

1 IMPLEMENTACION DETALLADA DEL UR

1.1 DIAGRAMA DE CASOS DE USO

En la Figura 1 se ilustra el diagrama de Casos de Uso del UR.

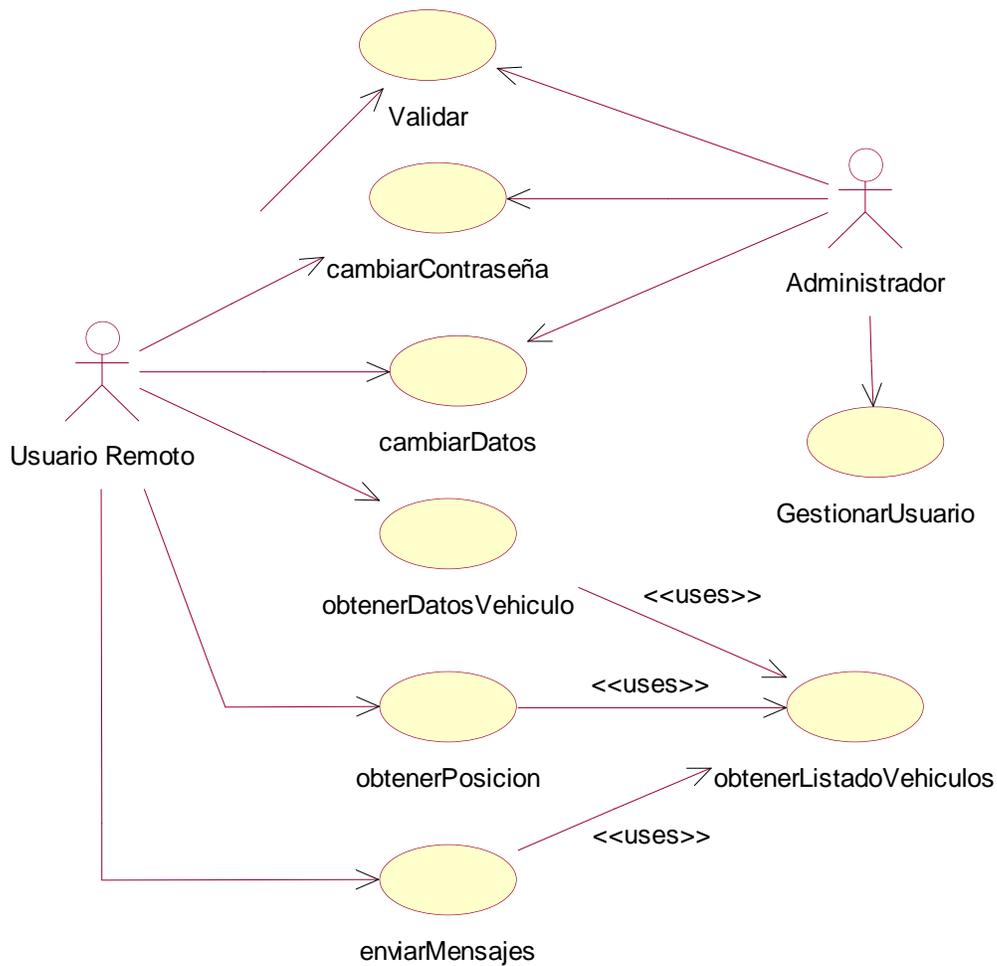


Figura 1. Diagrama de Casos de Usos del UR



1.1.1 Caso de Uso No. 1: Validar

Iniciador: Usuario Remoto, Administrador.

Precondición: Ninguno

Propósito: Permitir el ingreso al sistema sólo a usuarios que han sido registrados.

Resumen: La función de este caso de uso es verificar que los usuarios que ingresan al sistema estén registrados con anterioridad. Además por medio de los datos de login y contraseña suministrados por el usuario, se identifican las funcionalidades a las que tiene acceso dicho usuario (administrador o usuario del sistema).

Flujo Principal:

1. Iniciar la aplicación.
2. Ingresar el login y la contraseña en los campos correspondientes.
3. Si el usuario accede a la aplicación por medio del dispositivo móvil debe seleccionar *enviar*. Si el usuario ingresa por medio de la aplicación Web debe dar clic en *aceptar*.
4. El sistema se conecta al servidor y analiza los datos ingresados. Las posibles respuestas son:

 *Validación correcta:* De acuerdo a los datos ingresados identifica el tipo de usuario y despliega la pantalla correspondiente.

 *El usuario no se encuentra registrado:* Si el usuario accede a través del dispositivo móvil se despliega un Alert (mensaje de advertencia) indicando que el usuario no se encuentra registrado. Si el usuario ingresa a través de la aplicación Web, se despliega un script indicando que el usuario no se encuentra registrado.

Post Condiciones: Sección de administrador o usuario iniciada según los datos que se ingresen.

Excepciones:

- 🌐 *NullPointerException:* Se lanza cuando la Alerta o el Displayable son nulos.
- 🌐 *IllegalArgumentException:* Si el siguiente Displayable es un Alert.
- 🌐 *IllegalArgumentException:* Si el tamaño máximo del campo de texto es cero o negativo. También es lanzada en el caso de que los valores de los parámetros o el texto son inválidos o por que la longitud de la cadena ingresada excede la capacidad máxima especificada.
- 🌐 *NullPointerException:* Lanzada cuando la etiqueta de los Command son nulas.
- 🌐 *IllegalArgumenteException:* Si el commandType es un tipo inválido.
- 🌐 *IlegalArgummentException:* Se lanza cuando se introducen parámetros inválidos al abrir la conexión HTTP.
- 🌐 *ConnectionNotFoundException:* Si la petición de conexión no puede realizarse o si el tipo de protocolo especificado no existe.
- 🌐 *IOException:* Si ocurre alguna otra clase de error tipo I/O.

Recursos Especiales Utilizados: Conexión con el SC.

GUI's Relacionadas:

- 🌐 Acceso a través del dispositivo móvil: validación
- 🌐 Acceso a través de la página Web: index.html

1.1.2 Caso de Uso No. 2: cambiarContraseña

Iniciador: Usuario Remoto, Administrador.



Precondición: El usuario tiene que ser validado por el sistema.

Propósito: Permitir al usuario cambiar la contraseña de validación del sistema.

Resumen: El usuario debe ingresar la nueva contraseña y confirmarla en los campos respectivos. Si alguno de estos campos se encuentra vacío o los datos ingresados no coinciden el sistema no actualiza la contraseña. El sistema reporta el resultado de la operación.

