SERVICIO DE VIDEO POR DEMANDA GESTIONABLE CON SS7

ANEXOS

ANA CRISTINA MOYA ZAMORA

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES

DEPARTAMENTO DE TELEMÁTICA

POPAYÁN

2004

SERVICIO DE VIDEO POR DEMANDA GESTIONABLE CON SS7

ANEXOS

ANA CRISTINA MOYA ZAMORA

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES

DEPARTAMENTO DE TELEMATICA

POPAYÁN

2004

SERVICIO DE VIDEO POR DEMANDA GESTIONABLE CON SS7

ANEXOS

ANA CRISTINA MOYA ZAMORA

Trabajo de grado presentado como requisito para optar al título de Ingeniero en Electrónica y Telecomunicaciones

Director

HECTOR FABIO JARAMILLO ORDOÑEZ.

Ingeniero en Electrónica y Telecomunicaciones

UNIVERSIDAD DEL CAUCA

FACULTAD DE INGENIERIA ELECTRÓNICA Y TELECOMUNICACIONES

DEPARTAMENTO DE TELEMATICA

POPAYÁN

2004

CONTENIDO

ANEXO A. DESCRIPCIÓN DE COMPONENTES

A continuación se hace una breve descripción de los diferentes componentes que se han desarrollado en este proyecto, para facilitar su reutilización.

Esta descripción se basa en el trabajo realizado en el trabajo de grado “Modelo de Creación de Servicios de Telemedicina basado en el concepto de Red Inteligente”. [modelo 2004]

Componentes empresariales:

- Componentes de entidad: EntiServicio, EntiProveedor, EntiAdminPro, EntiAdmin
- Componentes de sesión: Proveedor, Servicio
- Componentes manejados por mensajes: Ninguno.

Componentes Web: ApliAdministradorPro, ApliAdministrador

Componentes clientes: Ninguno.

1. COMPONENTES EMPRESARIALES

1.1. Componentes de Entidad

1.1.1. EntiServicio

Descripción: Contiene los componentes que permiten realizar la comunicación con la base de datos relacionada con la información del proveedor de servicio de VoD.

EJBs:

- Videoteca
- ClienteSer

Archivo jar: EntiServicio.jar

1.1.1.1. Videoteca

Descripción: Componente que permite el acceso a la información relacionada con los diferentes videos que pueden ser accedidos por los usuarios

Tipo: Componente de entidad

Interfaces requeridas: ninguna

Interfaces ofrecidas:

- VideotecaHome
- Videoteca

Interfaz No 1: VideotecaHome

Operaciones:

- *Videoteca create(int id, String nombre, int categoria, int genero, String rutaOrigen1, String rutaEnvio1, String rutaOrigen2, String rutaEnvio2, int duracion) :* A partir de la información recibida, crea un nuevo registro en la base de datos Videoteca y retorna la interfaz Videoteca correspondiente
- *Videoteca create(int id):* A partir del identificador recibido, crea un nuevo registro en la base de datos Videoteca y retorna la interfaz Videoteca correspondiente
- *Videoteca findByPrimaryKey(VideotecaPK videotecaKey):* A partir del VideotecaPK obtiene la interfaz Videoteca correspondiente.

- *Enumeration findByCategoria(int categoria):* A partir de la categoría, retorna las diferentes interfaces Videoteca asociadas a los registros con esta categoría
- *Enumeration findByNombre(String nombre):* A partir del nombre, retorna las diferentes interfaces Videoteca asociadas a los registros con ese nombre
- *Collection findAll():* Retorna todas las interfaces Videoteca asociadas a todos los registros.

Interfaz No 2: Videoteca

Operaciones:

- *int getId():* Obtiene el Id del registro
- *String getNombre():*Obtiene el valor de la columna nombre del registro
- *void setNombre(String nombre):* Fija el valor del nombre en el registro
- *int getCategoria():* Obtiene el valor de la columna categoría del registro
- *void setCategoria(int categoria):* Fija el valor de la categoría en el registro
- *int getGenero():* Obtiene el valor de la columna genero del registro
- *void setGenero(int genero):* Fija el valor del género en el registro
- *String getRutaOrigen1():* Obtiene el valor de la columna RutaOrigen1 del registro
- *void setRutaOrigen1(String rutaOrigen1):* Fija el valor de la RutaOrigen1 del registro
- *String getRutaEnvio1():* Obtiene el valor de la columna RutaEnvio1 del registro
- *void setRutaEnvio1(String rutaEnvio1):* Fija el valor de la RutaEnvio1 del registro
- *String getRutaOrigen2():* Obtiene el valor de la columna RutaOrigen2 del registro
- *void setRutaOrigen2(String rutaOrigen2):* Fija el valor de la RutaOrigen2 del registro
- *String getRutaEnvio2():* Obtiene el valor de la columna RutaEnvio2 del registro
- *void setRutaEnvio2(String rutaEnvio2):* Fija el valor de la columna RutaEnvio2 del registro
- *int getDuracion():* Obtiene el valor de la columna Duración del registro
- *void setDuracion(int duracion):* Fija el valor de la duración del registro

ClienteSer

Descripción: Componente que permite el acceso a la información relacionada con los usuarios que pueden acceder al servicio de Video por Demanda.

Tipo: Componente de Entidad

Interfaces requeridas: Ninguna

Interfaces ofrecidas:

- ClienteSerHome
- ClienteSer

Interfaz No 1: ClienteSerHome

Operaciones:

- *ClienteSer create(int id, String nombre, long telefono, int suscripcion, String nombreUsuario, String contraseña, int categoria, String idioma, int activo, String estado):* A partir de la información recibida, crea un nuevo registro en la base de datos ClienteSer y retorna la interfaz ClienteSer correspondiente.
- *ClienteSer create(int id):* A partir del identificador recibido, crea un nuevo registro en la base de datos ClienteSer y retorna la interfaz ClienteSer correspondiente.
- *ClienteSer findByPrimaryKey(ClienteSerPK primaryKey):* A partir del ClienteSerPK obtiene la interfaz ClienteSer correspondiente

- *Enumeration findByNombre_usuario(String nombre_usuario):* A partir del nombre de usuario, retorna la interfaz ClienteSer asociadas al registro con este nombre de usuario.
- *Collection findAll():* Retorna todas las interfaces ClienteSer asociadas a todos los registros.

Interfaz No 2: ClienteSer

Operaciones:

- *int getId():* Obtiene el Id del registro
- *String getNombre()*: Obtiene el valor de la columna nombre del registro
- *void setNombre(String nombre):* Fija el valor del nombre en el registro
- *long getTelefono():* Obtiene el valor de la columna teléfono del registro
- *void setTelefono(long telefono):* Fija el valor del teléfono en el registro
- *int getSuscripcion():* Obtiene el valor de la columna Suscripción del registro
- *void setSuscripcion(int suscripcion):* Fija el valor de la suscripción en el registro
- *String getNombreUsuario():* Obtiene el valor de la columna nombre de usuario del registro
- *void setNombreUsuario(String nombreUsuario):* Fija el valor del nombre del usuario en el registro
- *String getContraseña():* Obtiene el valor del campo contraseña del registro
- *void setContraseña(String contraseña):* Fija el valor de la contraseña del registro
- *int getCategoria():* Obtiene el valor de la columna categoría del registro
- *void setCategoria(int categoria)*: Fija el valor de la categoría en el registro
- *String getIdioma():* Obtiene el valor de la columna Idioma del registro
- *void setIdioma(String idioma):* Fija el valor del idioma en el registro
- *int getActivo():* Obtiene el valor de la columna activo del registro
- *void setActivo(int activo):* Fija el valor de activo en el registro
- *String getEstado():* Obtiene el valor de la columna estado del registro
- *void setEstado(String estado):* Fija el valor del estado en el registro.

1.1.2. EntiProveedor

EJBs:

- ClientePro

Archivo jar: EntiProveedor.jar

1.1.2.1. ClientePro:

Descripción: Componente que permite el acceso a la información relacionada con los diferentes usuarios en el proveedor de servicio

Tipo: Componente de entidad

Interfaces requeridas: ninguna

Interfaces Ofrecidas:

- ClienteProHome
- ClientePro

Interfaz No 1: ClienteProHome

Operaciones:

- *ClientePro create(int id, String nombre, long telefono, int suscripcion, String nombreUsuario, String contraseña, int capacidad, int activo, String estado):* A partir de la información recibida, crea un nuevo registro en la base de datos ClientePro y retorna la interfaz ClientePro correspondiente.

- *ClientePro create(int id):* A partir del identificador recibido, crea un Nuevo registro en la base de datos ClientePro y retorna la interfaz ClientePro correspondiente
- *ClientePro findByPrimaryKey(ClienteProPK primaryKey):* A partir del ClienteProPK obtiene la interfaz ClientePro correspondiente
- *Enumeration findByNombre_usuario(String nombre_usuario):* A partir del nombre de usuario, retorna la interfaz ClientePro asociada al registro con este nombre de usuario
- *Collection findAll():* Retorna todas las interfaces ClientePro asociadas a todos los registros.

Interfaz No 2: ClientePro

Operaciones:

- *int getId():* Obtiene el Id del registro
- *String getNombre():* Obtiene el valor de la columna nombre del registro
- *void setNombre(String nombre):* Fija el valor del nombre en el registro
- *long getTelefono():* Obtiene el valor de la columna teléfono del registro
- *void setTelefono(long telefono):* Fija el valor del teléfono en el registro
- *int getSuscripcion():* Obtiene el valor de la columna suscripción
- *void setSuscripcion(int suscripcion):* Fija el valor de la suscripción en el registro
- *String getNombreUsuario():* Obtiene el valor de la columna nombre de usuario
- *void setNombreUsuario(String nombreUsuario):* Fija el valor del nombre de usuario en el registro
- *String getContraseña():*Obtiene el valor de la columna contraseña del registro
- *void setContraseña(String contraseña):* Fija el valor de la contraseña en el registro
- *int getCapacidad():*Obtiene el valor de la columna capacidad del registro
- *void setCapacidad(int capacidad):* Fija el valor de la capacidad en el registro
- *int getActivo():*Obtiene el valor de la columna activo del registro
- *void setActivo(int activo):* Fija el valor de activo en el registro

- *String getEstado():*Obtiene el valor de la columna estado del registro
- *void setEstado(String estado):* Fija el valor del estado en el registro

1.1.3. EntiAdminPro

EJBs:

- AdministradorPro

Archivo jar: EntiAdminPro.jar

1.1.3.1. AdministradorPro:

Descripción: Componente que permite el acceso a la información relacionada con los administradores del proveedor de servicio.

Tipo: Componente de entidad

Interfaces requeridas: ninguna

Interfaces Ofrecidas:

- AdministradorProHome
- AdministradorPro

Interfaz No 1: AdministradorProHome

Operaciones:

- *AdministradorPro create(int id, String nombre, String nombreUsuario, String contraseña):* A partir de la información recibida, crea un nuevo registro en la base de datos AdministradorPro y retorna la interfaz AdministradorPro correspondiente.

- *AdministradorPro create(int id) :* A partir de la información recibida, crea un nuevo registro en la base de datos AdministradorPro y retorna la interfaz AdministradorPro correspondiente.
- *AdministradorPro findByPrimaryKey(AdministradorProPK primaryKey):* A partir del AdministradorProPK obtiene la interfaz ClientePro correspondiente
- *Enumeration findByNombreUsuario(String nombre_usuario):* A partir del nombre de usuario, retorna la interfaz AdministradorPro asociada al registro con este nombre de usuario
- *Collection findAll():*Retorna todas las interfaces AdministradorPro asociadas a todos los registros.

Interfaz No 2: AdministradorPro

Operaciones:

- *int getId():* Obtiene el Id del registro
- *String getNombre():* Obtiene el valor de la columna nombre del registro
- *void setNombre(String nombre):* Fija el valor del nombre en el registro
- *String getNombreUsuario():* Obtiene el valor del la columna nombre de usuario del registro
- *void setNombreUsuario(String nombreUsuario):* Fija el valor del nombre de usuario en el registro
- *String getContraseña():* Obtiene el valor de la columna contraseña del registro
- *void setContraseña(String contraseña):* Fija el valor de la contraseña en el registro

1.1.4. EntiAdmin

EJBs

- Administrador

Archivo jar: EntiAdmin.jar

1.1.4.1. Administrador:

Descripción: Componente que permite el acceso a la información relacionada con los administradores del servicio de video por demanda.

Tipo: Componente de entidad

Interfaces requeridas: ninguna.

