implique presentarlos en la interfaz.

**ClearAll():** Función para eliminar todos los registros que la lista de servicios ha almacenado.

**Show Service ():** Orden que se aplica a la lista de servicios para presentarlos en la ventana o interfaz de usuario principal.

#### *B.2.2.1.4 ClaseDlgGetPath*

##### **Operaciones**

**GetPath():** Obtiene la ruta actual que se encuentre seleccionada en el cuadro de diálogo de selección de ruta.

**Activar():** Activa el cuadro de dialogo de selección de Ruta.

**Desactivar():** Deja de mostrar el cuadro de dialogo de selección de ruta.

#### *B.2.2.1.5 Explorador de Carpetas*

##### **Operaciones**

**Activar():** Muestra la estructura de directorios del dispositivo Local.

**Desactivar():** Dejan de mostrarse las carpetas del dispositivo local para que el usuario no pueda seleccionar ninguna.

**CancelarClick():** Pulsando el botón cancelar el usuario puede desistir de la transferencia del archivo; el cuadro de diálogo se cierra..

**AceptarClick():** Hace lo mismo que DobleClickArchivo.

#### *B.2.2.1.6 CajaMsgConfirmacion*

##### **Operaciones**

**Activar():** Activa la caja que indica la "transferencia exitosa".

**AceptarClick():** Cierra la caja que indica la "transferencia exitosa".

**Desactivar():** Deja de mostrar la caja que indica la "transferencia exitosa".

#### **B.2.2.2 Paquete Control de Aplicación BT\_FileTransfe**

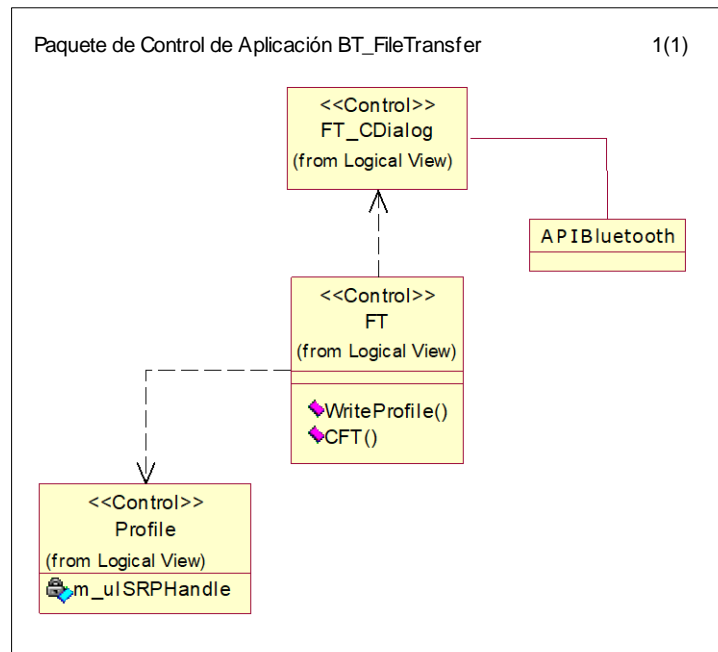


Figura B.8 Diagrama del Paquete de Control de Aplicación BT\_FileTransfer

#### B.2.2.2.1 Clase FT\_Cdialog

Se encarga de realizar las transferencias de archivos, de manejar eventos; representa la clase principal de control de esta aplicación.

#### Operaciones

**OnInquiry():** Función de control que responde al evento del botón de Buscar dispositivos. En esta función se ejecuta la función `HCI_ReqInquiry`, para dar inicio al proceso de indagación por dispositivos Bluetooth remotos.

**OnSelDevice():** Función de control que responde al evento del botón de Seleccionar dispositivos. En esta función se ejecuta la función `SCM_ReqConnect`, para establecer una conexión de datos entre los dos dispositivos.

**OnGetservices():** Función de control que responde al evento del Buscar Servicios. Dentro de esta función se llama a la función `SDC_ReqConnect`, con la cual se establece un



conexión a la base de datos del cliente para ejecutar los procesos de búsqueda de servicios en un dispositivo remoto seleccionado por el usuario.

**OnSelservices():** Función de control que responde al evento del botón de Seleccionar Servicios. En esta función se ejecuta la función DBM\_ReqRegisterService, para registrar en la base de datos, el servicio que ha sido descubierto en el dispositivo remoto seleccionado, así como sus atributos y descriptores específicos.

**OnRegistrar():** Función de control que responde al evento del botón de Activar Servicios. Esta función debe ser ejecutada por el dispositivo que desea suministrar un servicio para que otros dispositivo puedan encontrarlo y hacer uso de el.

**OnDesRegistrar():** Función de control que responde al evento del botón Desactivar Servicio. En esta función se llama a DBM\_ReqUnRegisterService para eliminar un registro de servicio en la base de datos del servidor, con el fin de que este pueda asumir el papel de cliente.

**OnConnect():** Función que permite conectarse al servicio que se ha encontrado en un dispositivo remoto. Para este caso a través de esta función se realiza una llamada a la función Obex\_ReqConnect, con el propósito de establecer una sesión Obex para intercambio de objeto.

**EnviarArchivo():** Inicia el envío del archivo, es decir, el primer trozo de él; debe desencadenar los procesos OBEX de transferencia. Esta función envía la información inicial del archivo para que en el servidor se pueda iniciar la recepción; es decir, envía la primera solicitud PUT hacia el servidor (con el numero de secuencia determinado por la constante "PRIMERA") . La solicitud PUT contiene los encabezados ó bloques de información "Name" y "Length"; El protocolo OBEX determina que puede enviarse también un encabezado "Body" con información de algún trozo de archivo, pero en este caso se ha optado por no enviarlo aún por cuestiones de simplicidad de la implementación. Según lo que se ha estudiado del protocolo OBEX, esta función cumple con los requerimientos de transferencia del protocolo OBEX descritos en la especificación Bluetooth y en la especificación IrOBEX de IrDA.

**OnExaminar():** Funcion que se invoca como respuesta al click del usuario en "examinar", Desencadena el proceso de busqueda y selección del archivo. este proceso de selección

bloquea la ejecución de la función hasta que se elige algún archivo; después de lo cual debe haber una recuperación del nombre y de la ruta del archivo escogido.

**OnEnviar():** Es una función que se ejecuta después de hacer click en el botón enviar de la interfaz. La función abre el archivo, obtiene su nombre y longitud para enviarlo en la primera solicitud PUT mediante la invocación de la función "EnviarArchivo". Luego de invocada esta función, la transferencia se seguirá llevando a cabo a medida que se hagan las solicitudes y lleguen sus correspondientes confirmaciones. La función EnviarArchivo solamente se invoca una vez.

**LeerArchivo():** Lee una cantidad determinada de bytes del archivo a transmitir obtenido mediante el botón "Examinar". En la última invocación de esta función no se encuentra información de archivo puesto que ya se acabó de enviar toda; la función se debe encargar de invocar a la función EnviarFragmento cada vez.

**EnviarFragmento():** Es una función utilitaria que envía los datos de archivo (solo contenido) que se le entregan en un apuntador como argumento y cuya cantidad de información en Bytes está en el argumento longitud. Verifica si la lectura del archivo se ha completado, en cuyo caso manda un encabezado "EndOfBody" en una solicitud PUT con el número de secuencia determinado por la constante "ULTIMA"; de esta manera se reconoce cuando llegue la respuesta. Si la lectura no ha terminado, entonces el apuntador contiene una Longitud en Bytes (diferente de cero) de información del archivo, en este caso se envía una solicitud PUT con el encabezado Body conteniendo dicha información. Se utiliza en este caso un número de secuencia determinado por la constante "INTERMEDIA" para reconocer posteriormente la llegada de la confirmación correspondiente a esta solicitud.

**CrearArchivo():** Crea un archivo con el nombre que llegó en una indicación PUT proveniente de un cliente. Deja el archivo listo para escribir en él cuando lleguen otras solicitudes con contenido de archivo y de esta manera reconstruirlo.

**EscribirArchivo():** Almacena un trozo de contenido de archivo resultante de una indicación PUT proveniente de un cliente. Cada trozo que llegue en una solicitud se escribe consecutivamente en el archivo de destino. La información que se escribe aquí es exactamente la misma que se leyó respectivamente en el cliente. La función debe poder manejar la situación de que le llegue cero bytes de información para poder quedar de

conformidad con el perfil de transferencia de archivos. (según el cual el ultimo encabezado - "EndOfBody" - puede llegar vacío ó no).

**OnSelRuta():** Función que maneja el evento de pulsación del botón Seleccionar Ruta; la función debe abrir el diálogo de obtención de ruta, desde el cual se puede recorrer el árbol de directorios local. cuando se detecta el "OK" del cuadro de dialogo, se captura la ruta seleccionada y la función termina.

**OnCloseapplication():** Función para finalizar y cerrar la aplicación.

**OnDbmRegisterServiceCnf():** Función que permite dar tratamiento a la llegada de un mensaje DBM\_REGISTER\_SERVICE\_CNF, el cual es recibido por un dispositivo (cliente o servidor) cuando se registra un servicio exitosamente. La función de registro es DBM\_ReqRegisterService. Después de que se recibe este mensaje, se pueden registrar los demás atributos del servicio.

**OnDbmAddAttrCnf():** Función que permite dar tratamiento a la llegada de un mensaje DBM\_ADD\_ATTR\_CNF, el cual es recibido por un dispositivo (cliente o servidor) cuando se registran exitosamente los atributos específicos de un servicio a través de la función DBM\_ReqAddAttr. Cabe recordar que los atributos son características de los servicios, como el nombre o los protocolos que emplea para su ejecución.

**OnDbmAddDescriptorCnf():** Función que permite dar tratamiento a la llegada de un mensaje DBM\_ADD\_DESCR\_CNF, el cual es recibido por un por un dispositivo (cliente o servidor) cuando se registran los descriptores específicos de un servicio a través de la función DBM\_ReqAddDescr. Un descriptor es un atributo simple del servicio, como por ejemplo el perfil al que pertenece.

**OnDbmUnRegisterServiceCnf():** Función que permite dar tratamiento a la llegada de un mensaje DBM\_UNREGISTER\_SERVICE\_CNF, el cual es recibido por un por un dispositivo (cliente o servidor) cuando la operación de remoción de un servicio de la base de datos de los servicios (DBM\_ReqUnRegisterService) ha sido exitosa.

**OnHCI\_Usb\_Serial():** Función que recibe la entrada de usuario, relacionada con la selección de la interfaz del PC hacia el módulo Bluetooth (USB-RS232). Un vez el usuario ha seleccionado una de estas dos opciones, se llama a la función SIL\_SetDevice.

**OnHciConfigurePortConfirm():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_CONFIGURE\_PORT\_CNF, el cual se recibe después de realizar un solicitud para establecer la configuración del puerto (HCI\_ReqConfigurePort), es decir de la interfaz del modulo Bluetooth. Esta configuración debe realizarse aún cuando la interfaz escogida sea la de USB.

**OnHciLocalAddressCnf():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_LOCAL\_ADDRESS\_CNF, el cual se recibe después de realizar una solicitud para conocer la dirección del dispositivo local a través de la función HCI\_ReqLocalAddress. Esta función se ejecuta para información del usuario.

**OnHciWriteEncryptionModeCnf():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_WR\_ENCRYPT\_MODE\_CNF, el cual se recibe después de realizar una solicitud HCI\_ReqWrEncryptMode, para habilitar o deshabitar la funcionalidad de encriptación en la conexión.

**OnHciWriteAuthenticationModeCnf():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_WR\_AUTH\_MODE\_CNF, el cual se recibe después de realizar una solicitud HCI\_ReqWrAuthMode, con la que se indica a la aplicación si se requiere ejecutar procesos de autenticación o no.

**OnHciWriteConnectTimeoutCnf():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_WR\_CONNECT\_TMO\_CNF, el cual se recibe después de realizar una solicitud HCI\_ReqWrConnectTmo, con la que se escribe el valor del Tiempo máximo de establecimiento de la conexión.

**OnHciWritePageTimeoutCnf():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_WR\_PAGE\_TMO\_CNF, el cual se recibe después de realizar una solicitud HCI\_ReqWrPageTmo, con la que se especifica el tiempo máximo de respuesta a paginación.

**OnHciWriteCodCnf():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_WR\_COD\_CNF, el cual se recibe después de realizar una solicitud HCI\_ReqWrCod, con la que se especifica la Clase de Dispositivo (El tipo de servicios que puede proporcionar).

**OnHciWriteNameCnf():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_WR\_NAME\_CNF, el cual se recibe después de realizar una solicitud HCI\_ReqWrName, con la cual se puede establecer un nombre para el dispositivo local. La idea de esta función es ofrecer un nombre amistoso al usuario, en lugar de usar la dirección del dispositivo.

**OnHciWriteScanEnableCnf():** Esta función permite dar tratamiento a la llegada del mensaje HCI\_WR\_SCAN\_ENABLE\_CNF, el cual se recibe después de realizar una solicitud HCI\_ReqWrScanEnable, con la cual se establece el modo de escaneo para el dispositivo.

**OnHciInquiryCnf():** Función que permite dar tratamiento a la llegada de un mensaje HCI\_INQUIRY\_CNF, el cual es recibido por un dispositivo cuando después de ejecutar la función HCI\_ReqInquiry, un dispositivo remoto contesta esta solicitud.

**OnHciInquiryEvt():** Función que permite dar tratamiento a la llegada de un mensaje HCI\_INQUIRY\_EVT, el cual es recibido por un dispositivo remoto cuando recibe un solicitud de indagación.