Flujo Principal:

1. Realizar los pasos de validación indicados en el caso de uso No. 1.
2. Si el usuario ingresa a través del teléfono celular, se despliega el menú inicial, debe escoger la opción de *Contraseña* que se encuentra de primera en la segunda fila. Si el usuario ingresa a través de la página Web, debe dar clic en el botón de cambiar contraseña.
3. Ingresar la nueva contraseña y confirmarla en los campos correspondientes.
4. Seleccionar la opción de cambiar/dar clic en el botón de cambiar.
5. Esperar respuesta del sistema. Estas pueden ser:

 *Cambio exitoso:* Si se ingresa a través del teléfono celular se despliega una alerta y vuelve al menú inicial. Si se ingresa por la página Web, sólo se da la confirmación y el usuario escoge la opción que desea seguir a través de los botones de navegación.

 *Cambio no exitoso:* Se despliega que el cambio no pudo ser realizado, se borran los datos que el usuario ha ingresado y se deja la pantalla lista para que el usuario ingrese de nuevo los datos.

Post Condiciones: Ninguna.

Flujos Alternativos:



- Si el usuario selecciona la opción de cambiar, el sistema antes de enviar la información analiza que los campos no se encuentren vacíos y que los datos ingresados coincidan. En caso contrario despliega una alerta al usuario indicando el problema correspondiente.
- Volver al menú inicial.
- Cancelar la operación.

Excepciones:

- *NullPointerException*: Se lanza cuando la Alerta o el Displayable son nulos.
- *IllegalArgumentException*: Si el siguiente Displayable es un Alert.
- *IllegalArgumentException*: Si el tamaño máximo del campo de texto es cero o negativo. También es lanzada en el caso de que los valores de los parámetros o el texto son inválidos o por que la longitud de la cadena ingresada excede la capacidad máxima especificada.
- *NullPointerException*: Lanzada cuando la etiqueta de los Command son nulas.
- *IllegalArgumenteException*: Si el commandType es un tipo inválido.
- *IlegalArgummentException*: Se lanza cuando se introducen parámetros inválidos al abrir la conexión HTTP.
- *ConnectionNotFoundExpection*: Si la petición de conexión no puede realizarse o si el tipo de protocolo especificado no existe.
- *IOException*: Si ocurre alguna otra clase de error tipo I/O.

Recursos Especiales Utilizados: Conexión al SC.

GUI's Relacionadas:

- Acceso a través del dispositivo móvil: menuInicial, contraseña.
- Acceso a través de la página Web: menuInicial.html, contraseña.html

1.1.3 Caso de Uso No. 3: cambiarDatos

Iniciador: Usuario Remoto, Administrador.

Precondición: El usuario debe estar validado al sistema.

Propósito: Permitir al usuario cambiar los datos de contacto: correo electrónico y número celular.

Resumen: El sistema debe tener registrado el correo electrónico y el número celular actual del usuario para poder enviar la información correspondiente a los servicios que presta SIRIO, tales como el envío de mensajes y mensajes de emergencia.

Flujo Principal:

1. Realizar los pasos de validación descritos en el caso de uso No. 1
2. Si el usuario ingresa a través del teléfono celular, se despliega el menú inicial, debe escoger la opción de *Act.Datos* que se encuentra de segunda en la primera fila. Si el usuario ingresa a través de la página Web, debe dar clic en el botón *Actualizar Datos*.
3. Se despliega la pantalla correspondiente con los datos que tiene almacenados el sistema.
4. El usuario debe ingresar los datos que desea actualizar en los campos correspondientes.
5. Seleccionar la opción de *Actualizar*/dar clic en el botón *Actualizar*.
6. Esperar respuesta del sistema. Estas pueden ser:
 -  *Actualización exitosa:* Si se ingresa a través del teléfono celular se despliega una alerta y vuelve al menú inicial. Si se ingresa por la página Web, sólo se da la confirmación y el usuario escoge la opción que desea seguir a través de los botones de navegación.



- 🌐 *Cambio no exitoso:* Se despliega que el cambio no pudo ser realizado, y se despliega nuevamente la pantalla con los últimos datos que ha ingresado el usuario.

Post Condiciones: Ninguna

Flujos Alternativos:

- 🌐 Si el usuario selecciona la opción de actualizar datos, el sistema analiza si se realizaron cambios, en caso negativo, no se realiza la actualización y vuelve al menú inicial si el usuario ha accedido a través del teléfono celular.
- 🌐 Volver al menú inicial.
- 🌐 Cancelar la operación.

Excepciones:

- 🌐 *NullPointerException:* Se lanza cuando la Alerta o el Displayable son nulos.
- 🌐 *IllegalArgumentExeption:* Si el siguiente Displayable es un Alert.
- 🌐 *IllegalArgumentExeption:* Si el tamaño máximo del campo de texto es cero o negativo. También es lanzada en el caso de que los valores de los parámetros o el texto son inválidos o por que la longitud de la cadena ingresada excede la capacidad máxima especificada.
- 🌐 *NullPointerException:* Lanzada cuando la etiqueta de los Command son nulas.
- 🌐 *IllegalArgumenteExeption:* Si el commandType es un tipo inválido.
- 🌐 *IlegalArgummentExeption:* Se lanza cuando se introducen parámetros inválidos al abrir la conexión HTTP.
- 🌐 *ConnectionNotFoundExeption:* Si la petición de conexión no puede realizarse o si el tipo de protocolo especificado no existe.
- 🌐 *IOException:* Si ocurre alguna otra clase de error tipo I/O.

Recursos Especiales Utilizados: Conexión al SC.

GUI's Relacionadas:

-  Acceso a través del dispositivo móvil: menuInicial, cambiarDatos.
-  Acceso a través de la página Web: menuInicial.html, cambiarDatos.html

1.1.4 Caso de Uso No. 4: obtenerListadoVehiculos

Iniciador: Usuario Remoto.

Precondición: El usuario debe tener acceso al menos a un vehículo.

Propósito: Permitir al usuario obtener la lista de vehículos a la que tiene acceso.

Resumen: Un usuario sólo puede acceder a los vehículos a los que tiene permiso. Por lo tanto este caso de uso despliega los códigos (o si prefiere la placa) de los vehículos de los cuales puede ver información relacionada, la ubicación actual y enviar mensajes de texto. El usuario puede seleccionar uno o más vehículos de la lista.

Flujo Principal:

1. Realizar los pasos de validación descritos en el caso de uso No. 1
2. Si el usuario ingresa a través del teléfono celular, se despliega el menú inicial en el cual debe escoger la opción de *Vehículos* que se encuentra de primera en la primera fila. Si el usuario ingresa a través de la página Web, debe dar clic en el botón *Lista Vehículos*.
3. Se despliega la pantalla la lista con los códigos de vehículos que tiene acceso el usuario.