Interfaces Ofrecidas:

- AdministradorHome
- Administrador

Interfaz No 1: AdministradorHome

Operaciones:

- *Administrador create(int id, String nombre, String nombreUsuario, String contraseña):* A partir de la información recibida, crea un nuevo registro en la base de datos Administrador y retorna la interfaz Administrador correspondiente.
- *Administrador create(int id):* A partir de la información recibida, crea un nuevo registro en la base de datos Administrador y retorna la interfaz Administrador correspondiente.
- *Administrador findByPrimaryKey(AdministradorPK administradorKey):* A partir del AdministradorPK obtiene la interfaz ClientePro correspondiente
- *Enumeration findByNombreUsuario(String nombre_usuario):* A partir del nombre de usuario, retorna la interfaz Administrador asociada al registro con este nombre de usuario

- *Collection findAll():*Retorna todas las interfaces Administrador asociadas a todos los registros.

Interfaz No 2: Administrador

Operaciones:

- *int getId():* Obtiene el Id del registro
- *String getNombre():* Obtiene el valor de la columna nombre del registro
- *void setNombre(String nombre):* Fija el valor del nombre en el registro
- *String getNombreUsuario():* Obtiene el valor de la columna nombre de usuario del registro
- *void setNombreUsuario(String nombreUsuario):* Fija el valor del nombre de usuario en el registro
- *String getContraseña():* Obtiene el valor de la columna contraseña del registro
- *void setContraseña(String contraseña):* Fija el valor de la contraseña en el registro.

1.2. Componentes de Sesión

1.2.1. Proveedor

EJBs:

- IA_l1gw
- namedUA_l1gw
- SubAgt_l1gw
- SF_l1gw
- SSM_l1gw
- PA_VoDSP

Archivo jar: Proveedor.jar

1.2.1.1. IA_l1gw

Descripción: Componente que permite la autenticación del usuario con el proveedor del servicio, permitiendo así una relación confiable entre el usuario y el proveedor. Igualmente, cuando el usuario finaliza la sesión de servicio permite liberar los recursos utilizados.

Tipo: Componente de Sesión Stateless

Interfaces requeridas:

- namedUA_l1gwHome
- namedUA_l1gwRemote
- ClienteProHome
- ClientePro

Interfaces ofrecidas:

- IA_l1gwHome
- IA_l1gwRemote

Interfaz No 1: IA_l1gwHome

Operaciones:

- *IA_l1gwRemote create():* Crea la interfaz IA_l1gwRemote

Interfaz No 2: IA_l1gwRemote

Operaciones:

- *namedUA_l1gwRemote verificar(String nombre_usuario, String contraseña):* Con el nombre y la contraseña suministrada, comprueba que el usuario se encuentre registrado con el proveedor del servicio, a través de las interfaces ClienteProHome

y ClientePro y retorna una referencia a la interfaz *namedUA_l1gwRemote* correspondiente.

- *boolean eliminar(String nombre_usuario):* A partir del nombre de usuario suministrado y con las interfaces ClienteProHome y ClientePro, elimina un registro de la base de datos ClientePro.

1.2.1.2. namedUA_l1gw

Descripción: Componente que representa a un usuario en el dominio proveedor, asegurando una relación confiable entre el usuario y el proveedor. Permite controlar y manejar (crear/ suspender/reiniciar) una sesión de servicio interactuando con el SF para crear un SSM y permite el acceso a la información de configuración de usuario suministrada por el $SubAgt_{L1GW}$. Igualmente, cuando el usuario finaliza la sesión de servicio permite liberar los recursos utilizados.

Tipo: Componente de Sesión StateFull

Interfaces requeridas:

- SubAgt_l1gwHome
- SubAgt_l1gwRemote
- SF_l1gwRemote
- SSM_l1gwRemote

Interfaces Ofrecidas:

- namedUA_l1gwHome
- namedUA_l1gwRemote

Interfaz No 1: namedUA_l1gwHome

Operaciones:

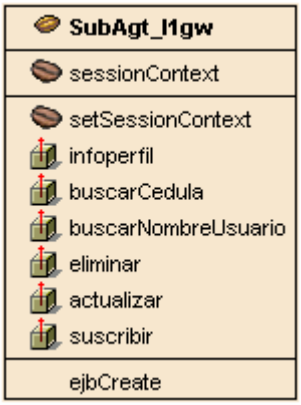
- *namedUA_l1gwRemote create(String nombre_usuario):* Crea la interfaz namedUA_l1gwRemote

Interfaz No 2: namedUA_l1gwRemote

Operaciones:

- *SSM_l1gwRemote iniciar_servicio():* Con el nombre de usuario, con las interfaces SubAgt_l1gwHome y SubAgt_l1gwRemote obtiene la información del perfil específico del usuario, creando una instancia del SF_l1gwRemote para que se inicialice la sesión de servicio, de acuerdo a las características de suscripción y retorna una referencia a la interfaz *SSM_l1gwRemote* entregada por el SF_l1gwRemote.

1.2.1.3. SubAgt_l1gw



Descripción: Permite acceder a la información de suscripción, con el proveedor del servicio, de un usuario particular y cuando el usuario finaliza la sesión de servicio permite liberar los recursos utilizados.

Tipo: Componente de Sesión Stateless

Interfaces Requeridas:

- SF_l1gwHome
- SF_l1gwRemote
- ClientePro
- ClienteProHome

Beans requeridos:

- InfoClientePro

Interfaces Ofrecidas:

- SubAgt_l1gwHome
- SubAgt_l1gwRemote

Interfaz No 1: SubAgt_l1gwHome

Operaciones:

- *SubAgt_l1gwRemote create():*Crea la Interfaz SubAgt_l1gwRemote

Interfaz No 2: SubAgt_l1gwRemote

Operaciones:

- *SF_l1gwRemote infoperfil (String nombre_usuario):* Retorna una instancia de la interfaz *SF_l1gwRemote* inicializada con las características de suscripción relacionadas al usuario.
- *InfoClientePro buscarCedula(int id):* A partir de la cédula de un usuario y con las interfaces ClienteProHome y ClientePro, retorna un objeto InfoClientePro con la información del usuario en el dominio proveedor.
- *InfoClientePro buscarNombreUsuario(String nombre_usuario):* A partir del un nombre de usuario y con las interfaces ClienteProHome y ClientePro, retorna un objeto InfoClientePro con la información del usuario en el dominio proveedor.
- *boolean eliminar(String nombre_usuario):* A partir del nombre de usuario y con las interfaces ClienteProHome y ClientePro elimina la información asociada al usuario en el dominio proveedor.
- *boolean actualizar(InfoClientePro info):* A partir de la información contenida en el objeto InfoClientePro y con las interfaces ClienteProHome y ClientePro actualiza la información asociada a este en el dominio proveedor
- *String suscribir(int id, String nombre, long telefono, int suscripcion, String nombreUsuario, String contraseña, int capacidad, int activo, String estado):* A partir de la información suministrada y con las interfaces ClienteProHome y ClientePro, suscribe un nuevo usuario en el dominio proveedor.

1.2.1.4. SF_l1gw

Descripción: Componente que instancia una sesión de proveedor de servicio creando los objetos de sesión SSM_{L1GW} necesarios para la prestación del servicio de video por demanda, igualmente permite la finalización de esta sesión y liberar los recursos.

Tipo: Componente de Sesión Statefull

Interfaces requeridas:

- SSM_l1gwHome
- SSM_l1gwRemote

Beans requeridos:

- infoClientePro

Interfaces ofrecidas:

- SF_l1gwHome
- SF_l1gwRemote

Interfaz No 1: SF_l1gwHome

Operaciones:

- *SF_l1gwRemote create(InfoClientePro infoClientePro):* Crea la interfaz SF_l1gwRemote, inicializándola con el objeto InfoClientePro.

Interfaz No 2: SF_l1gwRemote

Operaciones:

- *SSM_l1gwRemote inicializarSSM():* Crea la interfaz de sesión *SSM_l1gwRemote* necesaria para la prestación del servicio y la retorna.

1.2.1.5. SSM_l1gw

Descripción: Durante la sesión del servicio envía las solicitudes de control realizadas por el usuario a las interfaces VoDSPRemote, permitiendo seleccionar un video de la lista de contenidos y controlar la reproducción del video seleccionado.

Tipo: Componente de Sesión Statefull

Interfaces requeridas:

- PA_VoDSPHome
- PA_VoDSPRemote
- SSM_VoDSPRemote

Interfaces ofrecidas:

- SSM_l1gwHome
- SSM_l1gwRemote

Interfaz No 1: SSM_l1gwHome

Operaciones:

- *SSM_l1gwRemote create(InfoClientePro infoClientePro):* Crea la interfaz SSM_l1gwRemote

Interfaz No 2: SSM_l1gwRemote

Operaciones:

- *Vector obtener_lista_contenidos():* Solicita al SSMVoD_SP la lista de contenidos a los que el usuario tiene acceso.
- *void seleccion(String video):* Envía la información del video seleccionado al SSMVoD SP, para que se inicie su reproducción

1.2.1.6. PA_VoDSP

Descripción: Componente que permite la autenticación del usuario con el proveedor del servicio de video por demanda, interactuando con un agente inicial (interfaz IA_VoDSPRemote) y siendo referenciado a un agente de Usuario (interfaz namedUA_VoDSPRemote), permitiendo así una relación confiable entre estos. Igualmente permite la finalización de la sesión establecida entre el proveedor del servicio y el proveedor del servicio de video por demanda.

Tipo: Componente de Sesión Stateless

Interfaces requeridas:

- IA_VoDSPHome
- IA_VoDSPRemote
- SSM_VoDSPRemote

Interfaces ofrecidas:

- PA_VoDSPHome
- PA_VoDSPRemote

Interfaz No 1: PA_VoDSPHome

Operaciones

- *PA_VoDSPRemote create():* Crea la interfaz PA_VoDSPRemote

Interfaz No 2: PA_VoDSPRemote

Operaciones:

- *SSM_VoDSPRemote iniciar_dom_servicio(String nombre_usuario, String contraseña):* inicializa el servicio, para esto se comunica con el proveedor del servicio de video por demanda a través de las interfaces IA_VoDSPHome y

IA_VoDSPRemote para realiza la autenticación y obtener una referencia a la interfaz *SSM_VoDSPRemot*, para iniciar la prestación del servicio.

1.2.3. Servicio

EJBs

- IA_VoDSP
- namedUA_VoDSP
- SF_VoDSP
- SSM_VoDSP
- SubAgt_VoDSP
- GestorVideos

Archivo jar: Servicio.jar

1.2.3.1. IA_VoDSP

Descripción: Componente que permite la autenticación del usuario con el proveedor del servicio de video por demanda, permitiendo así una relación confiable entre el proveedor del servicio y el proveedor de servicio de video por demanda. Igualmente, cuando el usuario finaliza la sesión de servicio permite liberar los recursos utilizados

Tipo: Componente de Sesión Stateless.

Interfaces requeridas:

- namedUA_VoDSPHome
- namedUA_VoDSPRemote
- ClienteSerHome
- ClienteSer

Interfaces ofrecidas:

- IA_VoDSPHome
- IA_VoDSPRemote

Interfaz No 1: IA_VoDSPHome

Operaciones:

- *IA_VoDSPRemote create()*: Crea la interfaz IA_VoDSPRemote

Interfaz No 2: IA_VoDSPRemote

Operaciones:

- *namedUA_VoDSPRemote verificar(String nombre_usuario, String contraseña):* Con el nombre y la contraseña suministrada, comprueba que el usuario se encuentre registrado con el proveedor del servicio de video por demanda, a través de las interfaces ClienteSerHome y ClienteSer y retorna una referencia a la interfaz *namedUA_VoDSPRemote* correspondiente.
- *boolean eliminar(String nombre_usuario):* A partir del nombre de usuario suministrado y con las interfaces ClienteSerHome y ClienteSer, elimina un registro de la base de datos ClienteSer.