**OnHciRemoteNameCnf():** Función que da tratamiento al mensaje HCI\_REMOTE\_NAME\_CNF, el cual se recibe cuando un dispositivo desea conocer el nombre de un dispositivo remoto que ha contestado a la indagación. En esta función se extrae el nombre de la respuesta y se comunica al usuario por medio de la lista de dispositivos.

**OnObexStartCnf():** Función que permite dar tratamiento a la llegada de un mensaje OBEX\_START\_CNF, el cual es recibido por un dispositivo (cliente o servidor), para informarle que su solicitud OBEX\_ReqStart , para iniciar este componente en el API ha sido exitosa.

**OnObexConnectCnf():** Función que permite dar tratamiento a la llegada de un mensaje OBEX\_CONNECT\_CNF, el cual es recibido por un cliente cuando el servidor acepta su solicitud de conexión (función Obex\_ReqConnect). Después de la llegada de este mensaje, se llama a la función

**OnObexConnectInd():** Función que permite dar tratamiento a la llegada de un mensaje OBEX\_CONNECT\_IND, el cual es recibido por un servidor cuando se le hace una solicitud

de conexión. En esta función se llama a la función `Obex_RspConnect` para entregar una respuesta por parte del servidor a la solicitud de conexión del cliente.

**OnObexRegisterCnf():** Función que permite dar tratamiento a la llegada de un mensaje `OBEX_REGISTER_CNF`, el cual es recibido por un servidor cuando se registra este protocolo (por medio de la función `Obex_ReqRegister`) para que puedan ejecutarse los servicios que se soportan en él. Después de la llegada de este mensaje, se llama a la función `OBEX_ReqFillPdl` con el fin de llenar parámetros específicos del protocolo según los requerimientos del servicio que va a utilizarlo.

**OnObexFillPdlCnf():** Función que permite dar tratamiento a la llegada de un mensaje `OBEX_FILL_PDL_CNF`, el cual es recibido por un servidor cuando se registran parámetros específicos de este protocolo (a través de la función `Obex_ReqRegister`), con el fin de brindar soporte a un servicio en particular.

**OnObexPutCnf():** Esta función se ejecuta con la llegada de un mensaje de confirmación a una solicitud `PUT` previa hacia el servidor. La llegada del mensaje indica que la operación `PUT` en el servidor se llevó a cabo de manera exitosa. La función debe leer el código de respuesta del servidor (de una lista de códigos definidos en el documento `IrOBEX` de `IrDA`), y con este código sabe si puede ó no continuar con la transferencia. La función obtiene el número de secuencia (utilizado en la invocación a la solicitud `OBEX_ReqPut`) y de acuerdo a este número sabe si la solicitud es la primera solicitud, si es una solicitud intermedia ó es la última; la primera solicitud solo lleva el nombre y longitud del archivo, las solicitudes intermedias llevan información del archivo y la última no lleva información, pero sí el encabezado de fin de archivo para terminar la transferencia en el servidor. Cuando llega la confirmación de la última solicitud (con el número de secuencia dado por la constante `"ULTIMA"`) se debe cerrar el archivo para que no ocupe memoria en el equipo cliente. Esta subdivisión en tipos de solicitudes es propia de esta implementación; pero puede no ser la única posible.

**OnObexPutInd():** Es la función que maneja las indicaciones `PUT` provenientes de algún cliente, en este caso la aplicación está actuando como servidor de transferencia de archivos. Debe analizar los encabezados que llegan en la solicitud, si llegan los de longitud y nombre, debe crear el archivo, si llega el encabezado de cuerpo, debe escribirse en el archivo abierto de destino y si llega el fin de archivo debe escribir el último trozo de archivo (en este caso

cero bytes) y cierra el archivo. con cada indicación refresca la información de progreso de la transferencia en la ventana principal.

**OnScmConnectCnf():** Función que permite dar tratamiento a la llegada de un mensaje SCM\_CONNECT\_CNF, el cual es recibido por un dispositivo (cliente), como respuesta afirmativa a la solicitud SCM\_ReqConnect que tiene por finalidad establecer una comunicación de datos entre dos dispositivos.

**OnConnectAcceptInd():** Función que permite dar tratamiento a la llegada de un mensaje SCM\_CONNECT\_ACCEPT\_IND, el cual es recibido por un dispositivo (servidor), para indicarle que otro dispositivo desea establecer una comunicación de datos. En caso de que el servidor acepte, solicita la función SCM\_RspConnectAccept, que desencadena en el otro lado el mensaje de SCM\_CONNECT\_CNF

**OnScmRegisterCnf():** Función que permite dar tratamiento a la llegada de un mensaje SCM\_REGISTER\_CNF, el cual es recibido por un dispositivo (servidor), cuando se ha registrado satisfactoriamente al componente SCM del API; esto con el fin de recibir eventos asíncronos por parte de este componente. La función de solicitud de registro es SCM\_ReqRegister.

**OnScmDeRegisterCnf():** Función que permite dar tratamiento a la llegada de un mensaje SCM\_DEREGISTER\_CNF, el cual es recibido por un dispositivo (servidor), cuando se removido su registro del componente SCM. La función que solicita esta operación es la función SCM\_ReqDeRegister.

**OnSdcStartCnf():** Esta función permite dar tratamiento a la llegada del mensaje SDC\_START\_CNF, el cual se recibe después de realizar una solicitud SDC\_ReqStart (si la solicitud ha sido aceptada), la cual tiene por finalidad inicializar el componente SDC (en el cliente), para que sea posible enviar y recibir mensajes de este componente.

**OnSdcConnectCnf():** Esta función permite dar tratamiento a la llegada del mensaje SDC\_CONNECT\_CNF, el cual se recibe después de realizar una solicitud SDC\_ReqConnect, si la solicitud ha sido aceptada. La función SDC\_ReqConnect, permite establecer una conexión con la base de datos del cliente, empleada en el descubrimiento de servicios. La respuesta proporciona un identificador de enlace con la base de datos.

**OnSdcServiceSearchCnf():** Esta función permite dar tratamiento a la llegada del mensaje SDC\_SERVICE\_SEARCH\_CNF, el cual se recibe después de realizar una solicitud SDC\_ReqServiceSearch, cuando ha sido posible encontrar servicios que coincidan con el patrón de búsqueda que se especifica en la solicitud. La función SDC\_ReqServiceSearch, se emplea para preguntar a los dispositivos remotos por la presencia de servicios en sus bases de datos. El mensaje respuesta, contiene una lista de los identificadores de servicio encontrados en el dispositivo remoto.

**OnSdcServiceAttrCnf():** Esta función permite dar tratamiento a la llegada del mensaje SDC\_SERVICE\_ATTR\_CNF, el cual se recibe después de realizar una solicitud SDC\_ReqServiceAttr. La función SDC\_ReqServiceSearch, se emplea para preguntar a los dispositivos remotos por los atributos de un servicio específico que ha sido descubierto con anterioridad. El mensaje respuesta, contiene una lista de los identificadores de los atributos del servicio que se especifica en la solicitud.

**OnSdcDisconnectCnf():** Esta función permite dar tratamiento a la llegada del mensaje SDC\_DISCONNECT\_CNF, el cual se recibe después de realizar una solicitud SDC\_ReqDisconnect, si la solicitud ha sido aceptada. La función SDC\_ReqDisconnect, cierra la conexión hacia la base de datos para descubrimiento de servicios, presente en el cliente.

**OnSdsStartCnf:** Esta función permite dar tratamiento a la llegada del mensaje SDS\_START\_CNF, el cual se recibe después de realizar una solicitud SDC\_ReqStart (si la solicitud ha sido aceptada), la cual tiene por finalidad inicializar el componente SDS (en el servidor), para que sea posible enviar y recibir mensajes con este componente.

**OnSilSetDeviceCnf():** Función que permite dar tratamiento a la llegada de un mensaje SIL\_SET\_DEVICE\_CNF, el cual es recibido por un por un dispositivo (cliente o servidor) cuando la operación de establecimiento de la interfaz del módulo Bluetooth ha sido aceptada (SIL\_SetDevice). Después de recibir esta respuesta, la aplicación puede dar inicio al establecimiento de parámetros de configuración necesario para el establecimiento de una conexión Bluetooth.



#### *B.2.2.2.2 Clase FT*

##### **Operaciones**

**CFT():** Por medio de esta función se establece una instancia del servicio de transferencia de archivos. Una vez se ha determinado que es posible hacer uso de este servicio ofrecido por un dispositivo remoto, se emplea la función DBM\_ReqRegisterService, para registrarlo en la base de datos del cliente y posteriormente se invocan las funciones relacionadas con la búsqueda de atributos del servicio (ofrecidas por la clase FT\_Cdialog) con el fin de completar el registro correspondiente.

**WriteProfile():** Una vez se conocen los atributos del servicio, se invoca esta función para almacenarlos en la base de datos del cliente y este pueda proceder con la operación de conexión al servicio en el dispositivo remoto.

#### *B.2.2.2.3 Clase Profile*

##### **Operaciones**

Las operaciones que involucra esta clase están orientadas al almacenamiento de atributos de servicio en la base de datos del cliente, por lo cual emplean las funciones DBM\_ReqAddDescr y DBM\_ReqAddAttr para introducir características del servicio según se describe a continuación:

**AddAttr\_ProtocolDescrList():** Permite introducir la lista de protocolos que emplea un servicio para poder ejecutarse.

**AddAttr\_ServiceName():** Permite introducir el nombre de un servicio, de manera que la forma en que se presenta al usuario sea mas amistosa y clara.

**AddDesc\_ServiceClassIDList():** Permite introducir el identificador del perfil al que pertenece un servicio en particular.

#### **B.2.2.3 Paquete Entidad de Aplicación BT\_FileTransfer**

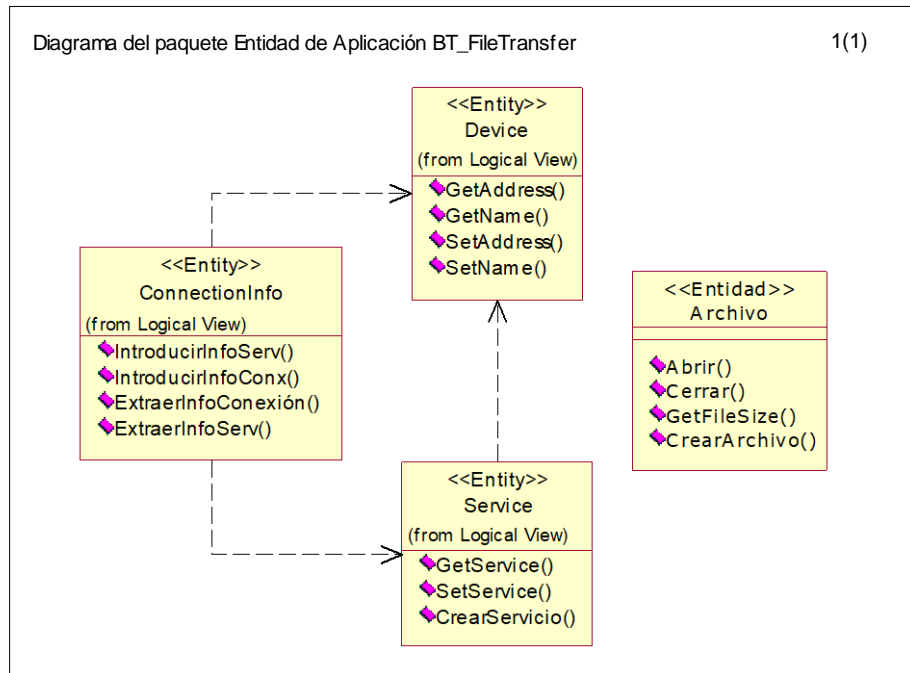


Figura B.9 Diagrama del paquete Entidad de Aplicación BT\_FileTransfer

#### B.2.2.3.1 Clase ConnectionInfo

##### Operaciones

**IntroducirInfoConexión():** Función que permite almacenar temporalmente los datos de la conexión que se establece con otro dispositivo (incluyendo la conexión hacia las bases de datos cuando se trata de realizar descubrimiento de servicios). También tiene como propósito almacenar información temporal relacionada con un dispositivo, en el momento en que se descubre al realizar las funciones de Inquiry.

**ExtraerInfoConexión():** Permite recuperar la información que se almacena con la función IntroducirInfoConexión().

**IntroducirInfoServicio():** Función que permite almacenar temporalmente los datos de los servicios que se encuentran al momento de ejecutar su búsqueda.

**ExtraerInfoServ():** Permite recuperar la información que se almacena con la función IntroducirInfoServicio().

#### *B.2.2.3.2 Clase Device*

##### **Operaciones**

**SetAddress():** Permite almacenar la dirección de un dispositivo remoto con el cual establecemos conexión para hacer uso de uno de sus servicios.

**SetName():** Permite almacenar el nombre de un dispositivo remoto con el cual establecemos conexión para hacer uso de uno de sus servicios.

**GetAddress():** Permite recuperar la información que se almacena con la función SetAddress().

**GetName():** Permite recuperar la información que se almacena con la función SetName().

#### *B.2.2.3.3 Clase Service*

##### **Operaciones**

**SetService():** Permite almacenar la información relevante de un servicio presente en un dispositivo remoto.

**GetService():** Permite recuperar la información que se almacena con la función SetService().

**CreateService():** Permite instanciar la clase servicio.

#### *B.2.2.3.4 Clase Archivo*

Clase que contiene la funcionalidad de manejo dinámico de archivos, son funciones de la MFC de Windows.

##### **Operaciones**

**Abrir():** Abre un Archivo dada su ruta y nombre, establece un Handle al archivo que permite realizar operaciones en él (tales como escritura, lectura y cierre).