Post Condiciones: El usuario puede solicitar la información, ubicación actual y enviar mensajes de los vehículos seleccionados.

Flujos Alternativos:

- 🌐 Volver al menú inicial.
- 🌐 Cancelar la operación.

Excepciones:

- 🌐 *NullPointerException*: Se lanza cuando la Alerta o el Displayable son nulos.
- 🌐 *IllegalArgumentException*: Si el siguiente Displayable es un Alert.
- 🌐 *IllegalArgumentException*: Si el tamaño máximo del campo de texto es cero o negativo. También es lanzada en el caso de que los valores de los parámetros o el texto son inválidos o por que la longitud de la cadena ingresada excede la capacidad máxima especificada.
- 🌐 *NullPointerException*: Lanzada cuando la etiqueta de los Command son nulas.
- 🌐 *IllegalArgumenteException*: Si el commandType es un tipo inválido.
- 🌐 *IlegalArgummentException*: Se lanza cuando se ingresan parámetros inválidos al abrir la conexión HTTP.
- 🌐 *ConnectionNotFoundException*: Si la petición de conexión no puede realizarse o si el tipo de protocolo especificado no existe.
- 🌐 *IOException*: Si ocurre alguna otra clase de error tipo I/O.

Recursos Especiales Utilizados: Conexión al SC.

GUI's Relacionadas:

- 🌐 Acceso a través del dispositivo móvil: menuInicial, listaVehiculos.
- 🌐 Acceso a través de la página Web: menuInicial.html, listaVehiculos.html

1.1.5 Caso de Uso No. 5: obtenerDatosVehiculos

Iniciador: Usuario Remoto.



Precondición: Seleccionar un solo vehículo de la lista mencionada en el caso de uso No. 4.

Propósito: Dar la opción al usuario de solicitar información de los vehículos que tiene acceso.

Resumen: El sistema puede guardar alguna información relacionada con el vehículo, tales como placa, código, modelo, marca y el nombre del conductor. Esta información puede ser ingresada por el usuario a través de la página Web. En caso que el usuario no tenga esta información, se le despliega una alerta informando que no se encuentran datos disponibles. Este caso de uso sólo se habilita cuando el usuario ha seleccionado un solo vehículo de la lista desplegada.

Flujo Principal:

1. Realizar los pasos de validación descritos en el caso de uso No. 1
2. Escoger una opción del listado de vehículos.
3. Desplegar el menú principal.
4. Seleccionar la opción *Información* que se encuentra de primera en la primera fila. Si el usuario ingresa a través de la página Web, debe dar clic en el botón *Información Vehículo*.
5. Se despliega la pantalla con los datos del vehículo seleccionado.

Post Condiciones: Ninguna

Flujos Alternativos:

-  Volver al menú inicial.
-  Cancelar la operación.

Excepciones:

-  *NullPointerException*: Se lanza cuando la Alerta o el Displayable son nulos.



- *IllegalArgumentException*: Si el siguiente Displayable es un Alert.
- *IllegalArgumentException*: Si el tamaño máximo del campo de texto es cero o negativo. También es lanzada en el caso de que los valores de los parámetros o el texto son inválidos o por que la longitud de la cadena ingresada excede la capacidad máxima especificada.
- *NullPointerException*: Lanzada cuando la etiqueta de los Command son nulas.
- *IllegalArgumenteException*: Si el commandType es un tipo inválido.
- *IlegalArgummentException*: Se lanza cuando se pasan parámetros inválidos al abrir la conexión HTTP.
- *ConnectionNotFoundException*: Si la petición de conexión no puede realizarse o si el tipo de protocolo especificado no existe.
- *IOException*: Si ocurre alguna otra clase de error tipo I/O.

Recursos Especiales Utilizados: Conexión al SC.

GUI's Relacionadas:

- Acceso a través del dispositivo móvil: menuInicial, menuPrincipal, obtenerDatos.
- Acceso a través de la página Web: menuInicial.html, menuPrincipal.html, obtenerDatos.html

1.1.6 Caso de Uso No. 6: obtenerPosicion

Iniciador: Usuario Remoto.

Precondición: Seleccionar uno o más vehículos de la lista mencionada en el caso de uso No. 4.

Propósito: Dar opción al usuario de solicitar la posición actual de los vehículos que tiene acceso.



Resumen: Este caso de uso es el encargado de desplegar un mapa digital con la ubicación actual de uno o varios vehículos. Es el encargado de hacer la conexión y consultas con MapServer a través de una URL que la obtiene del SC. También permite manipular el mapa.

Flujo Principal:

1. Realizar los pasos de validación descritos en el caso de uso No. 1
2. Seleccionar uno o varios vehículos del listado de vehículos.
3. Desplegar el menú principal.
4. Seleccionar la opción *Ubi.Actual* que se encuentra de segunda en la primera fila. Si el usuario ingresa a través de la página Web, debe dar clic en el botón *Ubicación Actual*.
5. Solicitar la URL respectiva al SC para hacer la consulta a MapServer.
6. Hacer la consulta a MapServer para obtener el mapa.
7. Desplegar en la pantalla un mapa digital con la posición actual del o de los vehículos.

Post Condiciones: Se permite al usuario manipular la imagen haciendo zoom (acercamiento/alejamiento) y otras consultas relacionadas con la información de los vehículos.

Flujos Alternativos:

-  Realizar acercamiento/alejamiento a la imagen desplazada.
-  Solicitar información de los vehículos desplegados (para que se habilite esta opción es necesario que seleccionar un vehículo a través del cursor).
-  Enviar un SMS a los vehículos desplegados.
-  Volver al menú inicial.
-  Cancelar la operación.

Excepciones:

- 🌐 *NullPointerException*: Se lanza cuando la Alerta o el Displayable son nulos.
- 🌐 *IllegalArgumentException*: Si el siguiente Displayable es un Alert.
- 🌐 *IllegalArgumentException*: Si el tamaño máximo del campo de texto es cero o negativo. También es lanzada en el caso de que los valores de los parámetros o el texto son inválidos o por que la longitud de la cadena ingresada excede la capacidad máxima especificada.
- 🌐 *NullPointerException*: Lanzada cuando la etiqueta de los Command son nulas.
- 🌐 *IllegalArgumentException*: Si el commandType es un tipo inválido.
- 🌐 *IllegalArgumentException*: Se lanza cuando se pasan parámetros inválidos al abrir la conexión HTTP.
- 🌐 *ConnectionNotFoundException*: Si la petición de conexión no puede realizarse o si el tipo de protocolo especificado no existe.
- 🌐 *IOException*: Si ocurre alguna otra clase de error tipo I/O.
- 🌐 *NullPointerException*: Se lanza en el momento de crear una imagen y la fuente es nula o no existe.
- 🌐 *NullPointerException*: Lanzada si el LayerManager es nulo en el momento de añadir elementos.
- 🌐 *NullPointerException*: Lanzada cuando se crea un Sprite y alguno de sus parámetros es nulo.