1.2.3.2. namedUA_VoDSP

Descripción:

Tipo: Componente de Sesión Statefull

Interfaces requeridas:

- SubAgt_VoDSPHome

- SubAgt_VoDSPRemote
- SF_VoDSPRemote
- SSM_VoDSPRemote

Interfaces ofrecidas:

- namedUA_VoDSPHome
- namedUA_VoDSPRemote

Interfaz No 1: namedUA_VoDSPHome

Operaciones:

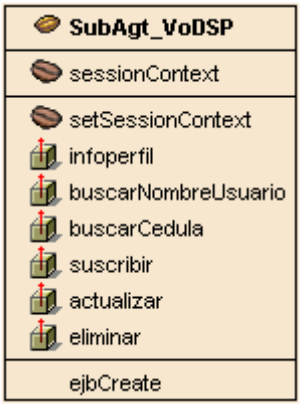
- *namedUA_VoDSPRemote create(String nombre_usuario):* Crea la interfaz namedUA_VoDSPRemote

Interfaz No2: namedUA_VoDSPRemote

Operaciones:

- *SSM_VoDSPRemote iniciar_servicio():*Con el nombre de usuario, con las interfaces SubAgt_VoDSPHome y SubAgt_VoDSPRemote obtiene la información del perfil específico del usuario, creando una instancia del SF_VoDSPRemote para que se inicialice la sesión de servicio, de acuerdo a las características de suscripción y retorna una referencia a la interfaz *SSM_VoDSPRemote* entregada por el SF_VoDSPRemote.

1.2.3.3. SubAgt_VoDSP



Descripción: Permite acceder a la información de suscripción, con el proveedor de servicio de video por demanda, de un usuario particular y cuando el usuario finaliza la sesión de servicio permite liberar los recursos utilizados.

Tipo: Componente de Sesión Stateless

Interfaces requeridas:

- ClienteSerHome
- ClienteSer

Beans requeridos:

- InfoClienteSer

Interfaces ofrecidas

- SubAgt_VoDSPHome
- SubAgt_VoDSPRemote

Interfaz No 1: SubAgt_VoDSPHome

Operaciones:

- *SubAgt_VoDSPRemote create():* Crea la interfaz SubAgt_VoDSPRemote

Interfaz No 2: SubAgt_VoDSPRemote

Operaciones:

- *SF_VoDSPRemote infoperfil(String nombre_usuario):* Retorna una instancia de la interfaz *SF_VoDSPRemote* inicializada con las características de suscripción relacionadas al usuario.
- *InfoClienteSer buscarNombreUsuario(String nombre_usuario):* A partir del nombre de usuario y con las interfaces ClienteSerHome y ClienteSer, retorna un objeto InfoClienteSer con la información del usuario en el dominio del servicio de video por demanda.
- *InfoClienteSer buscarCedula(int id):* A partir de la cédula de un usuario y con las interfaces ClienteSerHome y ClienteSer, retorna un objeto InfoClienteSer con la información del usuario en el dominio del servicio de video por demanda.

- *String suscribir(int id, String nombre, long telefono, int suscripcion, String nombreUsuario, String contraseña, int categoria, String idioma, int activo, String estado):* A partir de la información suministrada y con las interfaces ClienteSerHome y ClienteSer, suscribe un nuevo usuario en el dominio del servicio de video por demanda.
- *boolean actualizar(InfoClienteSer info):* A partir de la información contenida en el objeto InfoClienteSer y con las interfaces ClienteSerHome y ClienteSer actualiza la información asociada a este en el dominio del servicio de video por demanda.
- *boolean eliminar(String nombre_usuario):* A partir del nombre de usuario y con las interfaces ClienteSerHome y ClienteSer elimina la información asociada al usuario en el dominio del servicio de video por demanda.

1.2.3.4. SF_VoDSP

Descripción: Componente que instancia una sesión de servicio de video por demanda creando los objetos de sesión $SSM_{VoD\ SP}$ necesarios para la prestación del servicio de video por demanda, igualmente permite la finalización de esta sesión y liberar los recursos.

Tipo: Componente de Sesión Statefull

Interfaces requeridas:

- SSM_VoDSPHome
- SSM_VoDSPHome

Clases requeridas:

- InfoClienteSer

Interfaces ofrecidas:

- SF_VoDSPHome
- SF_VoDSPRemote

Interfaz No 1: SF_VoDSPHome

Operaciones

- *SF_VoDSPRemote create(InfoClienteSer infoClienteSer):* Crea la interfáz SF_VoDSPRemote, iniciándola con el objeto InfoClienteSer.

Interfaz No 2: SF_VoDSPRemote

Operaciones:

- *SSM_VoDSPRemote inicializarSSM():* Crea la interfaz de sesión SSM_VoDSPRemote necesaria para la prestación del servicio y la retorna.

1.2.3.5. SSM_VoDSP

Descripción: Durante la sesión del servicio envía las solicitudes de control realizadas por el usuario al Servidor de video, permitiendo seleccionar un video de la lista de contenidos y controlar la reproducción del video seleccionado.

Tipo: Componente de Sesión Statefull

Interfaces requeridas:

Interfaces ofrecidas:

- SSM_VoDSPHome
- SSM_VoDSPRemote

Interfaz No 1: SSM_VoDSPHome

Operaciones:

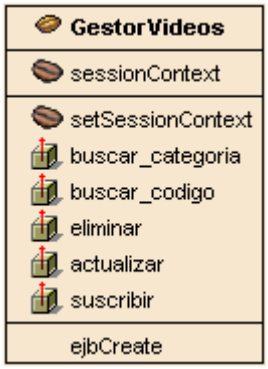
- *SSM_VoDSPRemote create(InfoClienteSer infoClienteSer):* Crea la interfaz SSM_VoDSPRemote, inicialiándola con el objeto InfoClienteSer

Interfaz No 2: SSM_VoDSPRemote

Operaciones:

- *Vector obtener_lista_contenidos():* Retorna la lista de contenidos, relacionada en el objeto InfoClienteSer, a los que el usuario tiene acceso.
- *void seleccion(String video):* Envía la información del video seleccionado al servidor de video, para que se inicie su reproducción.

1.2.3.6. GestorVideos



Descripción: Permite acceder a la información de los videos almacenados en la base de datos videoteca.

Tipo: Componente de Sesión Stateless

Interfaces requeridas:

- VideotecaHome
- Videoteca

Interfaces ofrecidas:

- GestorVideosHome
- GestorVideosRemote

Interfaz No 1: GestorVideosHome

Operaciones:

- *GestorVideosRemote create() :* Crea la interfaz GestorVideosRemote

Interfaz No 2: GestorVideosRemote

Operaciones:

- *Vector buscar_categoria(int categoria):* A partir de la categoría del video y con las interfaces VideotecaHome y Videoteca, retorna un vector con la información relacionada al código y el nombre de las diferentes películas asociadas a la categoría.
- *InformacionVideo buscar_codigo(int codigo):* A partir del código del video y con las interfaces VideotecaHome y Videoteca, retorna un objeto InformacionVideo, con la información almacenada en el registro identificado con ese código en la base de datos Videoteca
- *boolean eliminar(int id) :* A partir del código de la película y con las interfaces VideotecaHome y Videoteca, elimina el registro de la base de datos Videoteca con ese código.
- *boolean actualizar(InformacionVideo info):* A partir de la información contenida en el objeto InformacionVideo y con las interfaces VideotecaHome y Videoteca actualiza la información asociada a este en la videoteca.
- *String suscribir(int id, String nombre, int categoria, int genero, String ruta_origen1, String ruta_envio1, String ruta_origen2, String ruta_envio2, int duracion)*: A partir de la información suministrada y con las interfaces VideotecaHome y Videoteca, ingresa la información relacionada con un nuevo video a la videoteca.

2. COMPONENTES WEB

2.1. ApliAdministradorProveedor

Descripción: Componente de presentación que ofrece las páginas a las que puede acceder el administrador del proveedor del servicio para realizar la gestión de los usuarios.

Tipo: Componente Web

Interfaces requeridas:

- SubAgt_l1gwHome
- SubAgt_l1gwRemote
- AdministradorProHome
- AdministradorPro

Interfaces ofrecidas: ninguna.

Páginas html:

- *Validar.html:* Página inicial, permite el acceso a las diferentes interfaces del administrador del proveedor del servicio, para eso este debe ingresar un nombre de usuario y una contraseña correcta.

Páginas jsp:

- *Buscar.jsp:* Página que permite realizar la búsqueda de un usuario registrado con el proveedor del servicio, ya sea a partir del número de cédula o el nombre de usuario.
- *Editar.jsp:* Página que permite actualizar el registro de un usuario, permitiendo ingresar los nuevos valores de los diferentes campos que se desean actualizar.
- *Inicio.jsp:* Página que presenta el menú de las diferentes opciones que tiene el administrador del proveedor del servicio.
- *Mensajes.jsp:* Página que presenta la información que retorna el sistema
- *Suscribir.jsp:* Página que permite ingresar la información referente a un nuevo usuario.

Servlets:

- Buscar.java: Realiza la lógica de la búsqueda de un usuario registrado con el proveedor del servicio, ya sea a partir del número de cédula o el nombre de usuario, a través de las interfaces SubAgt_l1gwHome y SubAgt_l1gwRemote, desplegando la información obtenida en la página Editar.jsp
- Editar.java: Realiza la lógica de editar el registro de un usuario con el proveedor del servicio, a través de las interfaces SubAgt_l1gwHome y SubAgt_l1gwRemote, actualizando la información del usuario.

- Suscribir.java: Realiza la lógica de suscribir a un usuario con el proveedor, a través de las interfaces SubAgt_l1gwHome y SubAgt_l1gwRemote, almacenando la información del nuevo usuario.
- Validar.java: realiza la lógica de verificar que el nombre de usuario y la contraseña de administrador es correcta, a través de las interfaces AdministradorProHome y AdministradorPro, si la autenticación es correcta, inicia la página Inicio.jsp

Applets: ninguna

JavaBeans: ninguno

Archivo war: ApliAdministradorProveedor.war

2.2. ApliAdministrador

Descripción: Componente de presentación que ofrece las páginas a las que puede acceder el administrador del servicio de video por demanda para realizar la gestión de los usuarios y la de los diferentes videos a los cuales tiene acceso.

Tipo: Componente Web.

Interfaces requeridas:

- SubAgt_VoDSPHome
- SubAgt_VoDSPRemote
- AdministradorHome
- Administrador
- GestorVideosHome
- GestorVideosRemote

Interfaces ofrecidas: ninguna

Páginas html:

- *Validar.html:* Página inicial, permite el acceso a las diferentes interfaces del administrador del servicio de video por demanda, para esto este debe ingresar un nombre y una contraseña correcta.

Páginas jsp:

- *Buscar.jsp:* Página que permite realizar la búsqueda de un usuario registrado con el servicio de video por demanda, ya sea a partir del número de cédula o el nombre de usuario
- *BuscarVideo.jsp:* Página que permite realizar la búsqueda de un video de la videoteca del servicio, ya sea a partir del código de este o la categoría en la que se encuentra registrado
- *Editar.jsp:* Página que permite actualizar el registro de un usuario, permitiendo ingresar los nuevos valores de los diferentes campos que se desean actualizar.
- *EditarVideo.jsp:* Página que permite actualizar la información de un video de la videoteca, permitiendo ingresar los nuevos valores de los diferentes campos que se desean actualizar
- *Inicio.jsp:* Página que presenta el menú de las diferentes opciones que tiene el administrador del servicio de video por demanda.
- *ListarVideos.jsp:* Página que despliega la lista de videos asociados a una categoría específica, para seleccionar el que se desee obtener más información.
- *Mensajes.jsp:* Página que presenta la información que retorna el sistema
- *NuevoVideo.jsp:* Página que permite ingresar la información referente a un nuevo video disponible a los usuarios.
- *Opciones.jsp:* Página que presenta el menú de las diferentes opciones que se tienen para la gestión de los videos de la videoteca del servicio.
- *Suscribir.jsp:* Página que permite ingresar la información referente a un nuevo usuario.