**Cerrar():** Cierra el archivo especificado por el Handle que se le entrega como parámetro, esto con el fin de no ocupar más recursos del sistema con el archivo abierto una vez se haya terminado de trabajar con él.

**GetFileSize():** Obtiene la longitud en Bytes que tiene el archivo; es una función de la MFC de windows.

**CrearArchivo():** Crea un archivo con el nombre especificado y en la ruta por defecto seleccionada en el servidor mediante el botón de selección de ruta.

### **B.2.3 Diagrama de Clases de Análisis**

#### **B.2.3.1 Establecimiento de la Conexión**

*Ver página siguiente...*

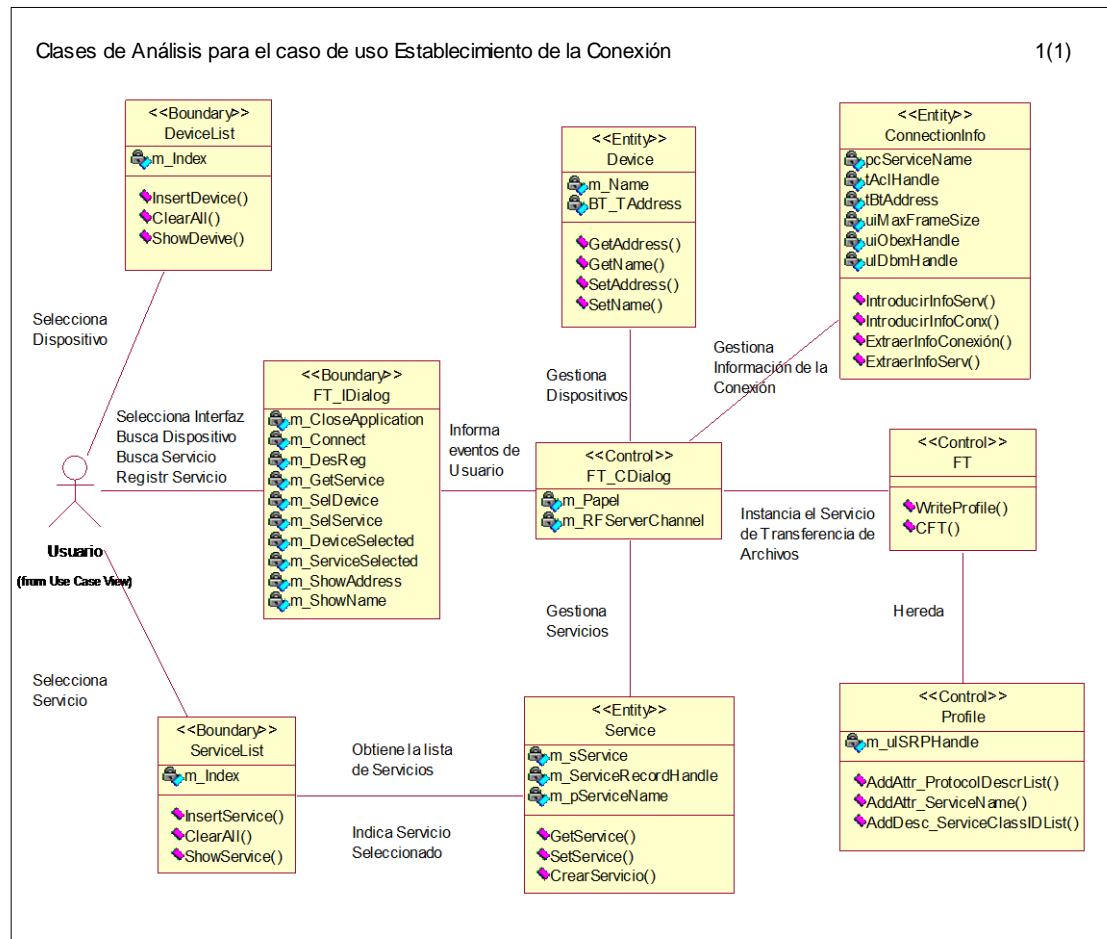


Figura B.10 Clases de Análisis para el caso de uso Establecimiento de la Conexión

### B.2.3.2 Transferir de Archivos

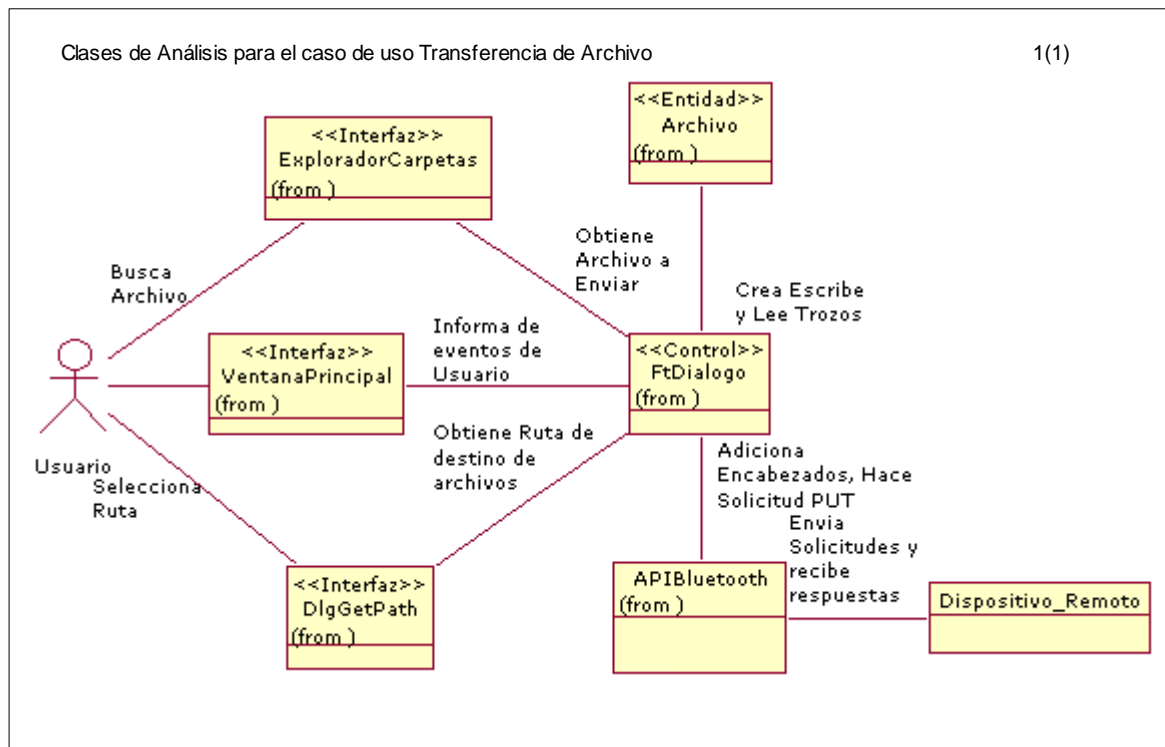
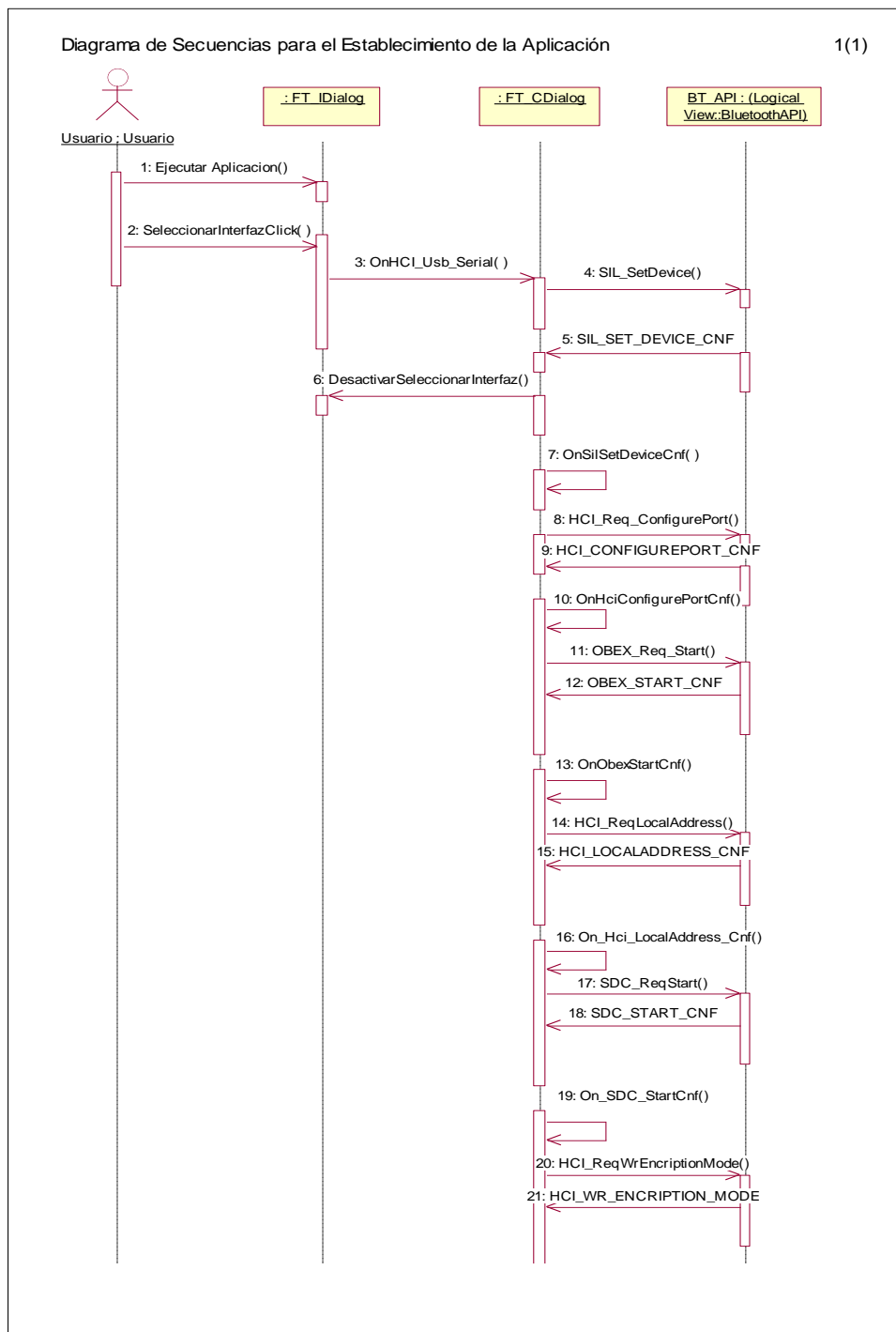


Figura B.11 Clases de Análisis para el caso de uso Transferencia de Archivo

## B.2.4 Diagramas De Interacción Del Sistema

### B.2.4.1 Establecimiento de la Conexión

#### B.2.4.1.1 Diagrama de secuencias



Continuación ...