Recursos Especiales Utilizados: Conexión al SC, Conexión a MapServer.

GUI's Relacionadas:

- 🌐 Acceso a través del dispositivo móvil: menuInicial, menuPrincipal, posicionActual.
- 🌐 Acceso a través de la página Web: menuInicial.html, menuPrincipal.html, posicionActual.html



1.1.7 Caso de Uso No. 7: enviarMensajes

Iniciador: Usuario Remoto.

Precondición: Seleccionar uno o más vehículos de la lista mencionada en el caso de uso No. 4.

Propósito: Dar opción al usuario de enviar SMS a los vehículos que tiene acceso.

Resumen: Este caso de uso implementa el API de JAVA Wireless Message API (WMA) el cual controla un puerto determinado para el envío de mensajes. Cuando el usuario selecciona los vehículos de la lista desplegada anteriormente, el sistema solicita el número del MODEM para conocer el destinatario. Este caso de uso sólo puede ser inicializado por el UR que accede a través del teléfono celular.

Flujo Principal:

1. Realizar los pasos de validación descritos en el caso de uso No. 1.
2. Escoger una opción del listado de vehículos.
3. Solicitar el número del MODEM de los vehículos seleccionados.
4. Desplegar el menú principal.
5. Seleccionar la opción de *Mensajes* que se encuentra de segunda en la segunda fila.
6. Desplegar la pantalla que le permita al usuario escribir el SMS.
7. Seleccionar la opción *enviar*.
8. Crear el cliente que es el encargado de implementar el API de WMA.
9. Reportar al cliente a que números se va a enviar el SMS.
10. Enviar el SMS a los destinos seleccionados.
11. Reportar al UR si los SMS han sido o no enviados correctamente.

Post Condiciones: Ninguna



Flujos Alternativos:

- 🌐 Volver al menú inicial.
- 🌐 Cancelar la operación.

Excepciones:

- 🌐 *java.lang.IllegalArgumentException*: Si el tipo de parámetros no es igual a TEXT_MESSAGE o BINARY_MESSAGE. Es lanzada por el MessageConnection al crear un nuevo mensaje.
- 🌐 *java.io.IOException*: Si el mensaje no puede ser enviado o por fallas de la red o si la conexión está cerrada. Lanzada por MessageConnection.
- 🌐 *java.lang.IllegalArgumentException*: Si el mensaje está incompleto o contiene información inválida. Esta excepción es lanzada por MessageConnection si la carga útil del mensaje excede la longitud máxima.
- 🌐 *java.io.InterruptedIOException*: Si el objeto Connection es cerrado durante el envío del mensaje o cuando expira el *timeout* mientras se está tratando de enviar el mensaje. Lanzada por MessageConnection.
- 🌐 *java.lang.NullPointerException*: Si hay algún parámetro vacío. Lanzada por MessageConnection.
- 🌐 *java.lang.SecurityException*: Si la aplicación no tiene permiso de enviar SMS. Lanzada por MessageConnection.
- 🌐 *java.io.IOException*: Si ocurre un error de I/O.

Recursos Especiales Utilizados: WMA

GUI's Relacionadas:

- 🌐 menuInicial, menuPrincipal, mensajes, cliente.

1.1.8 Caso de Uso No. 8: *gestionarUsuario*

Iniciador: Administrador.

Precondición: El usuario debe tener privilegios de administrador.

Propósito: Dar al administrador la opción de ingresar y eliminar usuarios del sistema.

Resumen: Este caso de uso es el encargado de gestionar todo lo relacionado con los usuarios. Solo se puede acceder a través de la página Web.

Flujo Principal:

1. Validarse al sistema con permisos de administrador.
2. Desplegar el menú de administrador.
3. Escoger la opción pertinente según la operación que necesite hacer.
4. Desplegar la pantalla correspondiente.
5. Ingresar los datos que sean solicitados en los campos correspondientes.
6. Desplegar los resultados de la operación al administrador.

Post Condiciones: Ninguna

Flujos Alternativos:

-  Volver al menú inicial.
-  Cancelar la operación.

Excepciones:

-  *IllegalArgumentException*: Se lanza cuando se introducen parámetros inválidos al abrir la conexión HTTP.



- 🌐 *ConnectionNotFoundException*: Si la petición de conexión no puede realizarse o si el tipo de protocolo especificado no existe.
- 🌐 *IOException*: Si ocurre alguna otra clase de error tipo I/O.

Recursos Especiales Utilizados: Conexión al SC.

GUI's Relacionadas:

- 🌐 Acceso a través de la página Web: menuAdministrador.html, ingresarUsuario.html, eliminarUsuario.html, listarUsuario.html

1.2 DIAGRAMA DE CLASES

Las clases implementadas por paquete son en el UR son:

- 🌐 *Control*: controlSirio
- 🌐 *Vista*: presentacion, validación, menuInicial, listaVehiculos, actualizarDatoUsuario, cambiarPasswd, autores, menuPrincipal, informaciónVehiculo, enviarSMS, ayuda
- 🌐 *Modelo*: beanUsuario, beanVehiculo, beanMensaje, conexionImagen, conexionSirio
- 🌐 *WMA*: clienteSMS, servidorSMS
- 🌐 *GUI*: ubicaciónActual, spriteCursor.

En la Figura 2 se ilustra el diagrama de clases del UR.