Servlets:

- Buscar.java: Realiza la lógica de la búsqueda de un usuario registrado con el servicio de video por demanda, ya sea a partir del número de cédula o el nombre de usuario, a través de las interfaces SubAgt_VoDSPHome y SubAgt_VoDSPRemote, desplegando la información obtenida en la página Editar.jsp

- BuscarVideo.java: realiza la lógica de la búsqueda de un video de la videoteca del servicio, ya sea a partir del código de este o la categoría en la que se encuentra registrado, a través de las interfaces GestorVideosHome y GestorVideosRemote
- Editar.java: Realiza la lógica de editar el registro de un usuario con el servicio de video por demanda, a través de las interfaces SubAgt_VoDSPHome y SubAgt_VoDSPRemote, actualizando la información del usuario.
- EditarVideo.java: Realiza la lógica de editar la información relacionada a un video de la videoteca, a través de las interfaces GestorVideosHome y GestorVideosRemote, actualizando la información del video.
- NuevoVideo.java: Realiza la lógica de agregar un video en la videoteca del servicio, a través de las interfaces GestorVideosHome y GestorVideosRemote y de obtener y almacenar los flujos de videos que van a ser enviados a los usuarios
- Selección.java: A partir del video seleccionado por el usuario en la página ListarVideos.jsp, despliega la información del video en una página EditarVideo.jsp
- Suscribir.java: Realiza la lógica de suscribir a un usuario con el servicio de video por demanda, a través de las interfaces SubAgt_VoDSPHome y SubAgt_VoDSPRemote, almacenando la información del nuevo usuario.
- Validar.java: realiza la lógica de verificar que el nombre de usuario y la contraseña de administrador es correcta, a través de las interfaces AdministradorHome y Administrador si la autenticación es correcta, inicia la página Inicio.jsp

REFERENCIAS

[modelo 2004] HERNÁNDEZ B., CLAUDIA M. TOBAR A. CARLOS H. “MODELO DE CREACIÓN DE SERVICIOS DE TELEMEDICINA”, 2004

ANEXO B. INFORMACIÓN CAPI 2.0

1. MENSAJES CAPI 2.0

Recibir Información sobre CAPI

CAPI_GET_PROFILE: Obtener las capacidades de la implementación del CAPI.

CAPI_GET_MANUFACTURER: Obtener la identificación de fábrica.

CAPI_GET_VERSION: Obtener la versión del CAPI.

CAPI_GET_SERIAL_NUMBRE: Obtener el número del serial.

CAPI_INTALLED: Verifica si el CAPI esta instalado.

CAPI_MANUFACTURER: Funciones específicas de fábrica.

Registrar y terminar el uso del CAPI

CAPI_REGISTER: Registrar una aplicación con el CAPI, definiendo el número máximo de conexiones lógicas, el tamaño del buffer de mensajes, el número de buffers de datos y el número de buffers requeridos por la aplicación.

CAPI_RELEASE: Libera una aplicación que ha sido previamente registrada en el CAPI.

Operaciones con pilas de mensajes.

CAPI_PUT_MESSAGE: Enviar un mensaje al CAPI.

CAPI_GET_MESSAGE: Obtener un mensaje del CAPI (Retorna OK: Dirección del mensaje, ó NO: Indicación de error).

CAPI_SET_SIGNAL* : Habilita un mecanismo que indica que hay un mensaje del CAPI.

CAPI_WAIT_FOR_SIGNAL*: Espera a que un nuevo mensaje esté disponible.

---

* Esta operación solo se encuentra disponible para ciertos sistemas operatives * Operations marked with an asterisk (*) are only available in implementations for certain operating systems.

Protocolo de señalización

CONNECT: Inicia una conexión física.

CONNECT_ACTIVE: Indica la activación de una conexión física de un canal B (PLCI) (IND y RESP).

DISCONNECT: Finaliza una conexión física.

ALERT: Encía una ALERTA (REQ y CONF).

INFO: Envío de información de señalización.

Conexiones lógicas.

CONNECT_B3: Inicia una conexión lógica (PLCI).

CONNECT_B3_ACTIVE: Indica la activación de una conexión lógica de un canal B (NCCI) (IND y RESP).

CONNECT_B3_T90_ACTIVE: Switch entre T 70 NL a T90 NL (IND y RESP).

DISCONNECT_B3: Finaliza una conexión lógica.

DATA_B3: Envío de datos sobre una conexión lógica identificada por NCCI.

RESERT_B3: Inicia el reinicio de una conexión lógica.

Mensajes de Administración

LISTEN: Activa una llamada e indicaciones de información (REQ y CONF).

FACILITY: Solicita facilidades adicionales.

SELECT_B_PROTOCOL: Selecciona el protocolo utilizado por una conexión lógica (REQ y CONF).

MANUFACTURERE: Operación de especificación de fábrica.

Datos:

Controller: El propósito del parámetro Controller es direccional una unidad hardware que provee a la aplicación con acceso a ISDN. Un controlador puede soportar cero, una o muchas conexiones físicas y lógicas.

PLCI: (Physical Link Connection Identifier (identificador de un enlace de conexión física)). El propósito del PLCI es identificar una conexión física entre dos terminales.

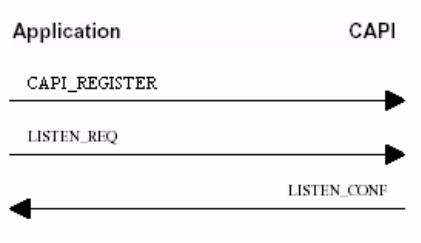
NCCI: (Network Control Connection Identifier (identificador de conexión de control de red)) El propósito del NCCI es identificar una conexión lógica.

NCPI: El propósito del parámetro NCPI es proveer información adicional específica del protocolo y por lo tanto es dependiente de este.

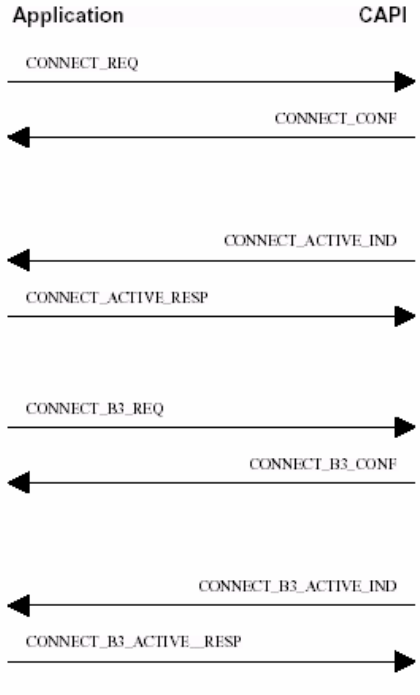
FACILITY: Handset, DTMF, V. 42bis, Servicios suplementarios, wakeup, Interconexión de línea.

2. DIAGRAMAS DE FLUJOS DE MENSAJES.

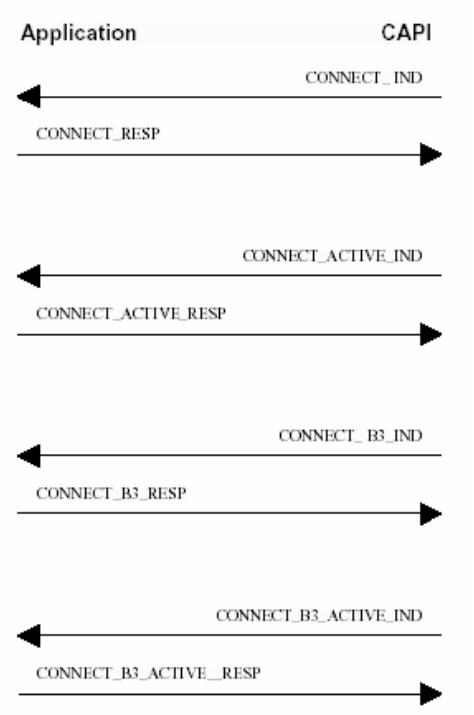
Iniciar: Este proceso se realiza antes del envío de cualquier mensaje al módem.



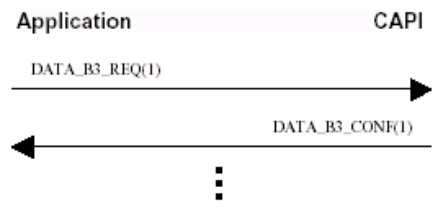
Hacer una llamada: Este es el procedimiento que debe realizar un usuario para comunicarse con otro que se encuentre corriendo una aplicación de CAPI.



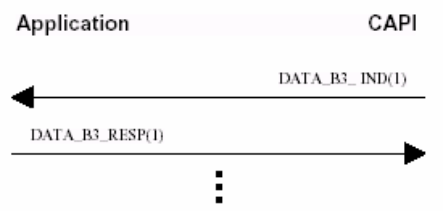
Atender una llamada: Este es el procedimiento que debe realizar un usuario para aceptar la comunicación de otro que se encuentre corriendo una aplicación de CAPI.



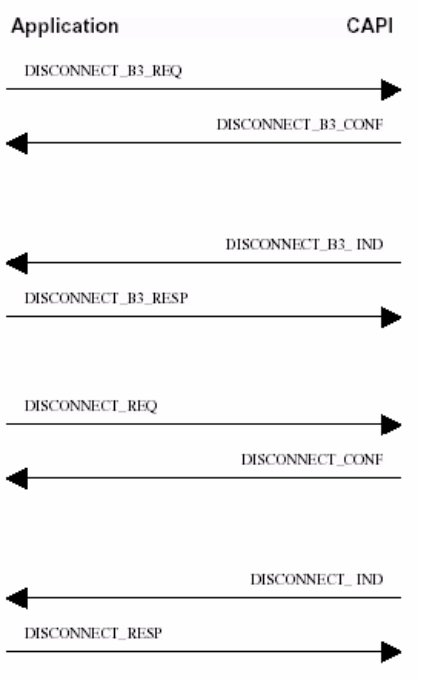
Transmitir Datos: Este es el procedimiento que debe realizar un usuario para enviar datos a otro, luego del establecimiento de una comunicación.



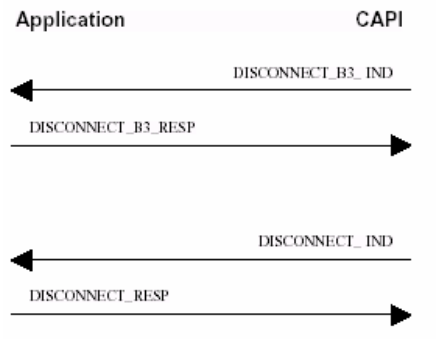
Recibir Datos: Este es el procedimiento que debe realizar un usuario para recibir mensajes enviados por otro usuario luego del establecimiento de la comunicación.



Finalizar una llamada: Este es el procedimiento que debe realizar un usuario para finalizar la comunicación, previamente establecida, con otro usuario.



Aceptar la finalización de una llamada: Este es el procedimiento que debe realizar un usuario para aceptar la finalización de una llamada, solicitada por otro usuario con el que previamente ha iniciado una comunicación.



3. JCAPI

A continuación se describen los principales mensajes del JCAPI.

CAPI_REGISTER

*public int **register**(int bufsize, int maxcon, int maxblocks, int maxlen) throws CapiException:* Equivalente al CAPI_REGISTER.

Parámetros:

bufsize: tamaño del buffer (típicamente 1024+(1024*maxcon)).

maxcon:el número máximo de conexiones lógicas.

maxblocks:el número máximo de bloques de datos recibidos.

maxlen: máximo tamaño de bloques de datos para ser transmitidos y recibidos.

Retorna:

Número de identificación único para esta aplicación.

CREATEMESSAGE

public *CapiMessage* **createMessage**(int appID, int type, int number) throws *CapiException*: Construye un nuevo objeto CapiMessage.

Parámetros:

appID: número de identificación de la aplicación asignado por register().

type: Tipo de mensaje, el cual corresponde con los campos de mensajes “command” y “subcommand”.

number: Número de identificación de este mensaje.

Retorna:

Un nuevo objeto CapiMessage.

CAPI_PUT_MESSAGE

public void ***putMessage****(CapiMessage msg) throws CapiException:* Equivalente al CAPI_PUT_MESSAGE.

Parametros:

appID: número de identificación asignado por register().

msg: El mensaje a ser enviado al CAPI.

CAPI_GET_MESSAGE

public *CapiMessage* **getMessage**(int appID) throws *CapiException*: Equivalente al CAPI_GET_MESSAGE.

Parameters:

appID: Número de identificación de la aplicación asignado por register().

Returns:

El CapiMessage entregado por el CAPI.

ADDLISTENER

public void ***addListener****(int appID, CapiListener listener):* Adiciona una referencia dada a una lista de CapiListeners que serán notificados por el CapiEvents para el identificador de la aplicación dado.

Parámetros:

appID: Número de identificación de la aplicación asignado por register().

CAPI_RELEASE

public void ***release****(int appID) throws CapiException:* Equivalente al CAPI_RELEASE.

Parámetros:

appID: Número de identificación de la aplicación asignado por register().

REMOVELISTENER

public void removeListener(int appID, CapiListener listener).

Remueve un CapiListener previemente añadido con un addListener().

Parámetros:

appID: Número de identificación de la aplicación asignado por register().

INSTALLED

public boolean ***installed****()throws CapiException:* Equivalente al CAPI_INSTALLED. Este método siempre retorna un valor true o envía una excepción si el CAPI 2.0 no está instalado correctamente.