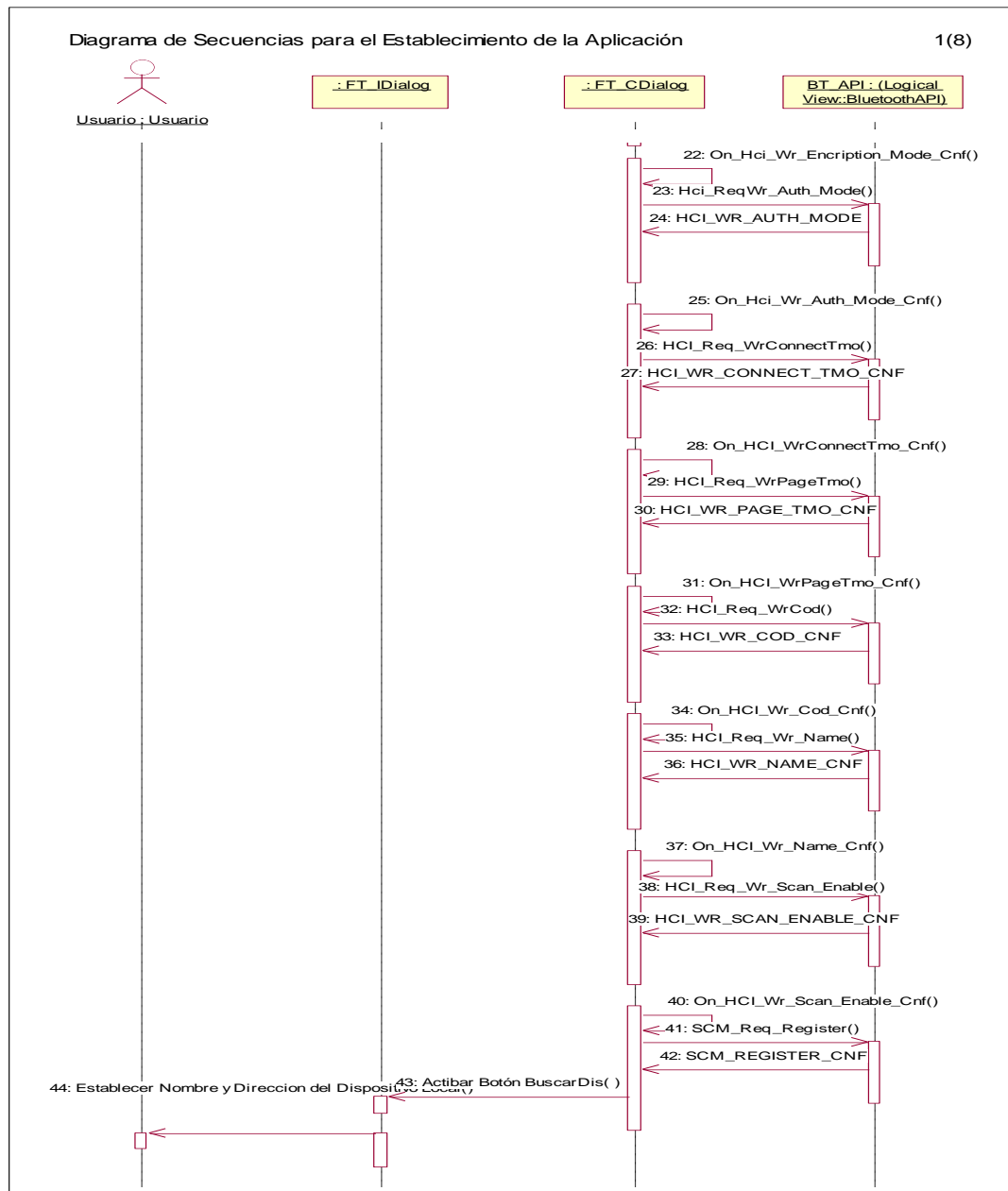


Figura B.12 CSM Inicialización de la Aplicación



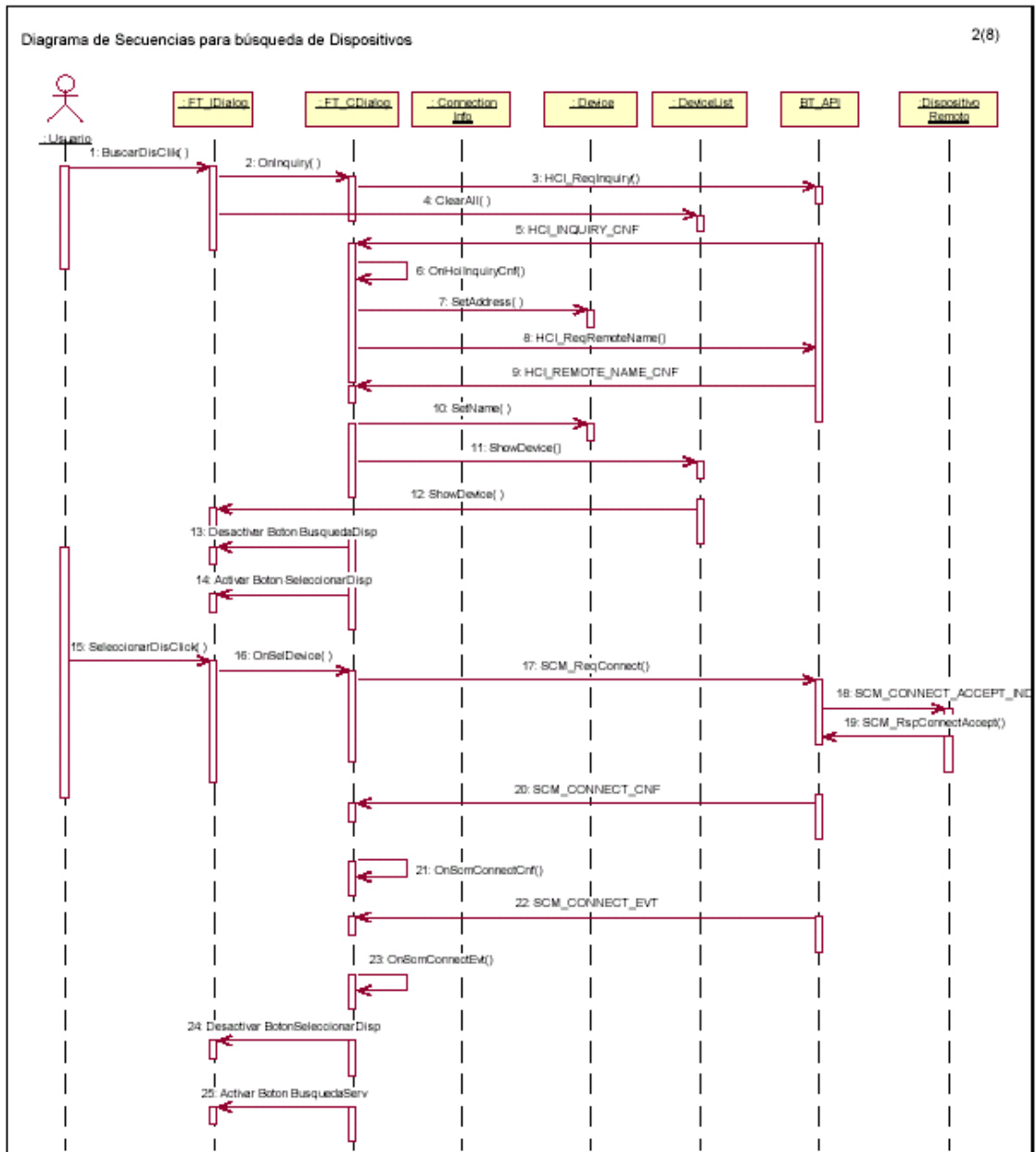


Figura B.13 CSM Búsqueda de Dispositivos

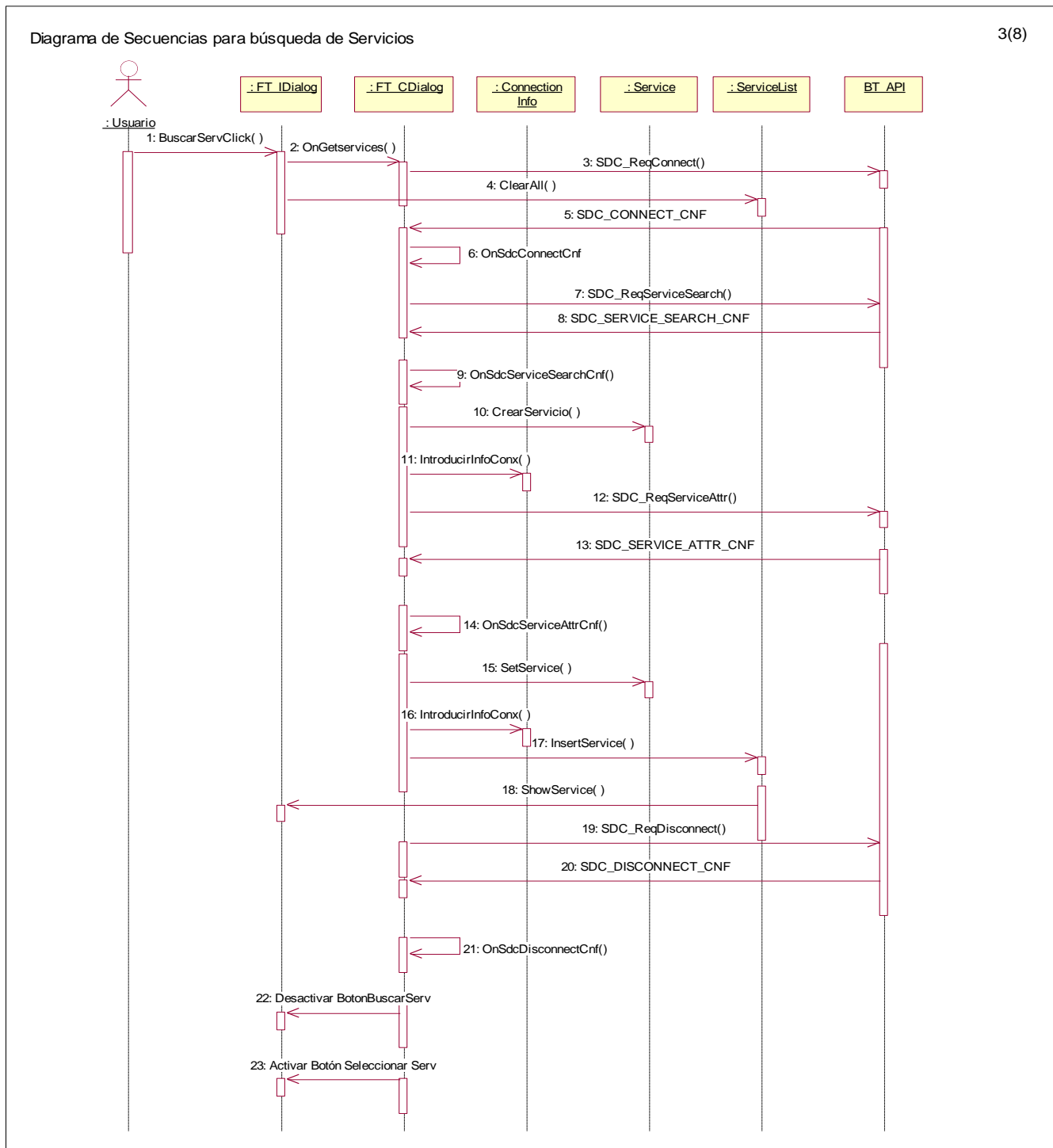


Figura B.14 CSM Búsqueda del Servicio

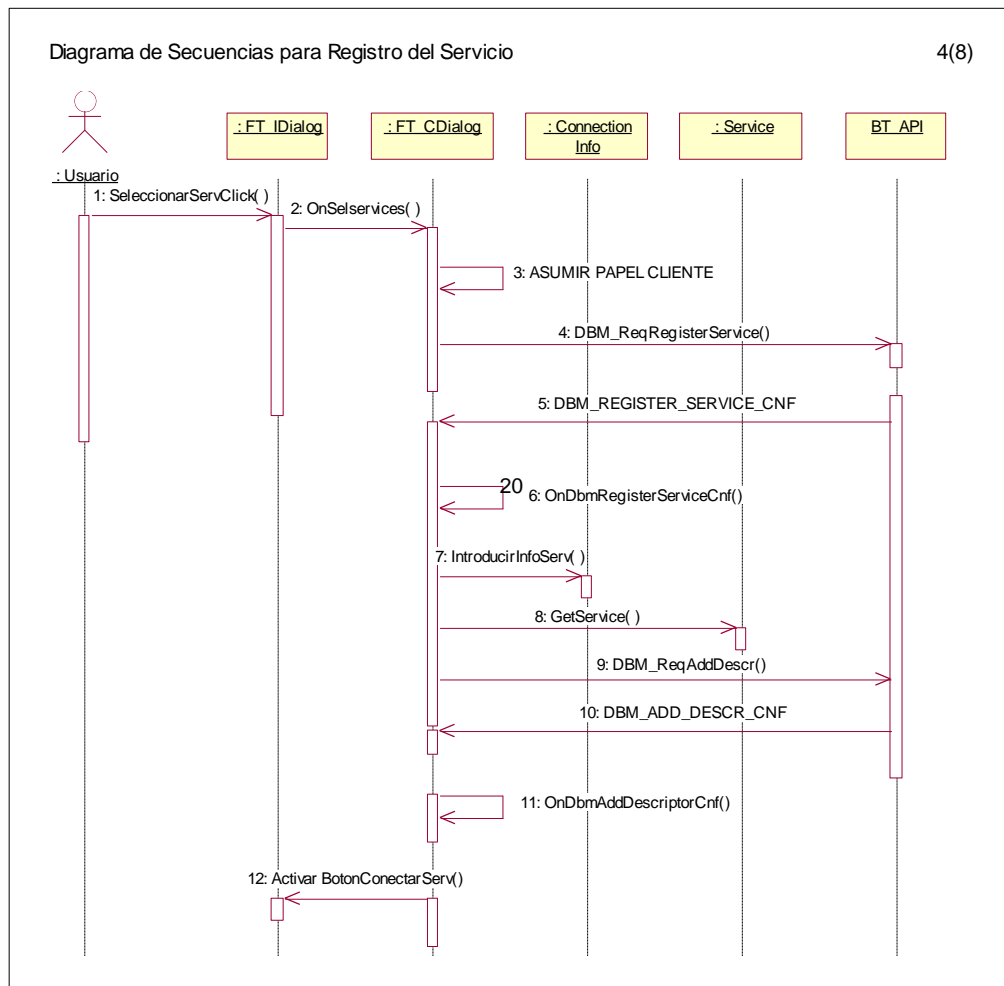


Figura B.15 CSM Registro del Servicio (Cliente)