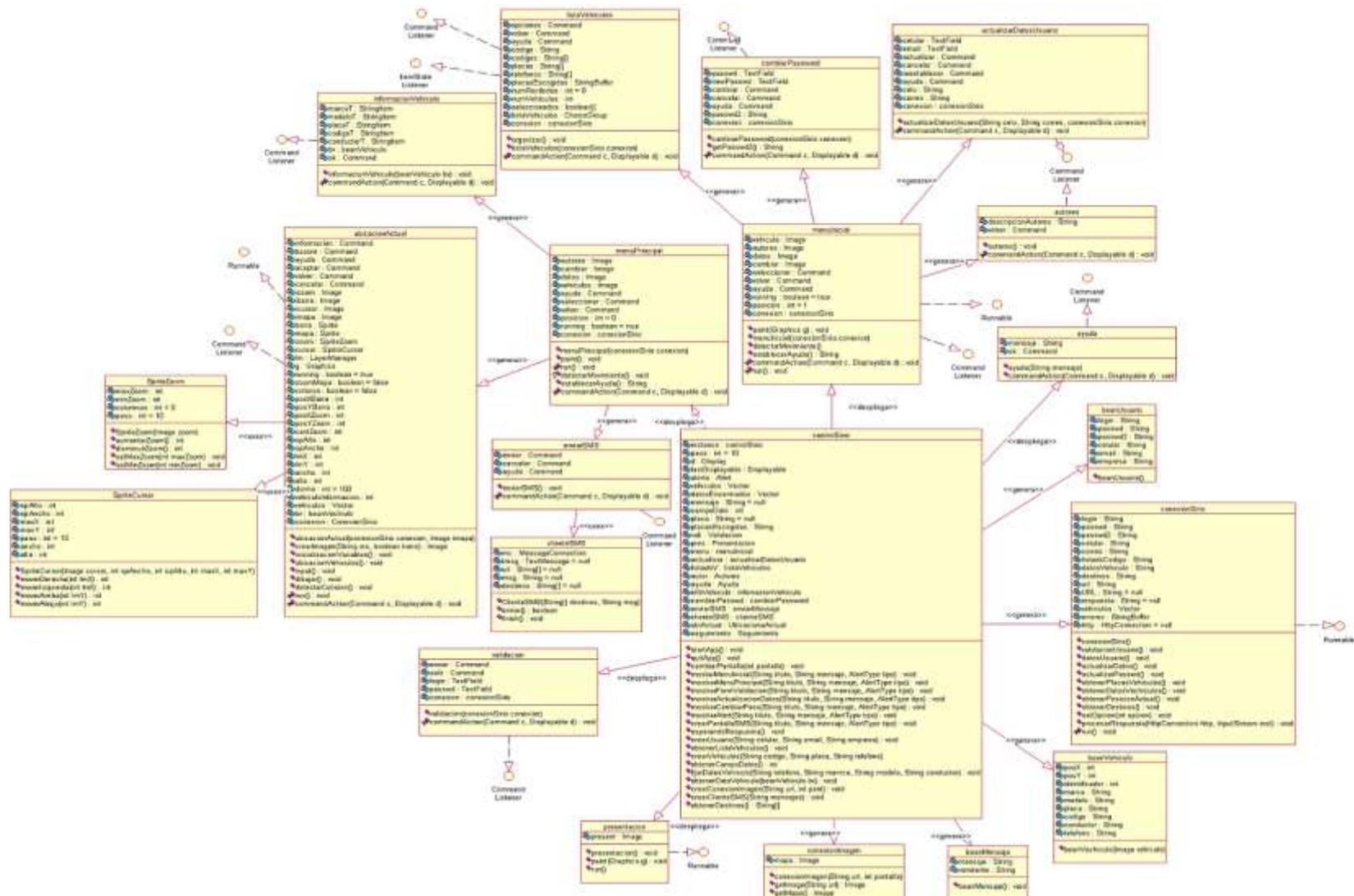


Figura 2. Diagrama de Clases del UR

1.3.1.2 Validación Incorrecta

En la Figura 4 se ilustra el diagrama de secuencia utilizado cuando la validación de usuario es incorrecta, o cuando se ha presentado un error en este proceso.

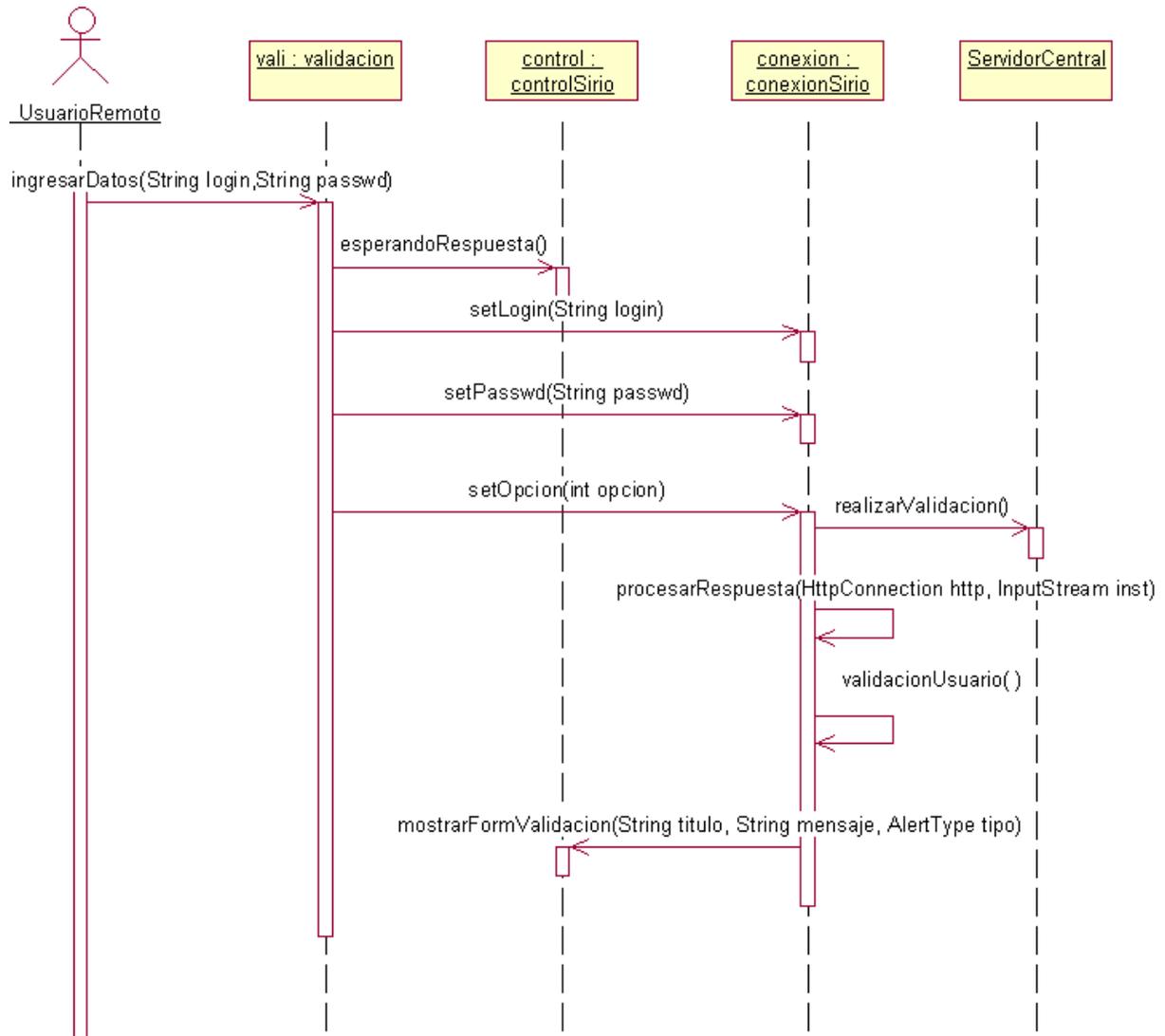


Figura 4. Diagrama de Secuencia para la Validación de Usuario incorrecta

1.3.2 Cambiar Contraseña

1.3.2.1 Cambio de contraseña exitoso

En la Figura 5 se ilustra el diagrama de secuencia utilizado para cuando el cambio de contraseña de usuario es exitoso.