Retorna:

True si el CAPI 2.0 esta instalado y listo.

GETIMPLEMENTATIONINFO

public java.lang.String ***getImplementationInfo****():* Retorna información acerca de la implementación utilizada para esta interfaz.

Retorna:

La descripción sobre la actual implementación.

GETMANUFACTURER

public java.lang.String ***getManufacturer****(int controller) throws CapiException:* Equivalente al CAPI_MANUFACTURER.

Parámetros:

controller : Número del controlador solicitado.

Retorna:

La identificación de fabrica del CAPI 2.0 (y probablemente del dispositivo).

GETVERSION

public int[] ***getVersion****() throws CapiException:* Equivalente al CAPI_GET_VERSION.

Retorna:

Un arreglo de 4 enteros: mayor versión del CAPI (debería ser 2), menor versión del CAPI (debería ser 0), mayor y menor versión de fábrica.

GETSERIALNUMBER

public java.lang.String ***getSerialNumber****(int controller) throws CapiException:* Equivalente al CAPI_GET_SERIAL_NUMBER.

Parámetros:

controller - El número del controlador solicitado.

Retorna:

El número serial codificado como un String.

GETNUMBEROFCONTROLLERS

public int ***getNumberOfControllers****() throws CapiException:* Equivelente al CAPI_GET_PROFILE con parámetro 0.

Retorna:

El número de controladores instalados.

GETPROFILE

public byte[] ***getProfile****(int controller) throws CapiException:* Equivalente al CAPI_GET_PROFILE con parámetro diferente de 0.

Parámetros:

controller – El número de el controlador; un valor de 0 puede causar una excepción.

Retorna:

Un arreglo de bytes que contienen el perfil del controlador solicitado.

CAPILISTENER

capiEventRaised

public void ***capiEventRaised****():* Es llamada por la implementación del CAPI para un evento CAPI asignado a este listener (por ejemplo una señal de mensaje para la aplicación) .

4. MENSAJES CAPI 2.0

CAPI_INSTALLED

Las aplicaciones llaman al CAPI_INSTALLED para determina si el controlador ISDN y todos los drivers necesarios se encuentran instalados.

CAPI_GET_MANUFACTURER

Las aplicaciones llaman al CAPI_GET_MANUFACTURER para recibir información acerca del fabricante del controlador ISDN específica. CAPI copia esta información en un buffer de 64 bytes pasado por la aplicación llamante.

CAPI_GET_VERSION

Las aplicaciones llaman al CAPI_GET_VERSION para recibir información sobre la versión del controlador ISDN específico.

En número de la mayor y menor versión es retornado por el CAPI y el fabricante de la implementación específica.

CAPI_GET_SERIAL_NUMBER

Las aplicaciones llaman al CAPI_GET_SERIAL_NUMBRE para recibir el número del serial del contorlador ISDN específico.

CAPI copia esta información a un buffer de 8 bytes para pasarlo por la aplicación llamante.

CAPI_GET_PROFILE

Las aplicaciones llaman al CAPI_GET_PROFILE para recibir información sobre las capacidades del CAPI. CAPI copia la información acerca de las características de implementación, el número total de controladores y los protocolos soportados por el controlador específico a un buffer de 64 bytes para ser pasado por la aplicación de llamante.

La aplicación debe ignorar bits desconocidos. CAPI fija cada campo reservado a cero. CAPI_GET_PROFILE llena el buffer con la siguiente estructura:

Type	Description
2 bytes	Número de controladores instalados, byte menos significativo primero.
2 bytes	Número de canales B soportados, byte menos significativo primero.
4 bytes	Opciones Globales (bit field): [0]: Soporta controlador interno [1]: Soporta equipo externo [2]: Soporta Microteléfono (El equipo externo debe ser fijado) [3]: Soporta DTMF [4]: Servicios Suplementarios [5]: Soporta canal de asignación (líneas leased) [6]: Soporta operación de parámetros de canal B

	[7]: Soporta línea interconectada [8]...[31]: reservados
4 bytes	Soporte de protocolo B1 (bit field): [0]: 64 kbit/s con HDLC framing, siempre fijo. [1]: 64 kbit/s operación bit-transparente con byte framming desde la red [2]: Operación asincrónica V.110 con start/stop byte framing [3]: Operación sincrónica V.110 con HDLC framing [4]: Módem T.30 para fax Grupo 3 [5]: 64 kbit/s invertico con HDLC framing. [6]: 56 kbit/s operación bit-transparente con byte framing desde la red [7]: Módem con todas las negociaciones [8]: Módem de operación asincrónica con byte start/stop framing [9]: Módem de operación sincrónica con HDLC framing [10]...[31]: reservados
4 bytes	Soporte de protocolo B2 (bit field): [0]: ISO 7776 (X.75 SLP), siempre fijo [1]: Transparente [2]: SDLC [3]: LAPD de acuerdo con Q.921 para canal D X.25 (SAPI 16) [4]: T.30 para fax grupo 3 [5]: Protocolo Punto a punto (PPP) [6]: Transparente (ignora errores de framing del protocolo B1) [7]: Módem con corrección de errores y compresión (V.42 bis o MNP5) [8]: ISO 7776 (X.75 SLP) modificado para soportar compresión V.42 bis [9]: Modo asincrónico V.120 [10]: Modo asincrónico V.120 soportando V.42 bis [11]: modo V.120 bit-transparente [12]: LAPD de acuerdo con Q.921 incluyendo selección libre de SAPI [13]...[31]: reservados
4 bytes	Soporte de protocolo B3 (bit field): [0]: Transparente, siempre fijo. [1]: T.90NL con compatibilidad con T.70NL de acuerdo con T.90 [2]: ISO 8208 (X.25 DTE-DTE) [3]: X.25 DCE [4]: T.30 para fax grupo 3 [5]: T.30 para fax grupo 3 con extensiones [6]: reservado [7]: Módem [8]...[31]: reservado
24 bytes	Reservado para uso del CAPI
20 bytes	Información específica de fabrica.

CAPI_REGISTER

Aplicaciones utilizan el CAPI_REGISTER para registrar su presencia con el CAPI, los parámetros de registro especifican el número máximo de conexiones lógicas ISDN, el tamaño del buffer de mensajes, el número de buffers de datos y el número de buffers requeridos para la aplicación. El tamaño del buffer de mensajes es normalmente calculado de acuerdo con la siguiente fórmula:

Tamaño del buffer de mensajes: 1024+(1024*número de conexiones lógicas ISDN))

Si el registro es correcto, el CAPI asigna y retorna al llamante un número de identificación único para la aplicación. Este identificador es utilizado luego para el llamado a funciones del CAPI y para la definición de mensajes del CAPI.

Algunos sistemas operativos requieren que las aplicaciones pasen memoria del buffer u otra información adicional al CAPI.

CAPI_PUT_MESSAGE

Las aplicaciones llaman al CAPI_PUT_MESSAGE para transferir un solo mensaje al CAPI, cuando el llamado a la función retorna, el área de memoria del mensaje puede ser reutilizado por la aplicación.

CAPI_GET_MESSAGE

Las aplicaciones llaman al CAPI_GET_MESSAGE para recibir un solo mensaje del CAPI si un mensaje esta disponible, la dirección del mensaje es entregado a la aplicación. Si no hay mensajes el CAPI_GET_MESSAGE retorna un indicador de error. El contenido del bloque de mensaje retornado por el CAPI_GET_MESSAGE es valido hasta que la misma aplicación llama al CAPI_GET_MESSAGE nuevamente. La aplicación que procesa los mensajes asincrónicamente o necesita preservar el mensaje luego del siguiente llamado al CAPI_GET_MESSAGE debe hacer una copia local del mensaje

LISTEN_REQ

Es utilizado para activar la señalización de eventos entrantes desde el CAPI a la aplicación.

Info mask: Define cuales eventos del protocolo de señalización deben ser indicados a la aplicación. Estos eventos son normalmente asociados con conexiones físicas.

CIP mask: define los criterios de selección basados en las capacidades portadores y capacidades de alto niveles, para especificar cuales llamadas entrantes estarán señaladas para una aplicación.

Más de una aplicación puede escuchar los mismos valores CIP. Cada aplicación escuchando a un valor correspondiente es informada sobre llamadas entrantes. Si más de una aplicación desea aceptar la llamada, la primera CONNECT_RESP recibida por el API será aceptada Las demás aplicaciones recibirán un mensaje DISCONNECT_IND, el cual indica esta situación en el parámetro Reason.

Controller: direcciona una unidad hardware que provee a la aplicación con acceso a ISDN. Un controlador puede soportar zero, una o muchas conexiones físicas y lógicas. El parámetro controller es una dword (1-127) el bit 7 indica si el mensaje aplica a un equipo externo o interno. Los controladores son numerados secuencialmente y pueden estar designados para manejar equipo externo en adición a capacidades internas, o puede proveer acceso exclusivamente a equipo externo, tal como un microteléfono por ejemplo.

El equipo externo como un manejador de canal B, no esta definido por el API

LISTEN_CONF

Confirma la aceptación de un LISTEN_REQ. Los errores están codificados en el parámetro Info.

Info: 0: Listen esta activa.

0x2002: Controlador incorrecto.

0x2005: No hay recursos Listen disponibles.

0x2007: Parámetros mensaje codificado incorrectamente.

CAPI_RELEASE

Las aplicaciones utilizan CAPI_RELEASE para terminar su registro con el CAPI, toda la memoria y otros recursos asignados para la aplicación por el CAPI serán liberados.

CONNECT_IND

Este mensaje indica una llamada entrante para una conexión física. A la llamada entrante se le asigna un PLCI el cual es utilizado para identificas esta conexión en los siguientes mensajes.

PLCI: Identificación de la conexión del enlace físico.

CIP Value: Información del perfil de compatibilidad.

Called party number: Número de la parte llamada.

Calling party number: Número de la parte llamante.

Called party subaddress: Subdireción de la parte llamada.

Calling party subaddress: Subdirección de la parte llamante.

BC: capacidades portadoras.

Capacidad de transferencia de información (octeto 3).

Bits

5 4 3 2 1

0 0 0 0 0 Conversación.

0 1 0 0 0 Información digital sin restricciones.

0 1 0 0 1 Información digital restringida.

1 0 0 0 0 Audio de 3,1 kHz.

1 0 0 0 1 Información digital sin restricciones con tonos/anuncios (nota 2).

1 1 0 0 0 Vídeo.

Los demás valores están reservados.

LLC: Capacidades de bajo nivel.

HLC: Capacidades de alto nivel.

Additional Info: Elementos de información adicional.

Calling party number: Segundo número de la parte llamante.

ALERT_REQ

Este mensaje debe ser utilizado por las aplicaciones para indicar compatibilidad con una llamada entrante. Este comando envía una alerta a la red para prevenir la llamada del

expiring (“no respuesta de usuario”). Si una aplicación es capaz de aceptar la llamada inmediatamente este no necesita utilizar este mensaje, pero puede publicar una CONNECT_RESP al CAPI directamente.

PLCI: Identificador de la conexión del enlace físico.

Additional info: Información adicional.

DISCONNECT_IND

Este mensaje indica el clareo del identificación del canal físico a través del parámetro

PLCI. El parámetro Reason indica la causa de este clareo.

PLCI : Identificación de la conexión del enlace físico.

Reason: 0: Causa no disponible.

0x3301: Error de protocolo, Nivel 1.

0x3302: Error de protocolo, Nivel 2.

0x3303: Error de protocolo, Nivel 3.

0x3304: Otra aplicación tomo la llamada.

0x3305: Clareada por Call Control Supervisión.

0x34xx: Causa de desconexión en acuerdo con Q.850/ETS 300 102-1. El valor de la causa esta indicada en el campo xx.

DISCONNECT_RESP

Con este mensaje, la aplicación acepta el clareo del canal físico.

PLCI : Identificación de la conexión del enlace físico.

INFO_IND

Este mensaje indica un evento para una conexión física como se expresa en el elemento de información (Info Element) cuyo código es descrito en el parámetro Info nombre. La conexión se identifica con el parámetro Controller/PLCI.

Controller/PLCI: Controlador / Identificación de la conexión del enlace físico.

Info number: Identificador del elemento de información.

Info element: Estructura dependiente del elemento de información.

INFO_RESP

Con este mensaje, la aplicación acepta la recepción de un INFO_IND.

CONNECT_REQ

Este mensaje inicia la configuración de una conexión física. Una aplicación solamente necesita proveer solamente parámetros relevantes(como: Controlador, CIP value y usualmente número de la parte llamante).