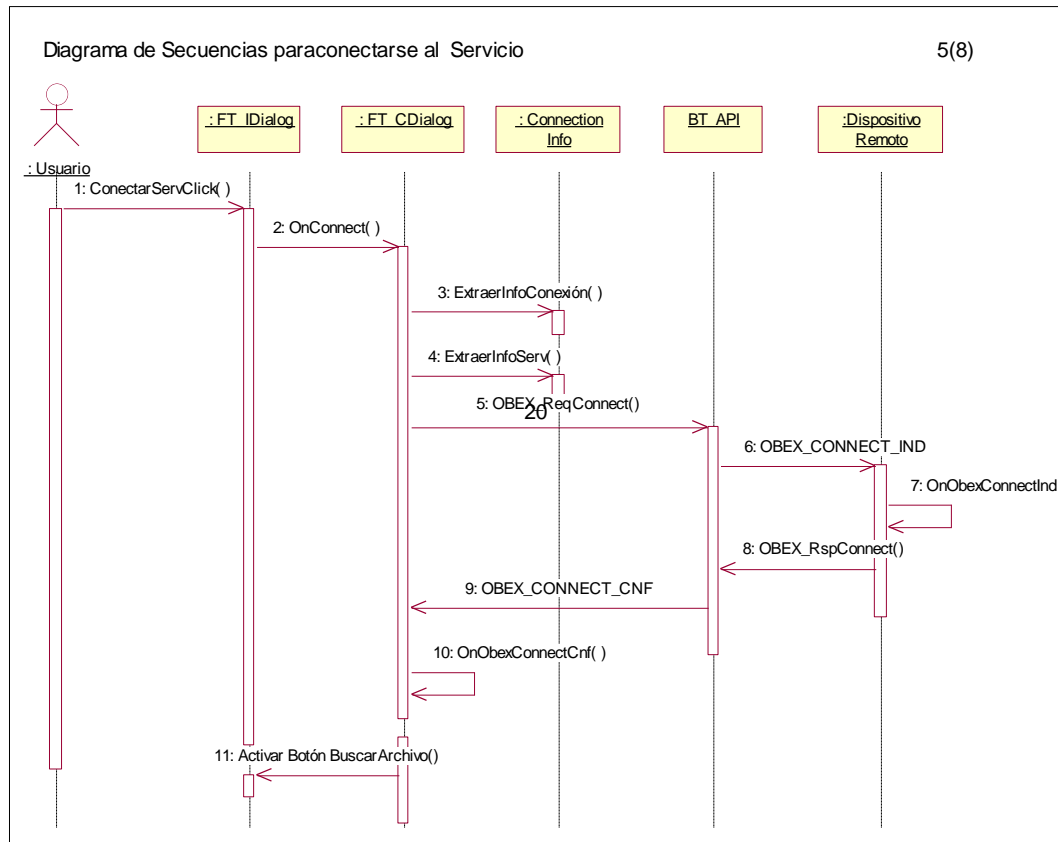


Figura B.16 CSM Conexión al Servicio

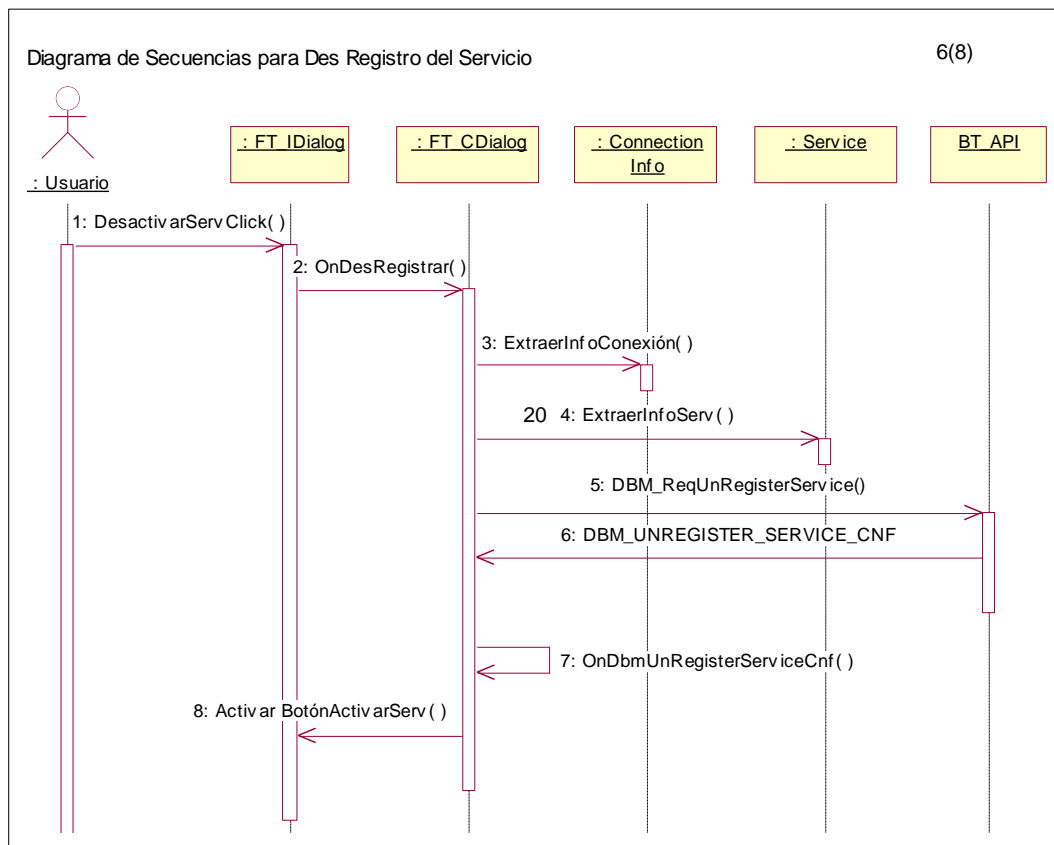


Figura B.17 CSM Des Registro del Servicio

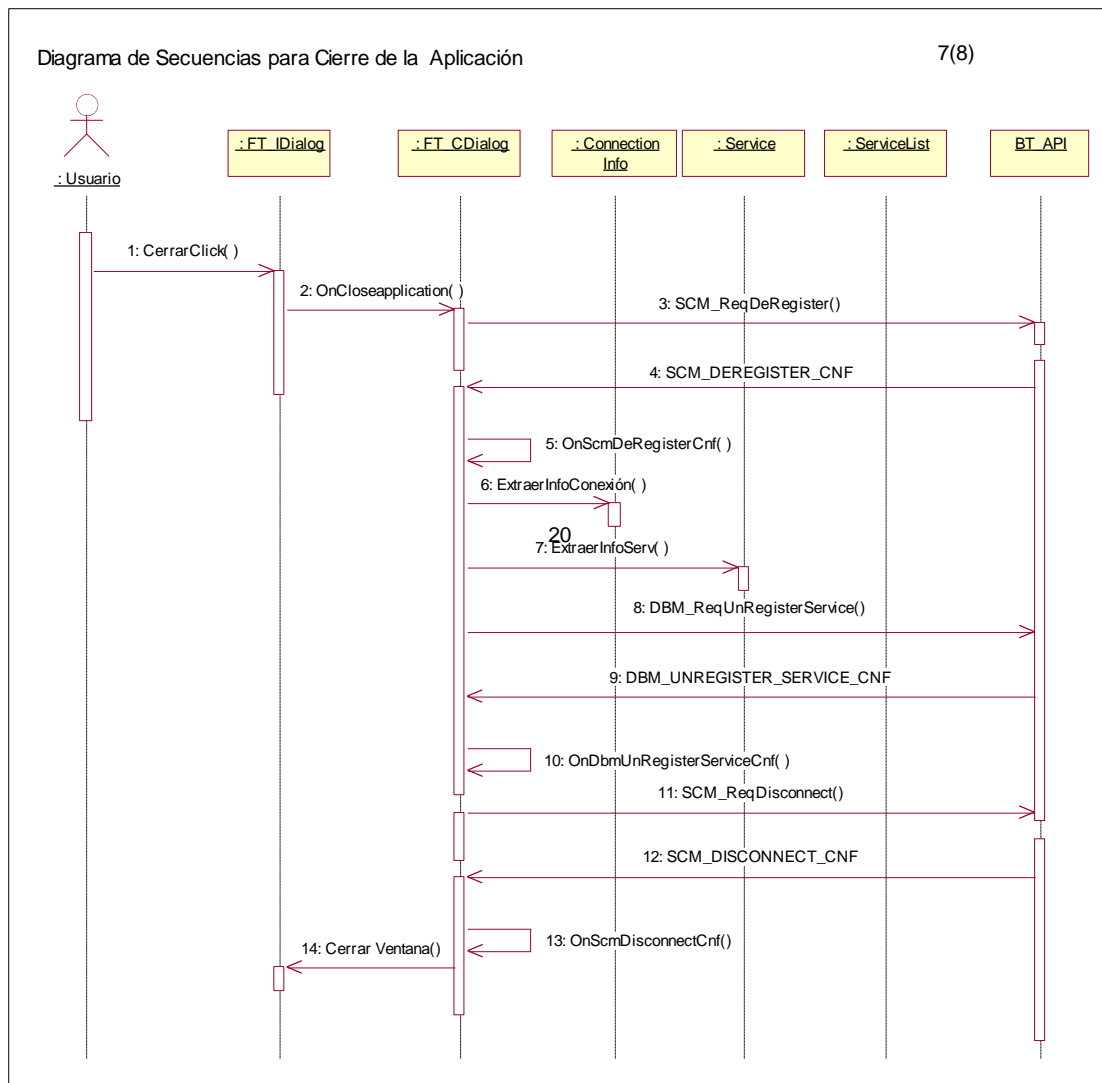


Figura B.18 CSM Cierre de la Aplicación

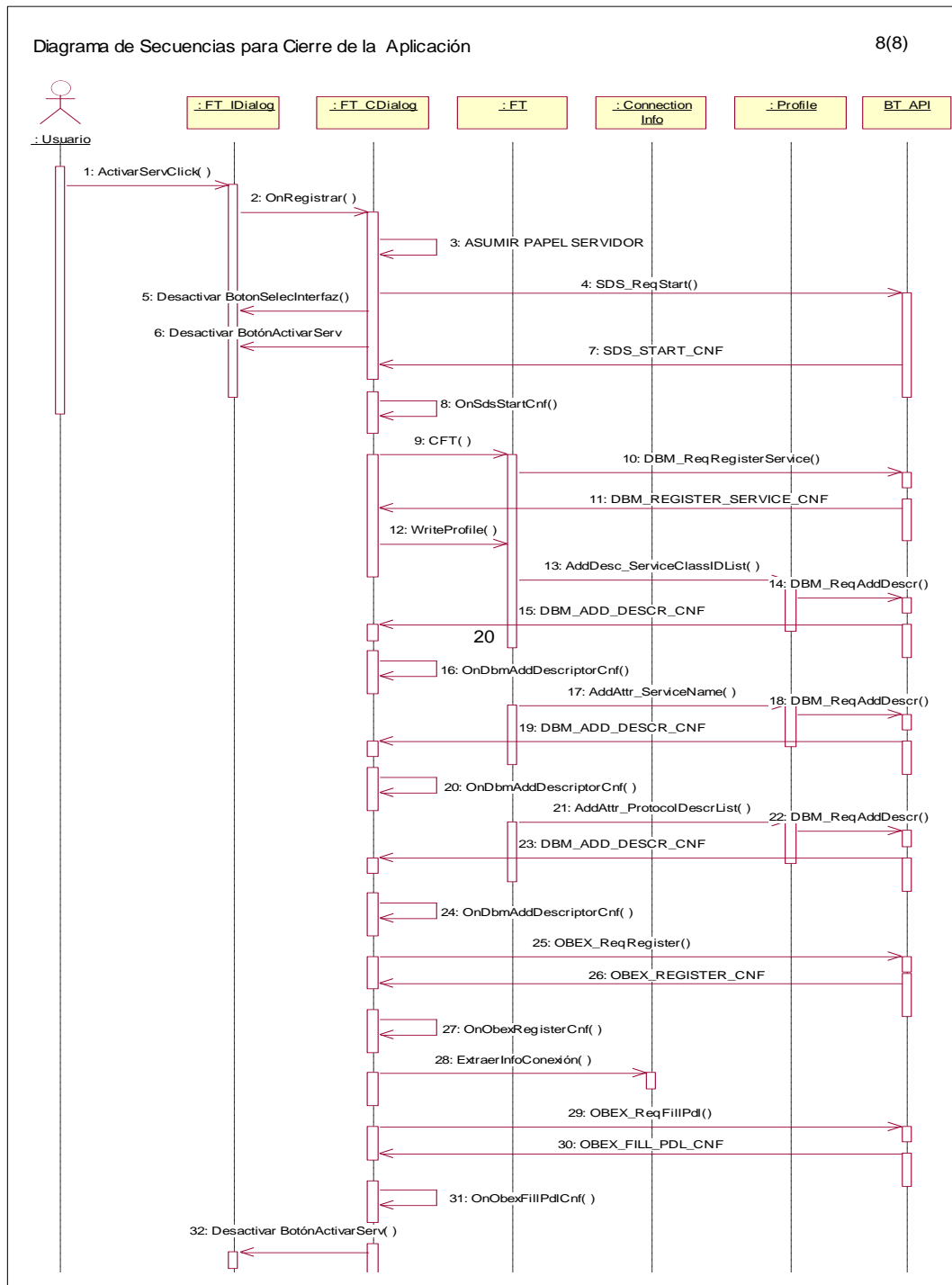


Figura B.19 CSM Registro del Servicio (Servidor)