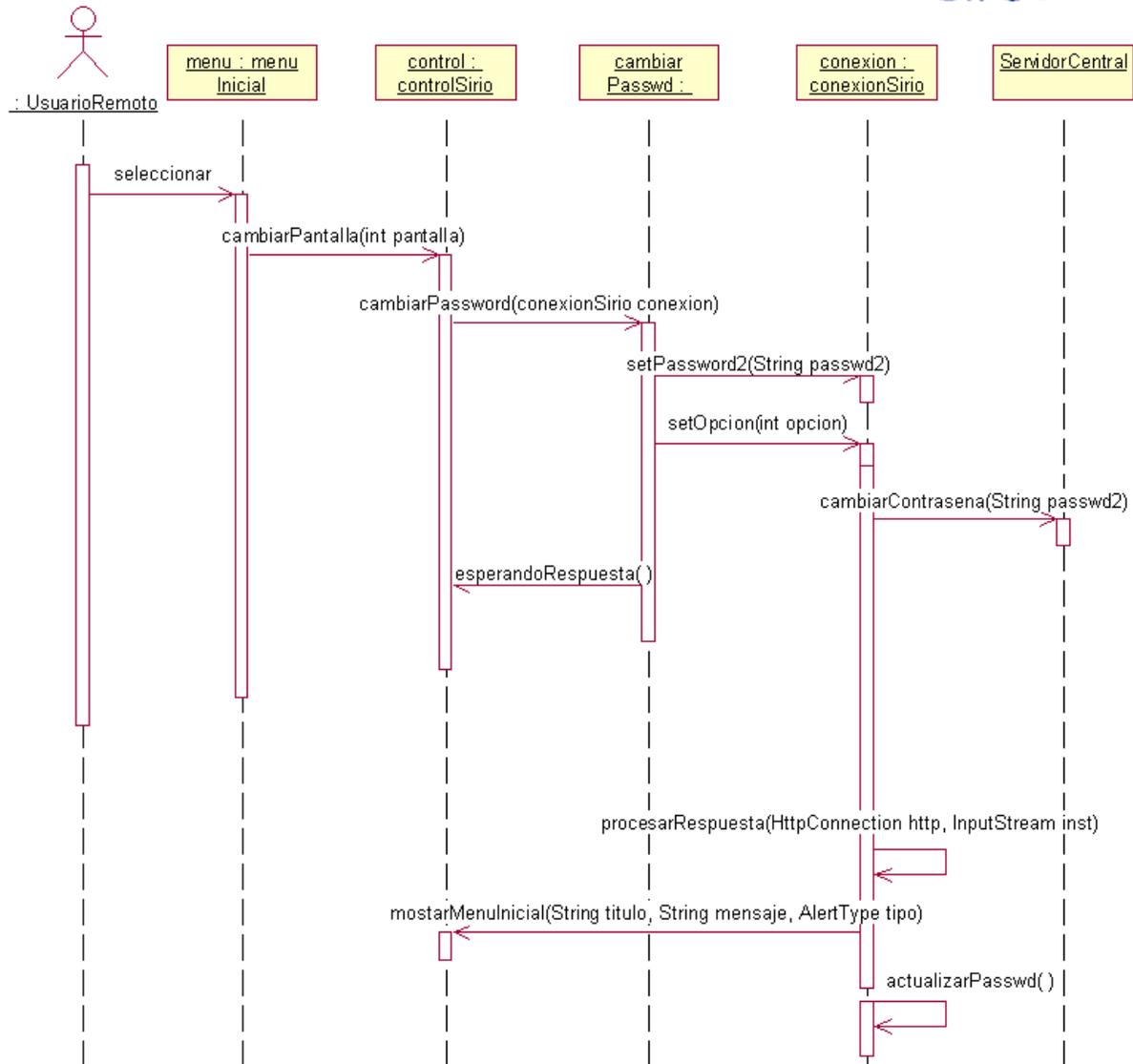


Figura 5. Diagrama de Secuencia de Cambio de Contraseña exitoso

1.3.2.2 Cambio de contraseña fallido

En la Figura 6 se ilustra el diagrama de secuencia utilizado para cuando el cambio de contraseña de usuario se ha presentado un error o es fallido.