Controller; Controlador.

CIP Value: Información del perfil de compatibilidad.

0:Perfil no definido.

1:Voz.

2:Unrestricted digital information.

3: Restricted digital information.

4: 3.1 kHz audio.

5: 7.0 kHz audio.

6: Video.

7: modo paquete.

8: 56 kbit/s rate adaptation.

9: Unrestricted digital information with tones/announce-ments.

10 a 15reserved.

16: Telephony.

17 Group 2/3 fax.

18: Group 4 fax class 1.

19: Teletex service (basic and mixed) and group 4 facsimile service Classes II and III

20: Teletex service (basic and processable).

21: Teletex service (basic).

22: International interworking for Videotex.

23: Telex.

24: Message handling systems according X.400.

25: OSI applications according X.200.

26: 7 kHz Telephony.

27: Video Telephony F.721, first connection.

28: Video Telephony F.721, second connection.

29 a 31: reserved.

Called party number: Número de la parte llamada.

Calling party number: Número de la parte llamante.

Called party subaddress: Subdireción de la parte llamada.

Calling party subaddress: Subdirección de la parte llamante.

B protocol: Protocolo del canal B, que será utilizado.

BC: capacidades portadoras.

Capacidad de transferencia de información (octeto 3).

Bits

5 4 3 2 1

0 0 0 0 0 Conversación.

0 1 0 0 0 Información digital sin restricciones.

0 1 0 0 1 Información digital restringida.

1 0 0 0 0 Audio de 3,1 kHz.

1 0 0 0 1 Información digital sin restricciones con tonos/anuncios (nota 2).

1 1 0 0 0 Vídeo.

Los demás valores están reservados.

LLC: Capacidades de bajo nivel.

HLC: Capacidades de alto nivel.

Additional Info: Elementos de información adicional.

CONNECT_RESP

Este mensaje es utilizado por la aplicación para reaccionar frente una llamada entrante. La llamada entrante esta identificada por el parámetro PLCI. El parámetro Reject es utilizado para aceptar, rechazar o ignorar la llamada.

PLCI : Identificación de la conexión del enlace físico.

Reject :

0: Aceptar la llamada.

1: Ignorar la llamada.

2: Rechazar la llamada, llamada normal clareada.

3: Rechazar la llamada, usuario ocupado.

4: Rechazar la llamada, solicitud de circuito o canal no disponible.

5: Rechazar la llamada, facilidad rechazada.

6: Rechazar la llamada, canal inaceptable.

7: Rechazar la llamada, incompatibilidad de destino.

8: Rechazar la llamada, destino averiado.

0x34xx: El contenido del byte xx será señalado por la red en un elemento de información de causa (octeto 4). Este es el responsable de proveer el valor que es correctamente codificado con Q.931/ETS 300 102-1.El controlador enviara su valor de causa codificado con el estándar CCITT (octeto 3).

B protocol: Protocolo de canal B a ser utilizado.

Connected number: Número conectado.

Connected subaddres: Dirección del número conectado.

LLC: Bajo nivel de compatibilidad.

Additional Info: elementos de información adicional.

CONNECT_B3_IND

Este mensaje indica la llegada de una conexión lógica en una conexión física (llamada entrante). La conexión que llega se le asigna un NCCI para identificación, si está disponible, es transferida por el parámetro NCPI.

NCCI Identificación de la conexión de control de red.

NCPI Información del protocolo de control de red.

CONNECT_B3_RESP

Con este mensaje, la aplicaicón acepta o rechaza una conexión lógica entrante. La llada entrante es identificada por el parámetro NCCI. La llamada es aceptada o rechazada utilizando el parámetro reject. El parámetro NCPI puede ser utilizado para transferir información adicional dependiente del protocolo.

NCCI: Identificación de la conexión de control de red.

Reject:

0: Aceptar la llamada.

2: Rechazar la llamara, normalmente clareo de llamada.

NCPI: Información del protocolo de control de red.

CONNECT_CONF

Este mensaje confirma la iniciación de una configuración de llamada. A esta conexión se le asigna un PLCI, el cual sirve para que sea identificada en procesos futuros. Los errores son retornados en el parámetro Info.

PLCI: Identificación de la conexión del enlace físico.

Info: 0: Iniciar la conexión.

0x2002: Controlador ilegal.

0x2003: No PLCI disponible.

0x2007: codificar parámetro de mensaje ilegal.

0x3001: Protocolo B1 no soportado.

0x3002: Protocolo B2 no soportado.

0x3003: Protocolo B3 no soportado.

0x3004: Parámetro del protocolo B1 no soportado.

0x3005: parámetro del protocolo B2 no soportado.

0x3006: Parametro del protocolo B3 no soportado.

0x3007: Combinación de protocolo B no soportado.

0x3009: Valor CIP desconocido.

CONNECT_ACTIVE_IND

Este mensaje indica la conexión física de un canal B. La conexión es identificada por el parámetro PLCI.

PLCI: Identificación de la conexión del enlace físico.

Connected number Número conectado.

Connected subaddress Subdirección del número conectado.

LLC Compatibilidad de bajo nivel.

CONNECT_ACTIVE_RESP

Con este mensaje la aplicación acepta la recepción de un CONNECT_ACTIVE_IND

PLCI: Identificación de la conexión del enlace físico.

CONNECT_B3_REQ

Este mensaje inicializa la configuración de una conexión lógica. La conexión física es identificada por el parámetro PLCI. Información dependiente del protocolo puede ser transferida utilizando el parámetro NCPI.

PLCI: Identificación de la conexión del enlace físico.

NCCI: Identificación de la conexión de control de red.

CONNECT_B3_CONF

Este mensaje confirma el inicio de la configuración de una conexión lógica. A la conexión lógica se le asigna un NCCI para identificación futura. La información de error es suministrada por el parámetro Info.

NCCI: Identificación de la conexión de control de red.

Info 0: Conexión iniciada.

0x0001: NCPI no soportado por el protocolo actual, NCPI ignorada.

0x2001: Mensaje no soportado en el estado actual.

0x2002: Ilegal PLCI.

0x2004: No NCCI disponible.

0x3008: NCPI no soportado.

CONNECT_B3_ACTIVE_IND

Este mensaje indica la conexión lógica de un canal B. La conexión es identificada por el parámetro NCCI. El parámetro NCPI es utilizado para trasferir información adicional dependiente del protocolo.

NCCI: Identificación de la conexión de control de red.

NCPI: Información del protocolo de control de red.

CONNECT_B3_ACTIVE_RESP

Con este mensaje la aplicación acepta la recepción de un CONNECT_B3_ACTIVE_IND

NCCI: Identificación de la conexión de control de red.

DATA_B3_REQ

Este mensaje envia datos dentro de la conexión lógica identificada por el NCCI. Los datos a ser enviados son referenciados por los parámetros Data/Data length. El dato no esta cometido en el mensaje: un puntero de 32 bits es utilizado para transferir la dirección del área de datos. La aplicación publica un identificador único para este bloque de datos en el parámetro Data hadle. Este manejador es utilizado en un posterior DATA_B3_CONF. Es posible fijar una información adicional, como una indicación de más flujos de datos, entrega de confirmación, etc. en el parámetro Flags. Las banderas no son soportadas por todos los protocolos.

NCCI: Identificación de la conexión de control de red.

Data: Puntero al dato a ser enviado.

Data length: Tamaño del área de datos que será enviada.

Data handle: Referenciado en DATA_B3_CONF.

Flags [0]: Qualifier bit.

[1]: Más datos bit.

[2]: Confiramción de envío bit.

[3]: datos enviados.

[4]: UI frame.

[5] to [15]: reservados.

Data64: Para aplicaciones de 64bit: puntero de 64-bit para los datos a ser enviados para las otras aplicaciones está reservado.

Nota: La transferencia de datos no soporta ensamble o re ensamblado de datos.

DISCONNECT_B3_IND

Este mensaje indica el clareo del identificador de conexión lógica por el parámetro NCCI. El parámetro Reason_B3 indica si este clear-down fue causado por un comportamiento incorrecto del protocolo o por una llamada de control de supervisión. El parámetro NCPI es utilizado para indicar una información adicional dependiente del protocolo, si esta disponible.

NCCI: Identificación de la conexión de control de red.

Reason_B3: 0: Clareo de acuerdo al protocolo.

0x3301: Error de Protocolo, Nivel 1.

0x3302: Error de Protocolo, Nivel 2.

0x3303: Error de Protocolo, Nivel 3.

0x3305: Clareado por supervición de control de llamada.

NCPI: Información del protocolo de control de red

DISCONNECT_B3_RESP

Con este mensaje, la aplicación acepta el clareo de una conexión lógica

NCCI: Identificación de la conexión de control de red

DISCONNECT_REQ

Este mensaje inicializa el clareo de una conexión física, identificado por el parámetro PLCI.

PLCI: Identificación de la conexión del enlace físico.

Additional Info: Elementos de información adicional.

DISCONNECT_B3_CONF

Este menaje confirma que el clareo de una conexión lógica ha sido inicializado. Cualquier error es codificado en el parámetro Info.

NCCI: Identificación de la conexión de control de red.

Info 0: Desconexión iniciada.

0x0001: NCPI no soportado por el protocolo actual, NCPI ignorada.

0x2001: Mensaje no soportado en el estado actual.

0x2002: Ilegal PLCI.

0x2004: No NCCI disponible.

0x3008: NCPI no soportado.

DISCONNECT_CONF

Este mensaje confirma la iniciación del clareo de una conexión física. Cualquier error es codificado en el Parámetro Info.

PLCI: Identificación de la conexión del enlace físico.

Info 0: Desconexión iniciada.

0x2001: Mensaje no soportado en el estado actual.

0x2002: Ilegal PLCI.

0x2007: Ilegal codificado del parámetro del mensaje.

DATA_B3_CONF

Este mensaje confirma la aceptación de un paquete de datos a ser enviado. La conexión lógica es identificada por el parámetro NCCI. El parámetro Data handle contiene el identificador dado por la aplicación asociado con DATA_B3_REQ. Luego de recibir este mensaje, la aplicación puede reutilizar el área de datos referenciada.

NCCI: Identificación de la conexión de control de red.

Data handle word Identifies the corresponding DATA_B3_REQ.

Info: 0: Transmisión de datos inicializada.

0x0002: Banderas no soportada por el protocolo actual, banderas ignoradas.

0x2001: Mensaje no soportado en el estado actual.

0x2002: Ilegal NCCI.

0x2007: Ilegal codificado de parámetros de mensaje.

0x300A: Banderas no soportadas (bits reservados).

0x300C: Longitud de datos no soportados por el protocolo actual.

DATA_B3_IND

Este mensaje indica la llegada de datos dentro de una conexión lógica. La conexión lógica es identificada por el NCCI. La longitud del área de datos de entrada es indicada por el parámetro Data length. El área de datos de llegada por si misma puede ser referenciada por el parámetro Data. Los datos son están contenidos en el mensaje: un puntero de 32 bits es utilizado para indicar la dirección el área de datos. CAPI comúnmente publica un manejador correspondiente a esta área en el parámetro Data handle. Cuando la aplicación confirma la recepción de el data enviando un mensaje DATA_B3_RESP, este debe igualmente suplir este manejador. La información adicional específica del protocolo disponible (como una indicación de más flujos de datos, confirmación de envió, etc.) es suplida en el parámetro Flags.

NCCI: Identificación de la conexión de control de red.

Data: Puntero al dato recibido.

Data length: Tamaño del área de datos que será recibida.

Data handle: Referenciado en DATA_B3_RESP.

Flags: [0]: Qualifier bit.

[1]: Más datos bit.

[2]: Confirmación de envío bit.

[3]: datos enviados.

[4]: Brak (protocolos B2 9,10,11).

[5] a [14]: reservados.

[15]: framing error bit: los datos pueden ser inválidos (Protocolo B2 6, 9, 11).

Data64: Para aplicaciones de 64bit: puntero de 64-bit para los datos a ser enviados para las otras aplicaciones está reservado.

Nota: La transferencia de datos no soporta ensamble o re ensamblado de datos.

DATA_B3_RESP

Con este mensaje, la aplicación acepta la recepción de un paquete de datos entrante. La conexión lógica es identificada por el parámetro NCCI. El parámetro Data handle identifica el correspondiente DATA_B3_IND.

NCCI: Identificación de la conexión de control de red.

Data handle: Referenciado en DATA_B3_IND.