B.2.4.1.2 Diagramas de colaboración

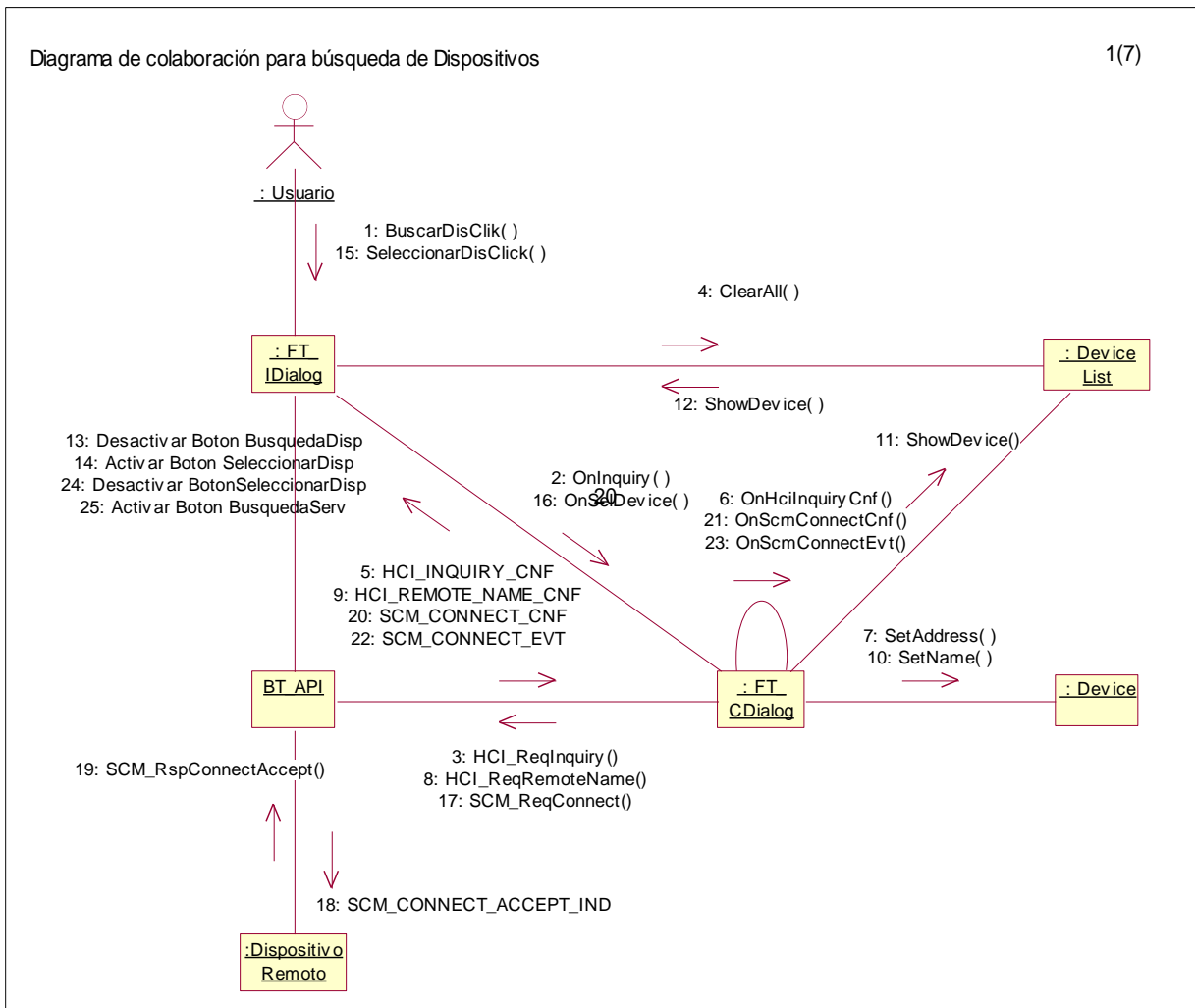


Figura B.20 Diagrama de colaboración Búsqueda de Dispositivos



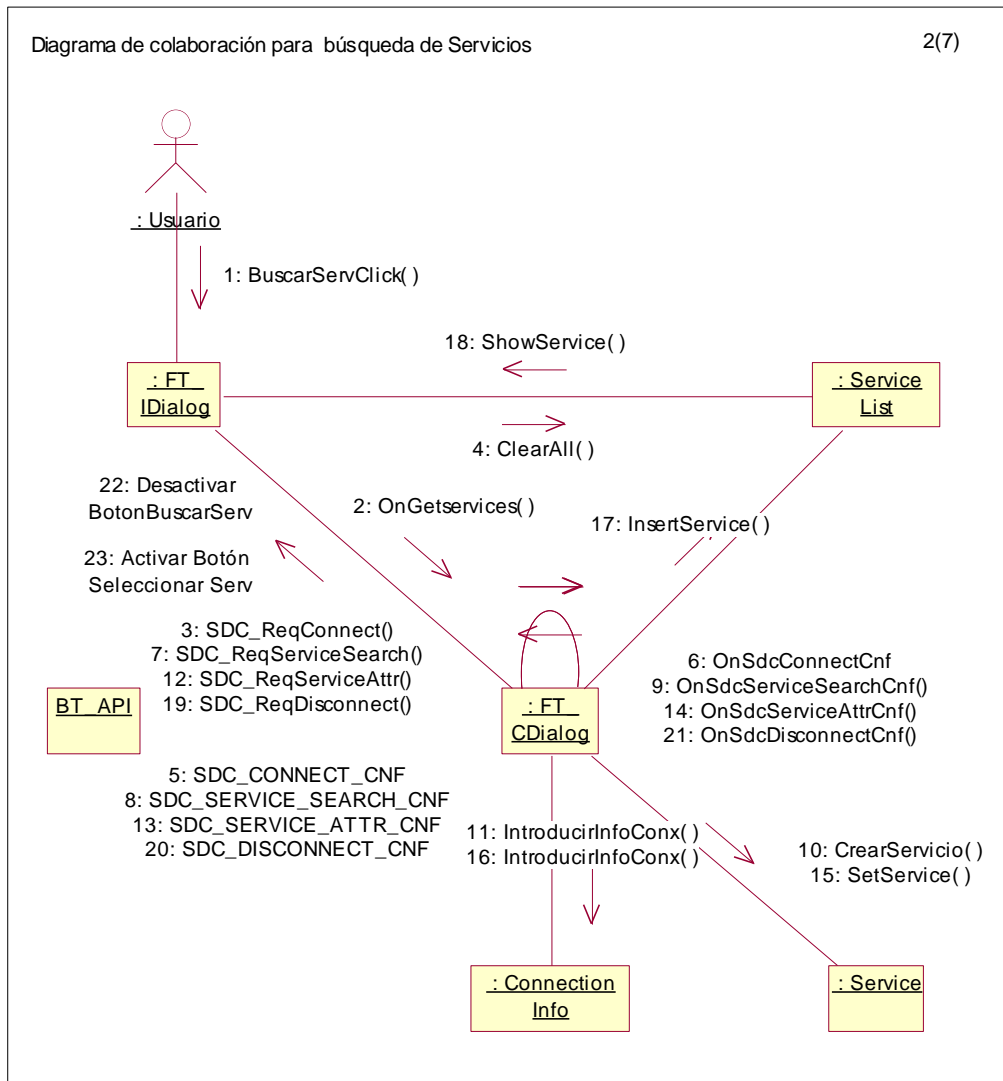


Figura B.21 Diagrama de colaboración para Búsqueda del Servicio

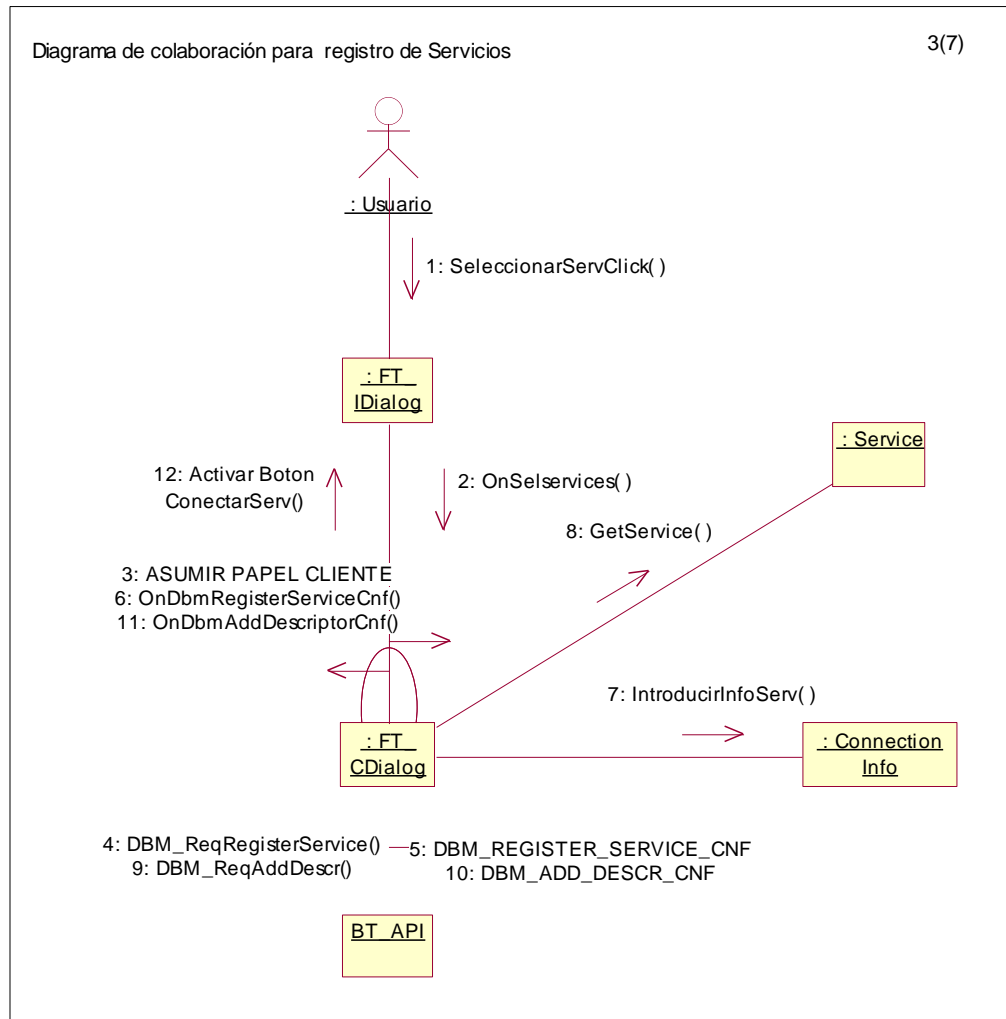


Figura B.22 Diagrama de colaboración para Registro del Servicio (Cliente)