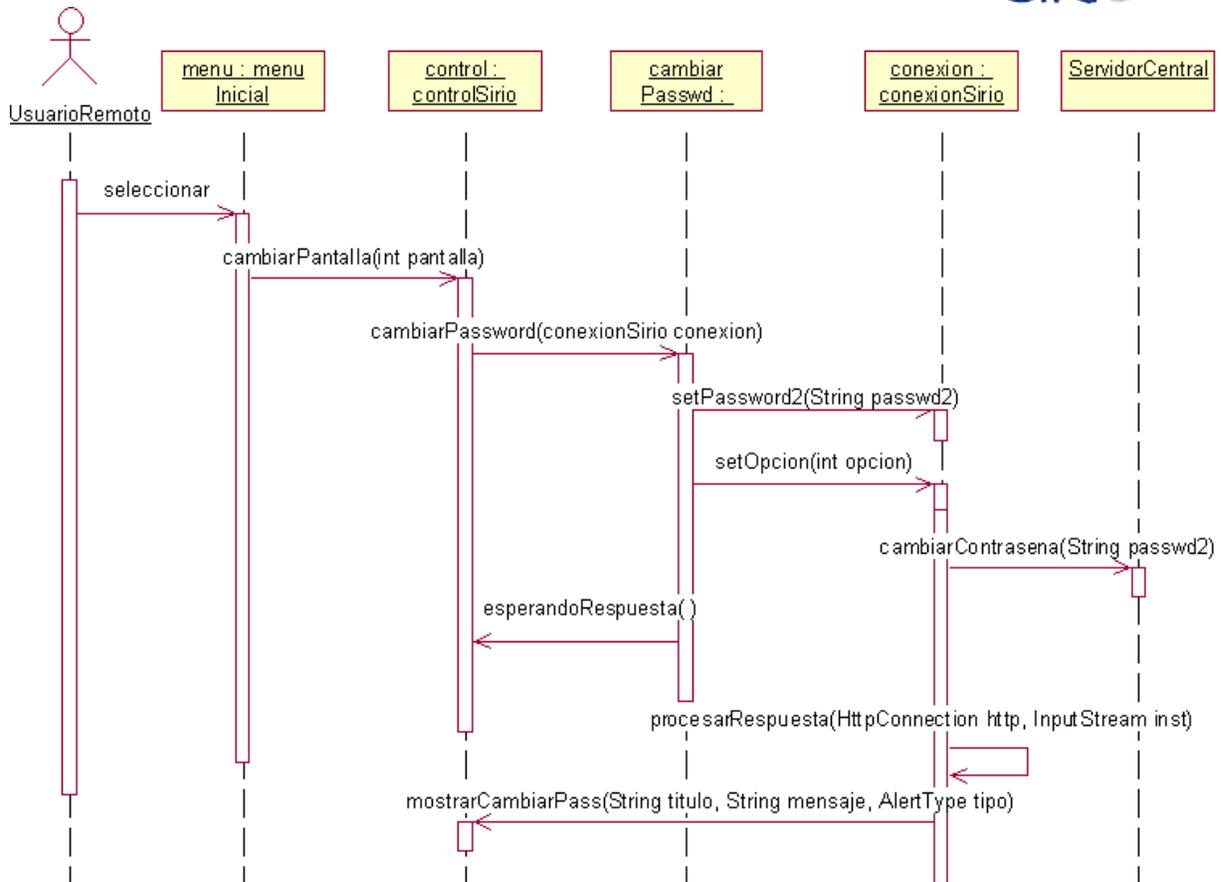


Figura 6. Diagrama de Secuencia de Cambio de Contraseña fallido

1.3.3 Cambiar Datos Usuario

1.3.3.1 Cambio de datos de usuario exitoso

En la Figura 7 se ilustra el diagrama de secuencia utilizado para cuando el cambio de datos de usuario es exitoso.

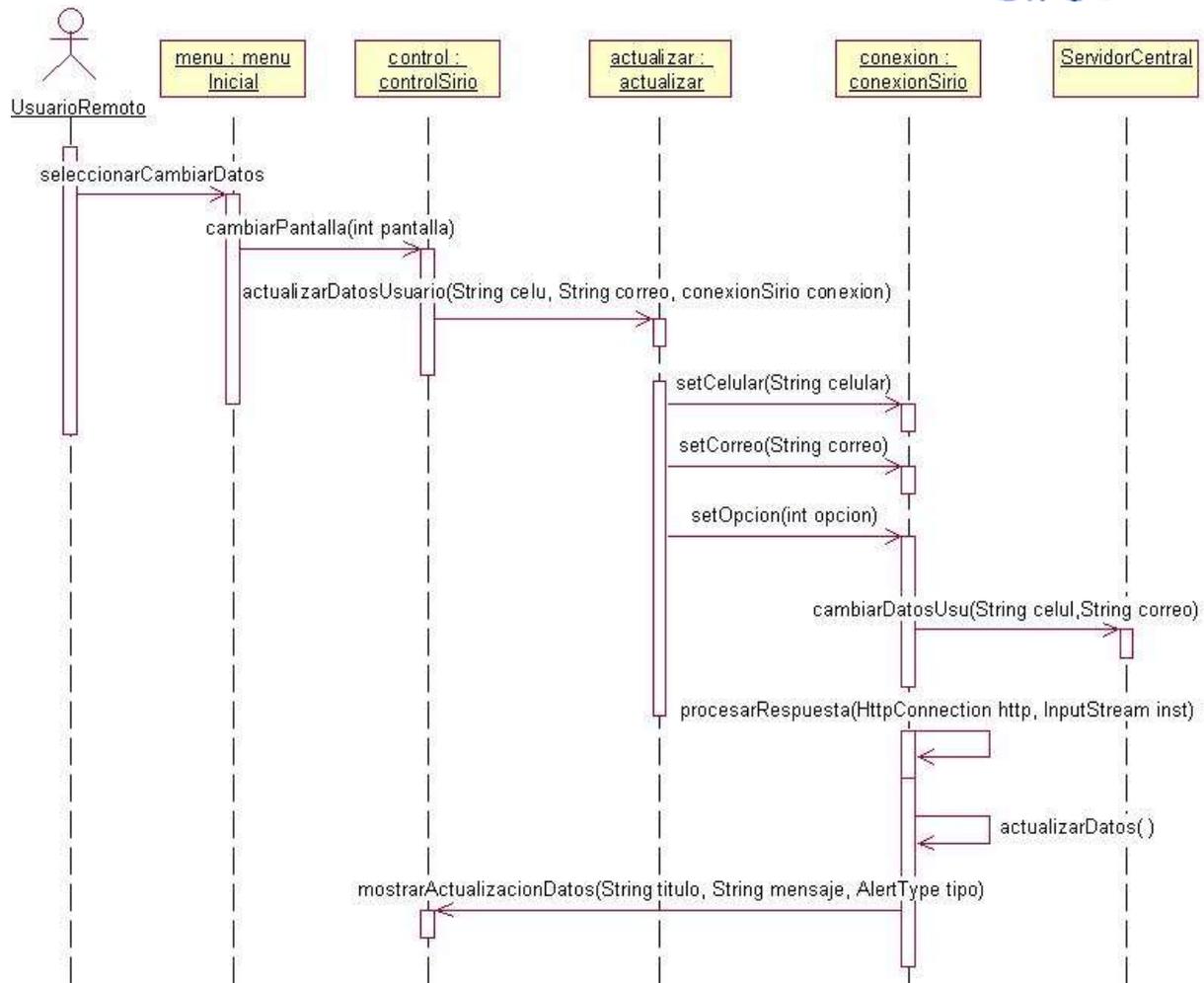


Figura 7. Diagrama de Secuencia para el cambio de datos del usuario exitoso

1.3.3.2 Cambio de datos de usuario fallido

En la Figura 8 se ilustra el diagrama de secuencia utilizado para cuando el cambio de datos de usuario ha presentado un error o es fallido.

1.3.4 Obtener Listado de Vehículos

En la Figura 9 se ilustra el diagrama de secuencia utilizado cuando se obtiene exitosamente el listado de los vehículos del usuario.