DISCONNECT_B3_REQ

Este mensaje inicializa el clareo del identificador de conexión lógica por el parámetro NCCI. El parámetro NCPI puede ser utilizado para transferir información adicional dependiente del protocolo.

NCCI: Identificación de la conexión de control de red.

NCPI: Información del protocolo de control de red.

ANEXO C. MANUALES DE USUARIO

El Servicio de Video por Demanda, le permitirá a los usuarios que se han suscrito previamente, desde la comodidad de sus hogares acceder a una librería de películas que se encuentran almacenadas remotamente en formato digital, realizando una búsqueda y reproduciendo la selección en su equipo terminal.

El usuario obtiene las funcionalidades que le presta un VCR convencional, como es adelantar, retroceder, parar o pausar una película que se encuentre reproduciendo; igualmente le permite modificar la calidad con la que está recibiendo el video, mejorando el ancho de banda asignada a su comunicación, todo esto en tiempo real.

Al administrador del servicio le permite gestionar la lista de videos, suscribir usuarios al sistema y suspender temporal o definitivamente usuarios del mismo, al igual que establecer las características iniciales para la prestación del servicio a todos sus usuarios.

Requerimientos Hardware:

- Procesador 486 o Pentium (Recomendado Pentium 100MHz o superior).
- Tarjeta de video con soporte a modo 8,16,24 o 32 (16-bit recomendado para presentación de video rápido).
- Tarjeta de sonido.

Requerimientos Software:

- Windows 95/98 o superior.
- JDK 1.1.6. o superior
- Clases JMF y librerías natives (inlcuidas en el CD)

INSTALACIÓN

Para realizar la instalación del sistema, por favor inicie la aplicación de instalación y siga las instrucciones. Este instalador configurará su computador para que pueda hacer uso del servicio y descargara todas las librerías que requiere.

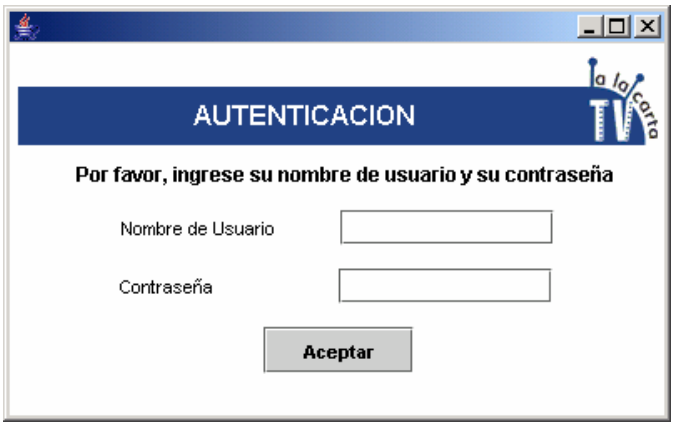
UTILIZACIÓN

USUARIOS:

Bienvenidos al servicio de video por demanda, luego de haberse suscrito al servicio (tanto con el proveedor de servicios, como con el proveedor del servicio de video por demanda) y de haber realizado una adecuada instalación del sistema en su computador, por favor siga los siguientes pasos para poder disfrutar de las mejores películas en la comodidad de sus hogares:

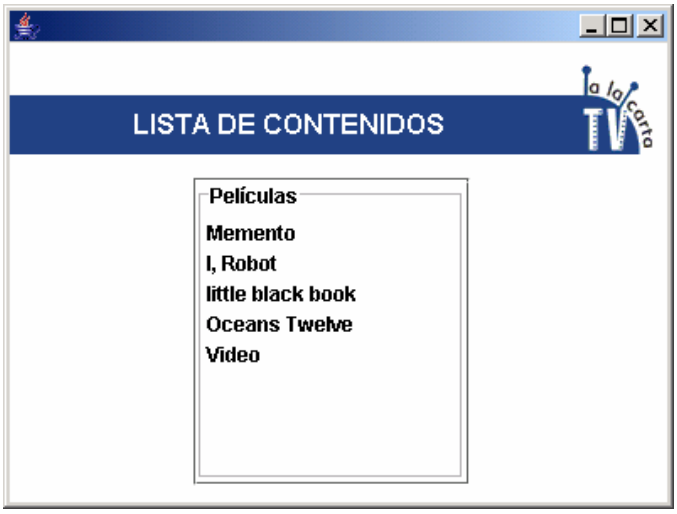
Inicie la aplicación de usuario para el servicio de video por demanda que ha instalado previamente. Esta es la interfaz gráfica que encontrará:

Seleccione la opción: Registrarse en el sistema y a continuación se le desplegará la siguiente interfaz gráfica que le permitirá realizar la correcta autenticación del servicio:

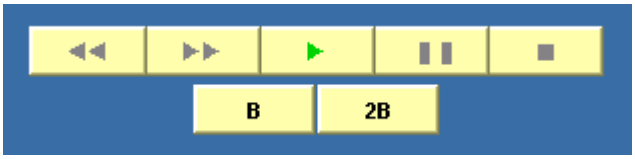


En este momento deberá ingresar su nombre de usuario y contraseña con la cual previamente se ha suscrito tanto con el proveedor de servicios, como con el proveedor del servicio de video por demanda, a continuación seleccione el botón Aceptar.

Si su autenticación es correcta, se le desplegará la siguiente interfaz con la lista de las diferentes películas a las cuales se encuentra suscrito.



Para seleccionar la película que desee que sea reproducida, simplemente haga clic sobre la película de su elección y en pocos momentos iniciará la reproducción de la película que ha seleccionado, y se iniciará la siguiente interfaz que le permite modificar la reproducción de la misma.



Esta interfaz le da las funcionalidades de un VCR convencional, por lo tanto le permitirá adelantar, retroceder, parar o pausar una película que se encuentre reproduciendo, todo esto en tiempo real.
Una funcionalidad adicional que presenta este servicio, es la posibilidad que le brinda al usuario de modificar la calidad de reproducción de la película, para esto simplemente seleccione el botón B (para seleccionar un canal 64Kbps) o 2B (para seleccionar dos canales 128Kbps) con el cual se le disminuye o incrementa la calidad del video.

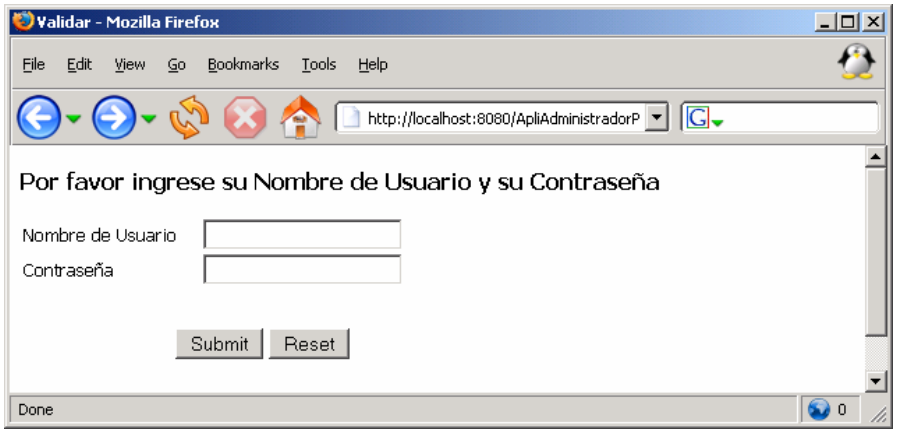
En cualquier momento puede finalizar la reproducción del video, para esto haga clic sobre el botón parar reproducción.

Al finalizar la reproducción de la película se finalizará la comunicación con el servicio de video por demanda,

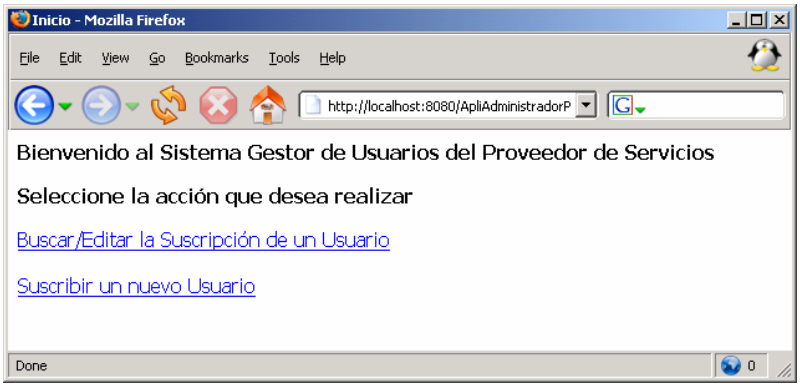
ADMINISTRADOR DEL PROVEEDOR DE SERVICIOS:

Bienvenidos a la aplicación del proveedor de servicios, esta aplicación le permitirá administrar a la información de los diferentes usuarios que se encuentran registrados con el proveedor de servicio:

Luego de iniciar un navegador e introducir la dirección en la que se encuentra la aplicación de administrador, se inicia la siguiente página en la que el administrador debe registrarse al sistema:

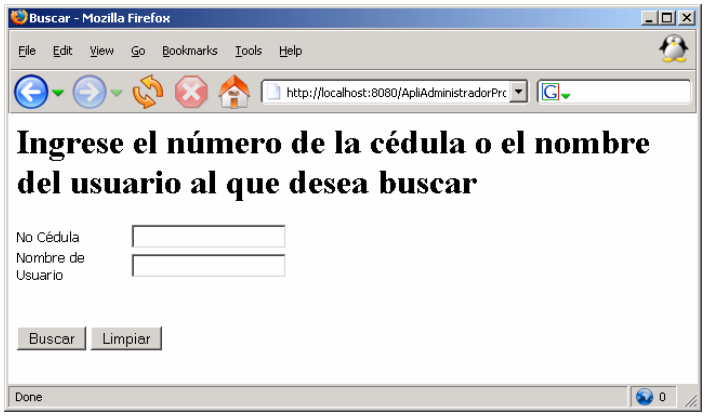


Luego de introducir el nombre de usuario y la contraseña correcta el sistema le despliega el siguiente menú con las opciones que tiene:

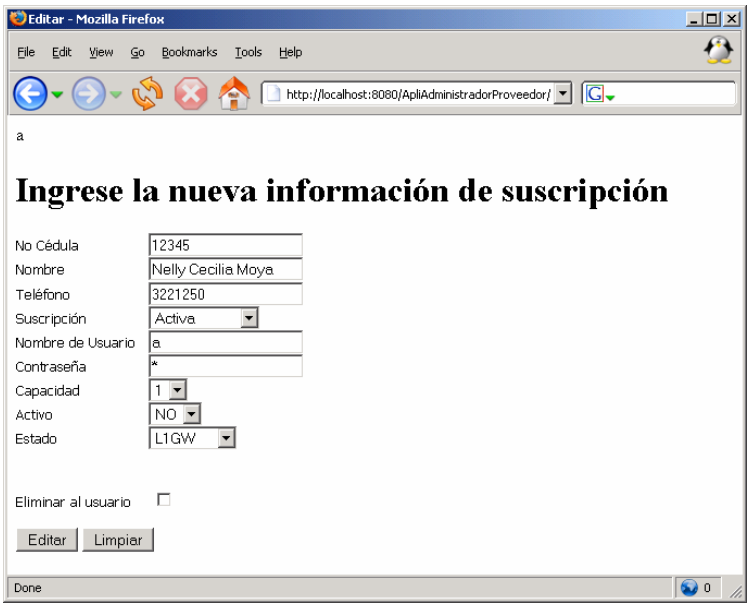


- Buscar/Editar la Suscripción de un Usuario: Le permite buscar a partir del nombre de usuario o la cédula de este su información correspondiente (número de cédula, nombre, teléfono, suscripción (Activa o Suspendida), nombre de usuario, contraseña, capacidad, Activo (si, no) y Estado) permitiéndole igualmente editarla o eliminar al usuario.