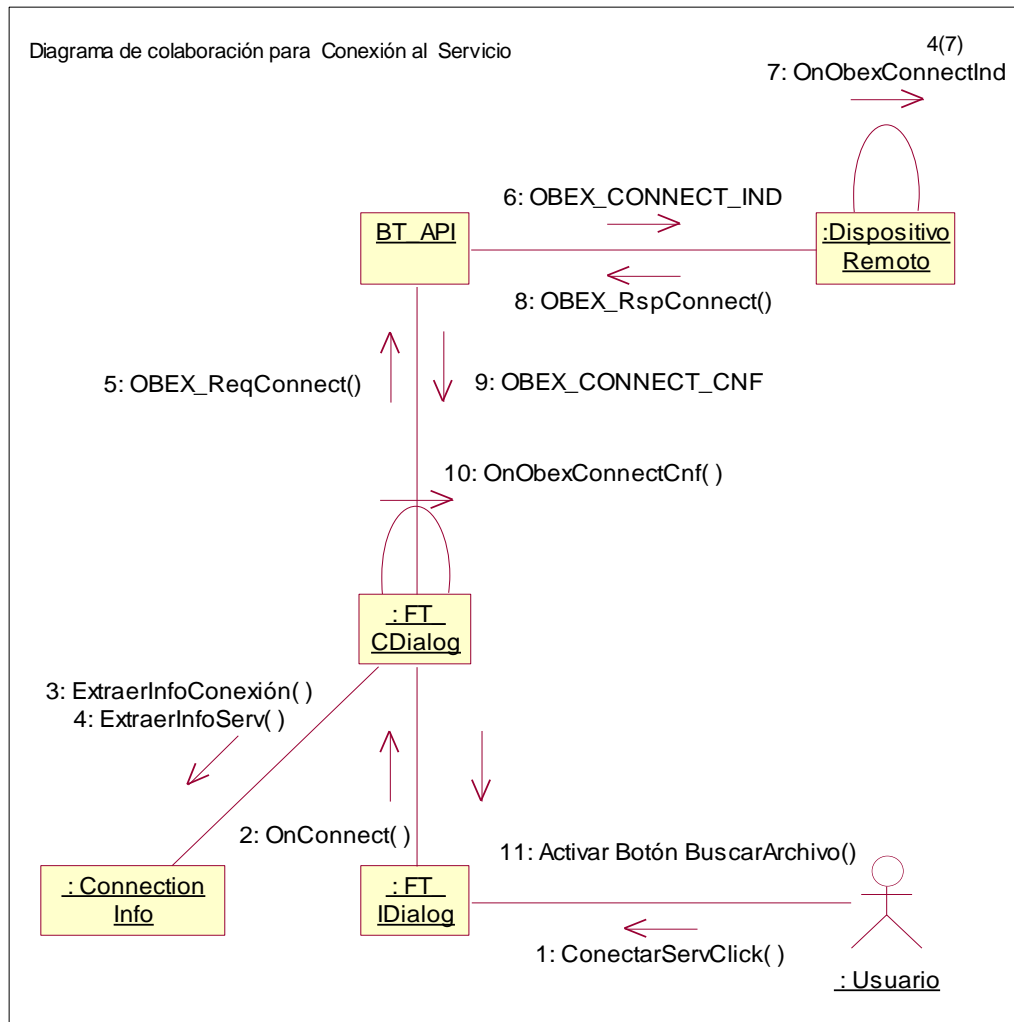


Figura B.23 Diagrama de colaboración para Conexión al Servicio

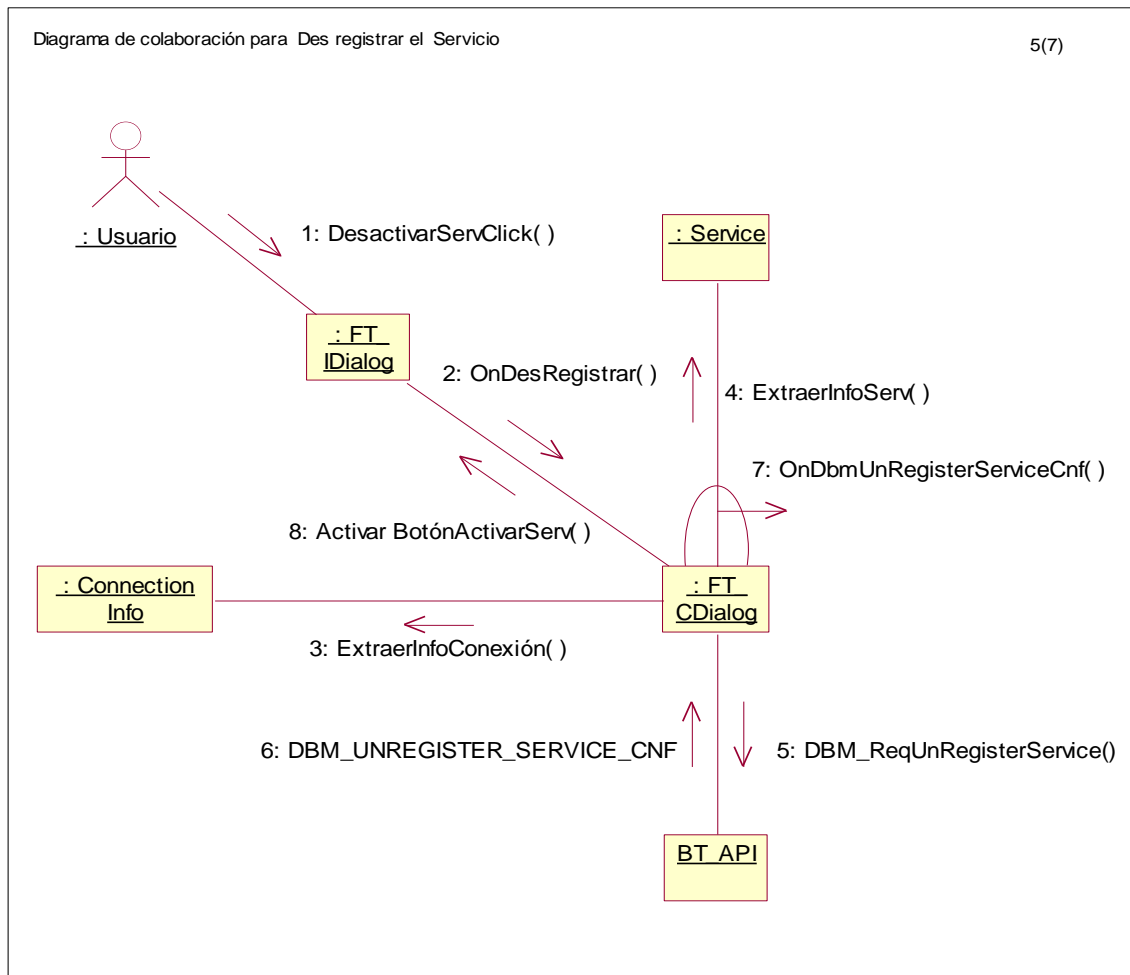


Figura B.24 Diagrama de colaboración Des Registrar el servicio

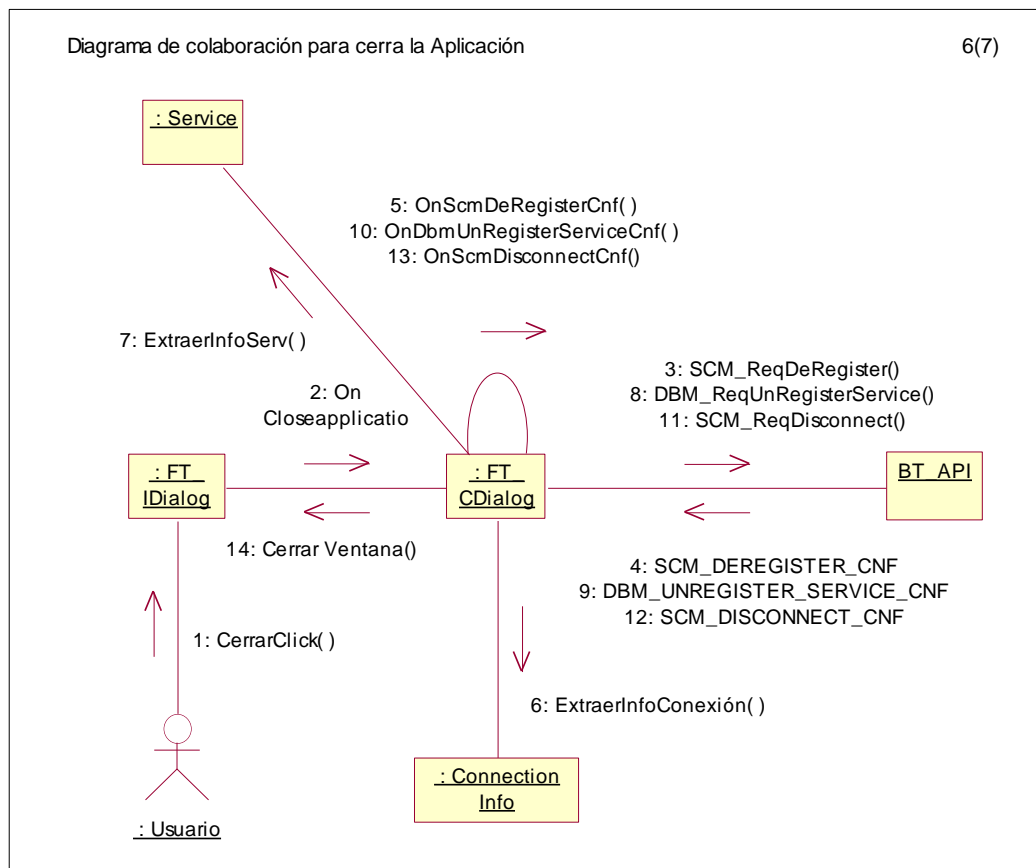


Figura B.25 Diagrama de colaboración Cerrar Aplicación

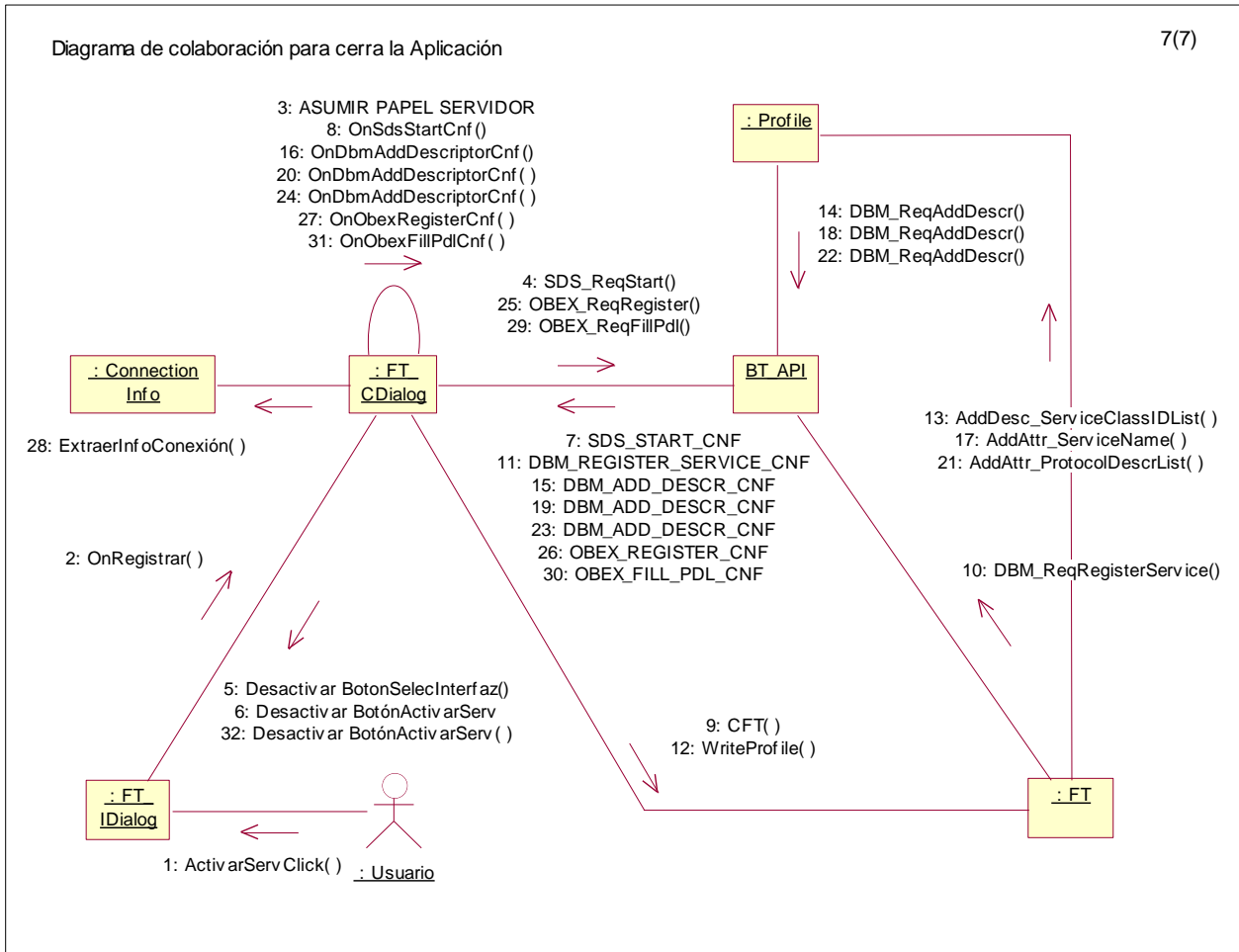


Figura B.26 Diagrama de colaboración Registro del Servicio

**B.2.4.2 Transferir Archivo**

**B.2.4.2.1 Diagrama de secuencias**

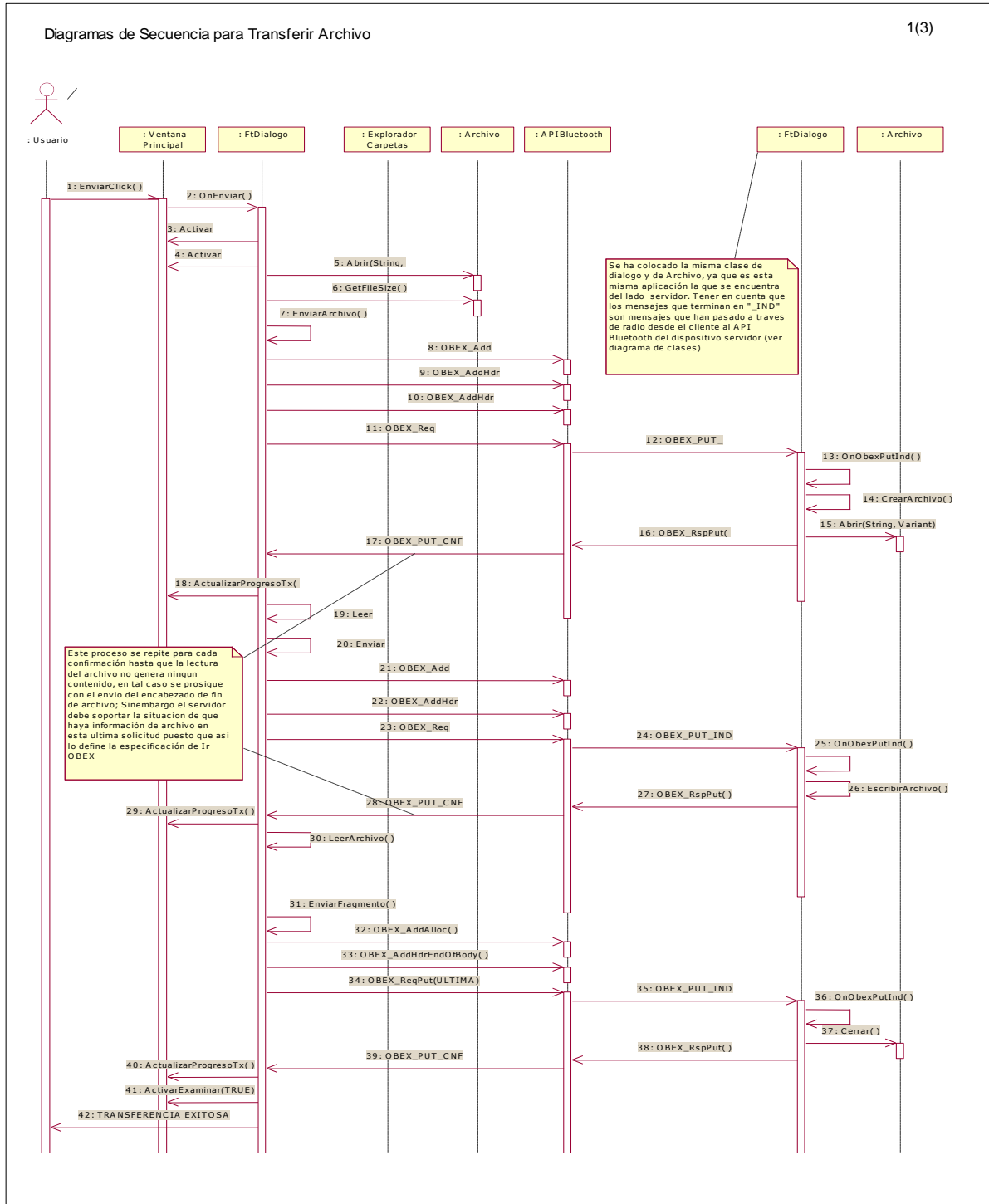


Figura B.27 CSM para el caso de uso Transferencia de Archivo

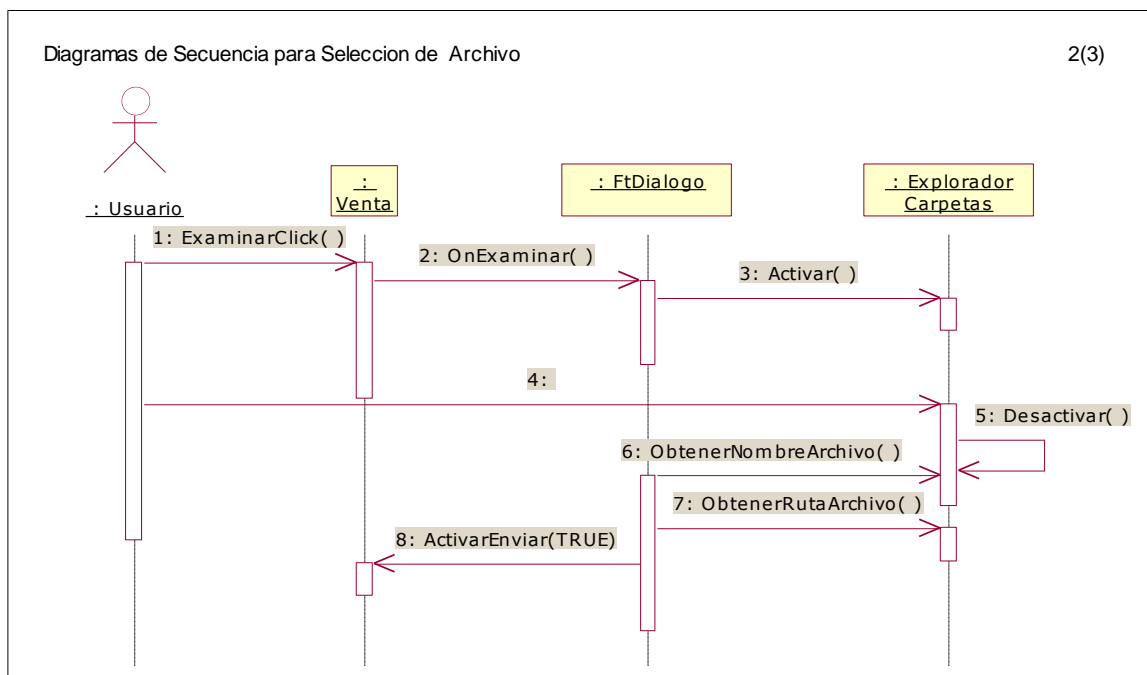


Figura B.28 CSM Selección de Archivo



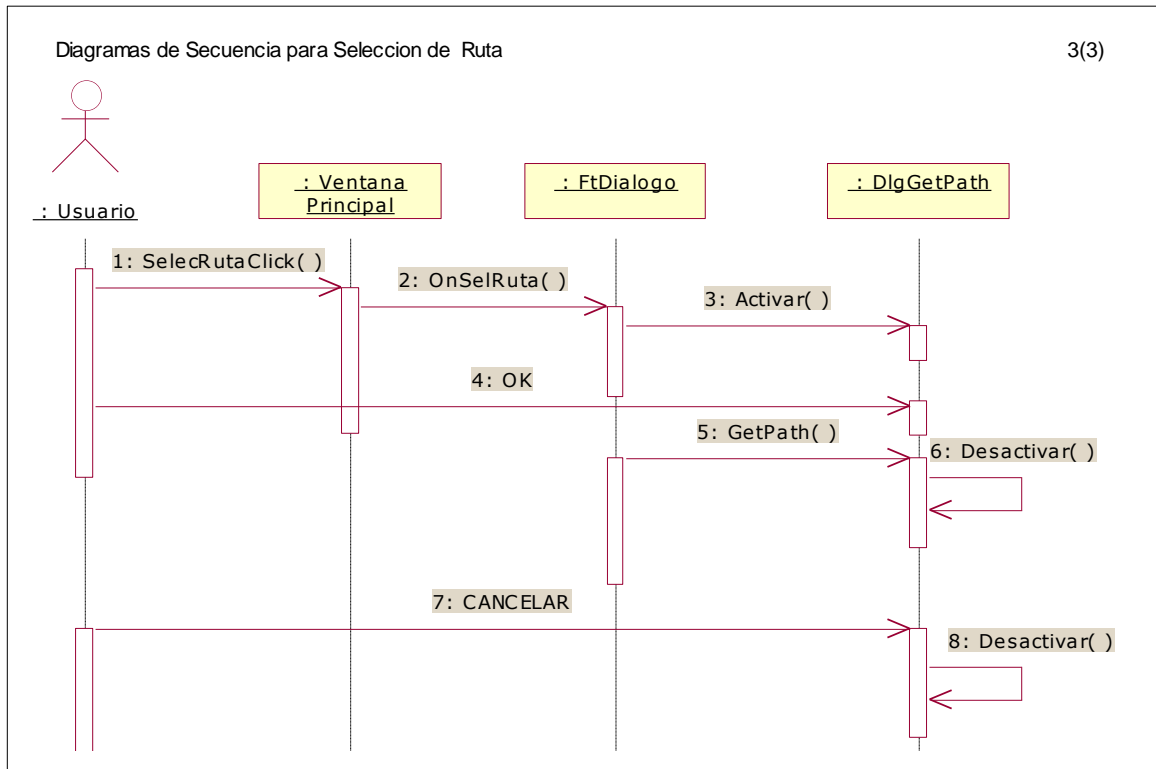


Figura B.29 CSM para Seleccionar Ruta

B.2.4.2.2 Diagramas de colaboración

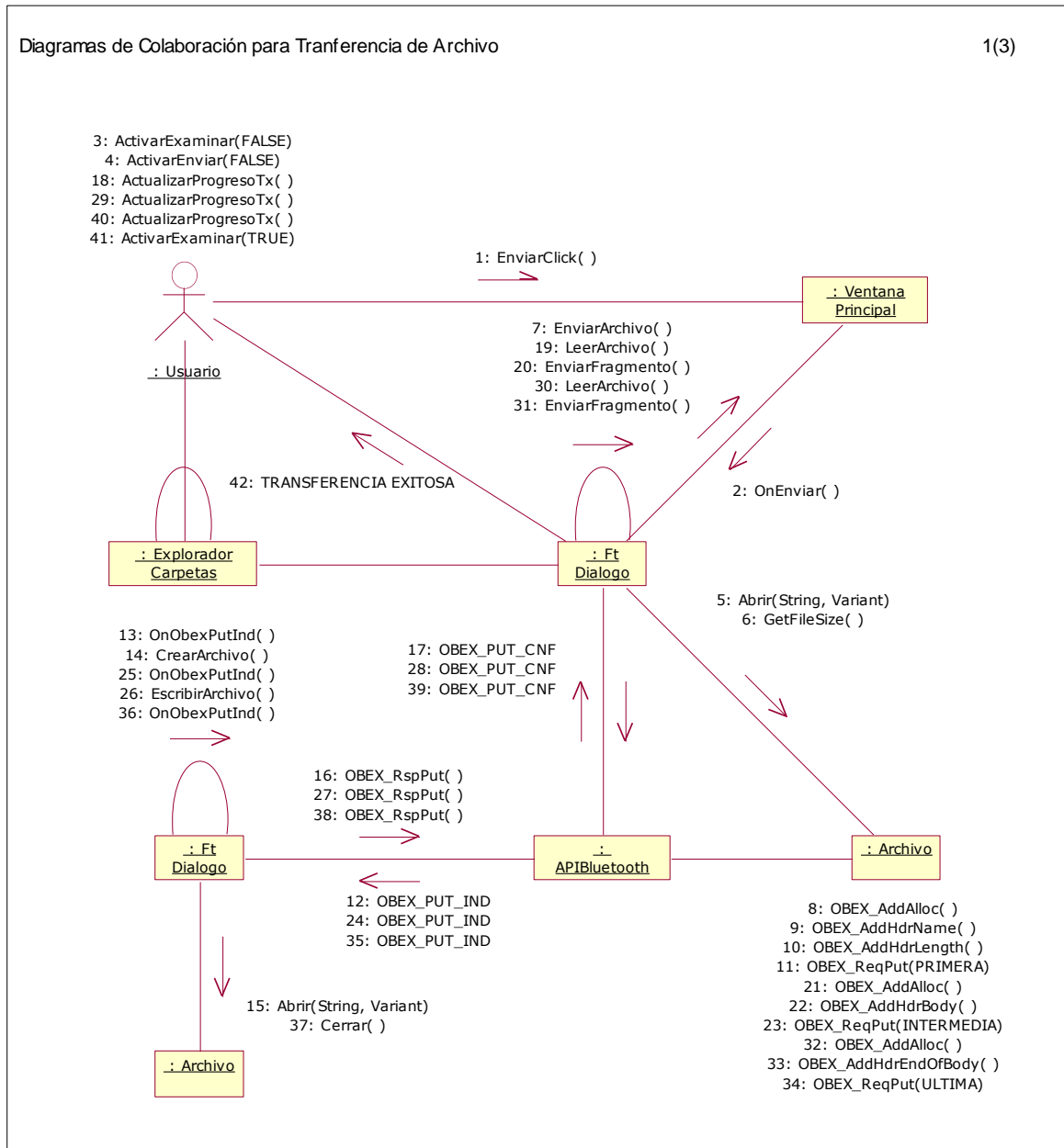


Figura B.30 Diagrama de colaboración para Transferencia de Archivo

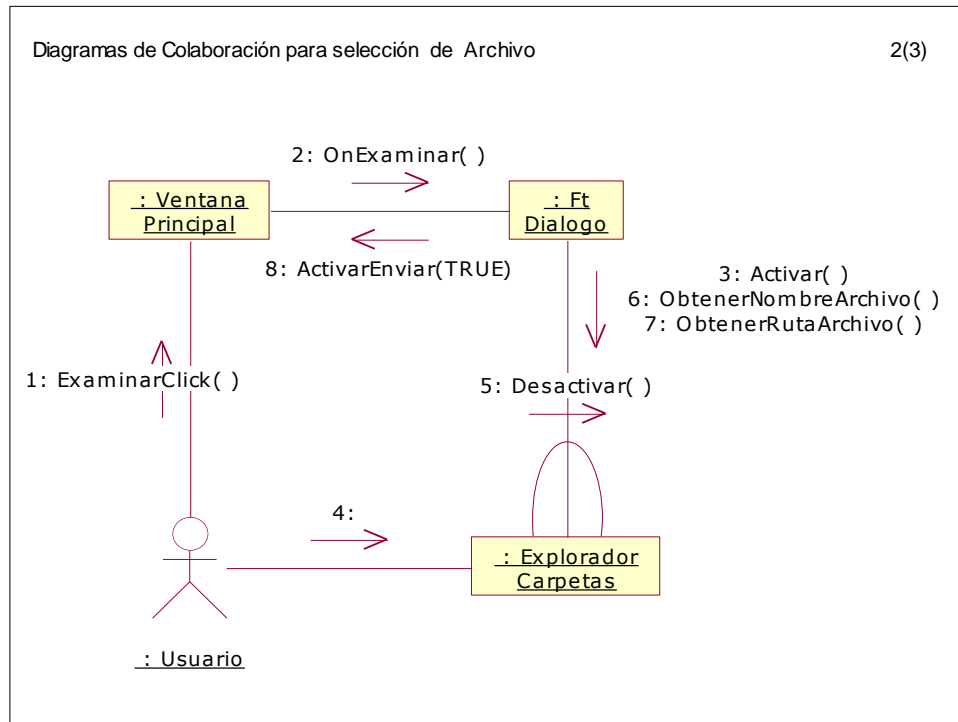


Figura B.31 Diagrama de colaboración para Seleccionar Archivo

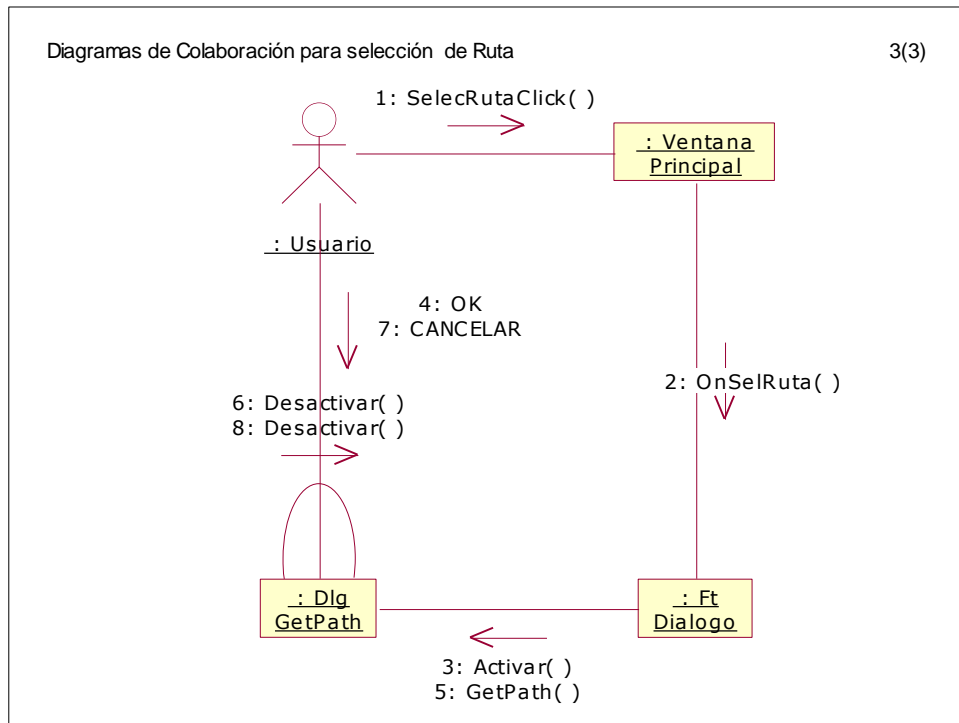


Figura B.32 Diagrama de colaboración Seleccionar Ruta