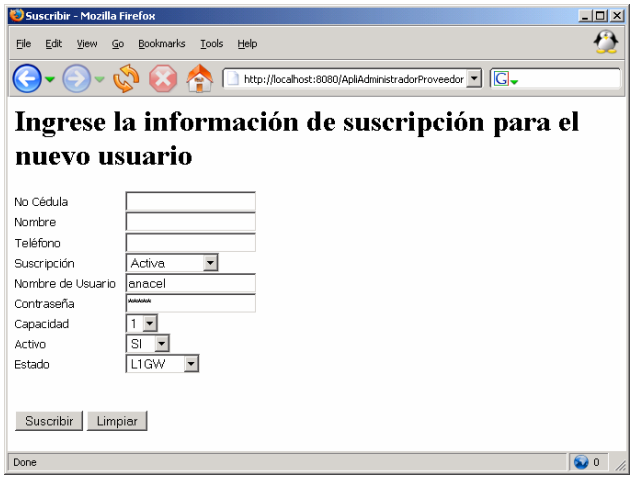
  En esta interfaz el usuario introduce el tipo de búsqueda que desea realizar:



  En esta interfaz le aparece la información del usuario para que el administrador pueda modificar:



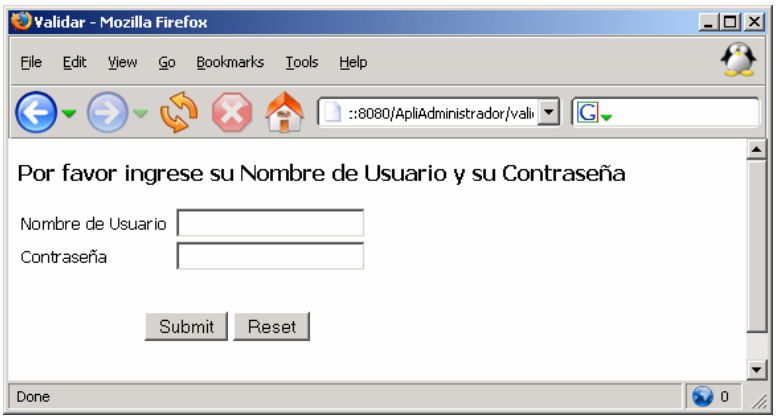
- Suscribir un nuevo Usuario: La cual le permite agregar a la base de datos la información correspondiente a un nuevo usuario, es importante tener en cuenta que ni el número de cédula ni el nombre de usuario pueden encontrarse ya inscritos.
  En esta interfaz el administrador introduce la información de un nuevo usuario.



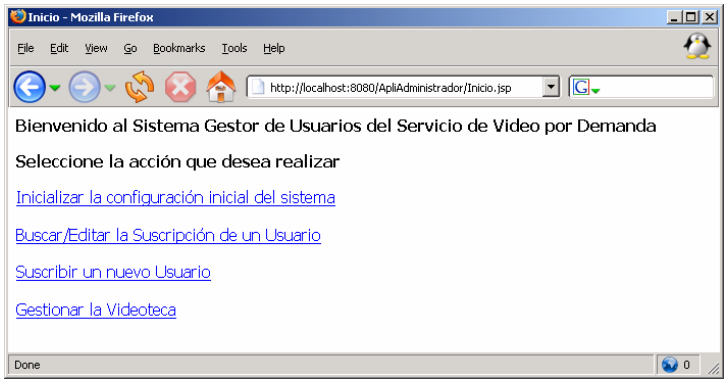
ADMINISTRADOR DEL PROVEEDOR DE SERVICIO DE VIDEO POR DEMANDA:

Bienvenidos a la aplicación del proveedor de servicio de video por demanda, esta aplicación le permitirá administrar a la información de los diferentes usuarios que se encuentran registrados con el servicio:

Luego de iniciar un navegador e introducir la dirección en la que se encuentra la aplicación de administrador, se inicia la siguiente página en la que el administrador debe registrarse al sistema:

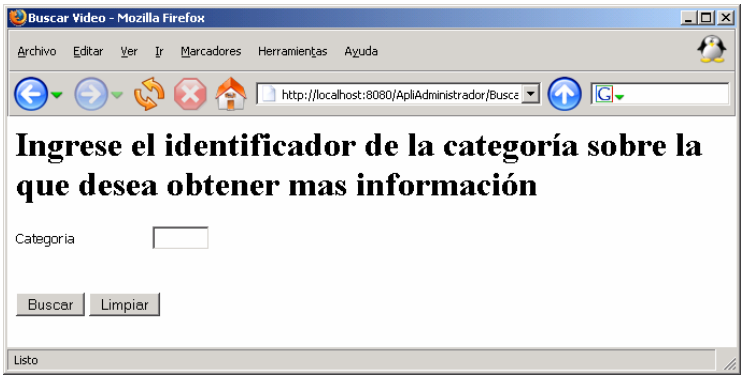


Luego de introducir el nombre de usuario y la contraseña correcta el sistema le despliega el siguiente menú con las opciones que tiene:

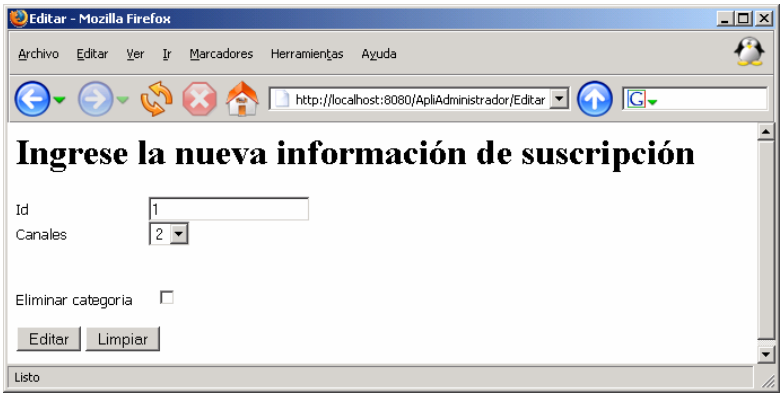


- Inicializar la configuración inicial del sistema: Le permite buscar a partir del identificador de una categoría en particular, el número de canales con los cuales los diferentes usuarios asociados a esa categoría, inician el servicio.

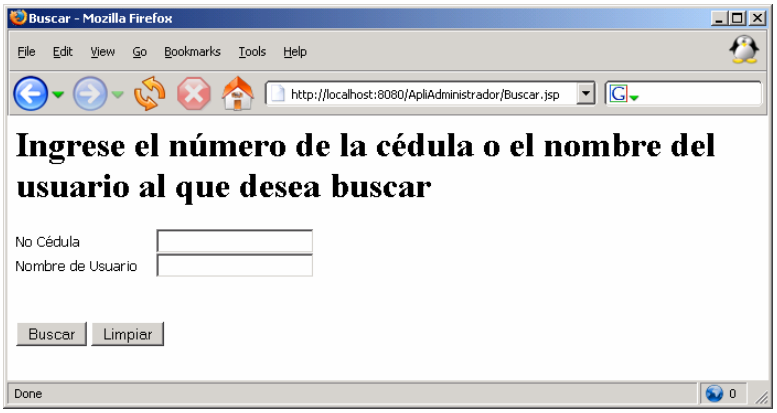
  En esta interfaz, el administrador introduce el identificador de la categoría sobre la que desea obtener más información:



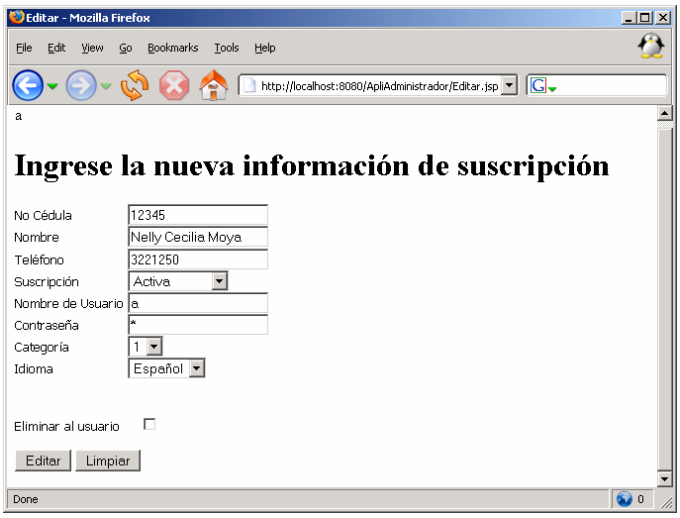
En esta interfaz le aparece la información de la configuración inicial de la categoría seleccionada, para que el administrador pueda modificar:



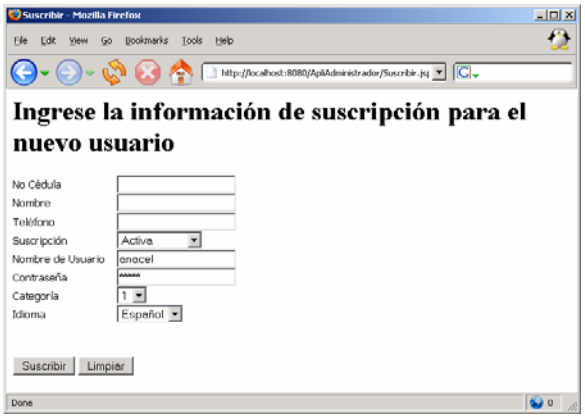
- Buscar/Editar la Suscripción de un Usuario: Le permite buscar a partir del nombre de usuario o la cédula de este su información correspondiente (número de cédula, nombre, teléfono, suscripción (Activa o Suspendida), nombre de usuario, contraseña, Categoría e Idioma) permitiéndole igualmente editarla o eliminar al usuario.
  En esta interfaz el usuario introduce el tipo de búsqueda que desea realizar:



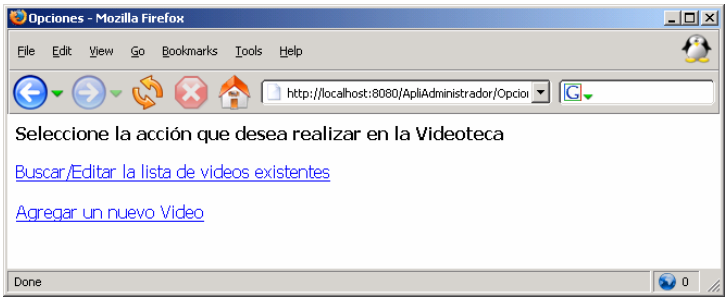
En esta interfaz le aparece la información del usuario para que el administrador pueda modificar:



- Suscribir un nuevo Usuario: La cual le permite agregar a la base de datos la información correspondiente a un nuevo usuario, es importante tener en cuenta que ni el número de cédula ni el nombre de usuario pueden encontrarse ya inscritos.
  En esta interfaz el administrador introduce la información de un nuevo usuario.

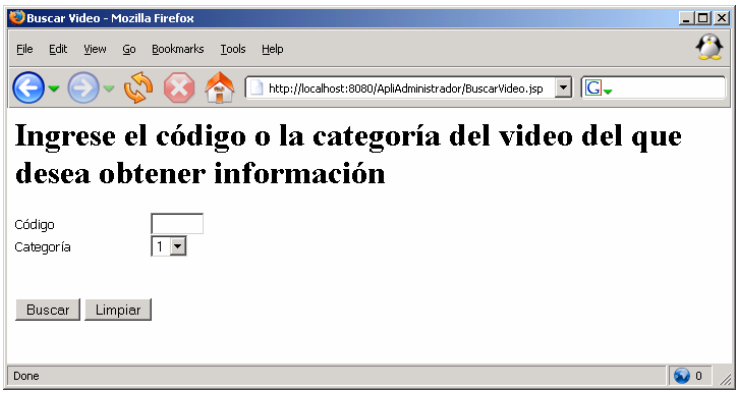


- Gestionar la videoteca: le permite al administrador:
  En esta interfaz el usuario obtiene la lista de funcionalidades que puede realizar sobre la videoteca:



- Buscar/ editar la lista de video existentes: Le permite buscar a partir del código del video y la categoría de este la información correspondiente al video (Código, Nombre, Categoría, Género, Ruta Archivo de Origen 1B, Ruta Archivo de Envío 1B, Ruta Archivo de Origen 2B, Ruta Archivo de Envío 2B y duración en segundos) permitiéndole igualmente editarla o eliminar el video

  En esta interfaz el usuario introduce el tipo de búsqueda que desea realizar:



- Agregar un nuevo video: La cual le permite agregar a la base de datos la información correspondiente a un nuevo video, es importante tener en cuenta que ni el código ni el nombre del video pueden encontrarse ya inscritos. Esta aplicación carga la información del video a partir de la dirección de origen, convirtiéndola en un objeto de datos para ser enviado por el módem, la duración de este proceso será igual a la duración del video, luego de terminar la serialización del video para 1B, realiza el mismo procedimiento para el 2B, para finalmente realizar una tabla de comparación que va a permitir el cambio de la transmisión de un tipo de video al otro.

  En esta interfaz el administrador introduce la información de un nuevo usuario.