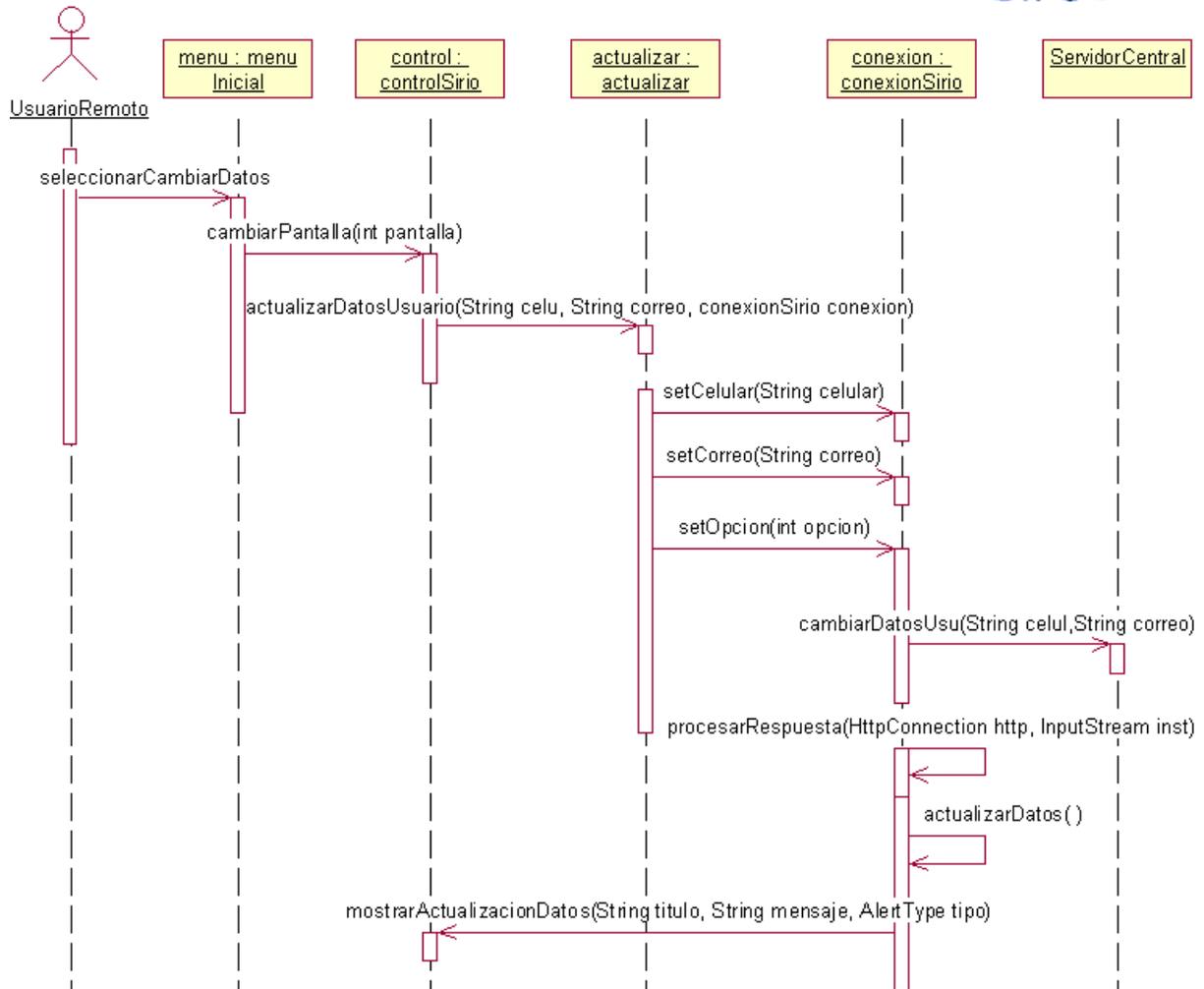


Figura 8. Diagrama de Secuencia de Actualización Datos de Usuario fallido

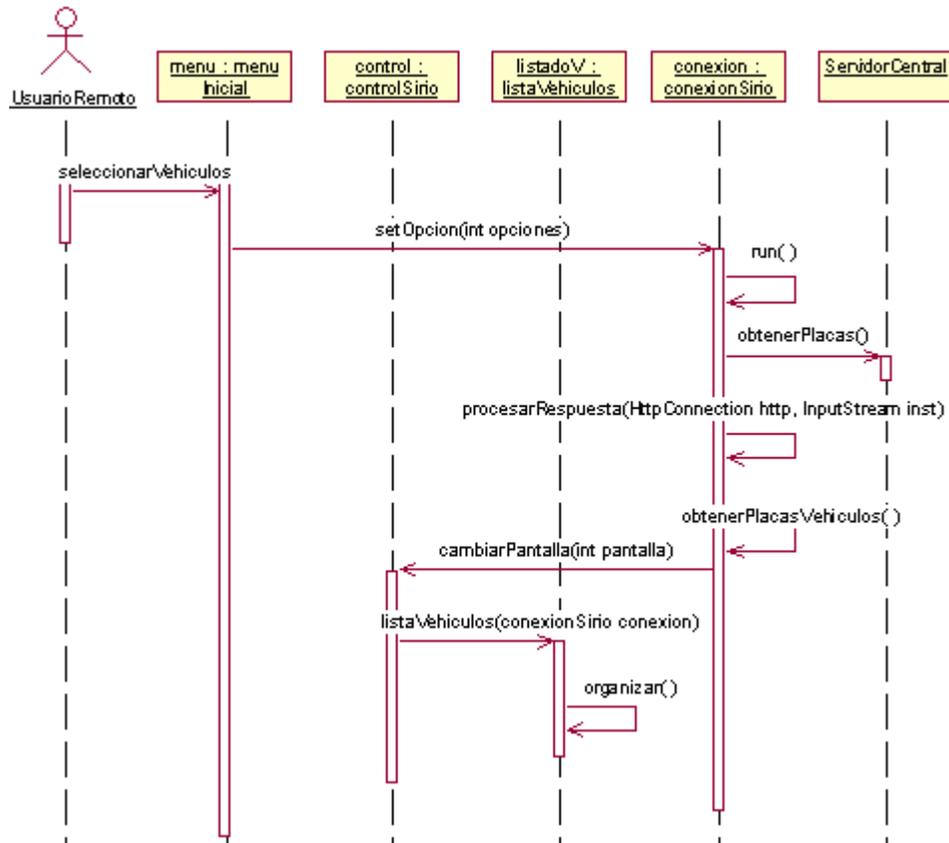


Figura 9. Diagrama de Secuencia para obtención exitosa de la lista de vehículos

1.3.5 Obtener Datos Vehículos

1.3.5.1 Obtención de datos de vehículos exitoso

En la Figura 10 se ilustra el diagrama de secuencia utilizado cuando se obtiene exitosamente los datos del vehículo seleccionado.

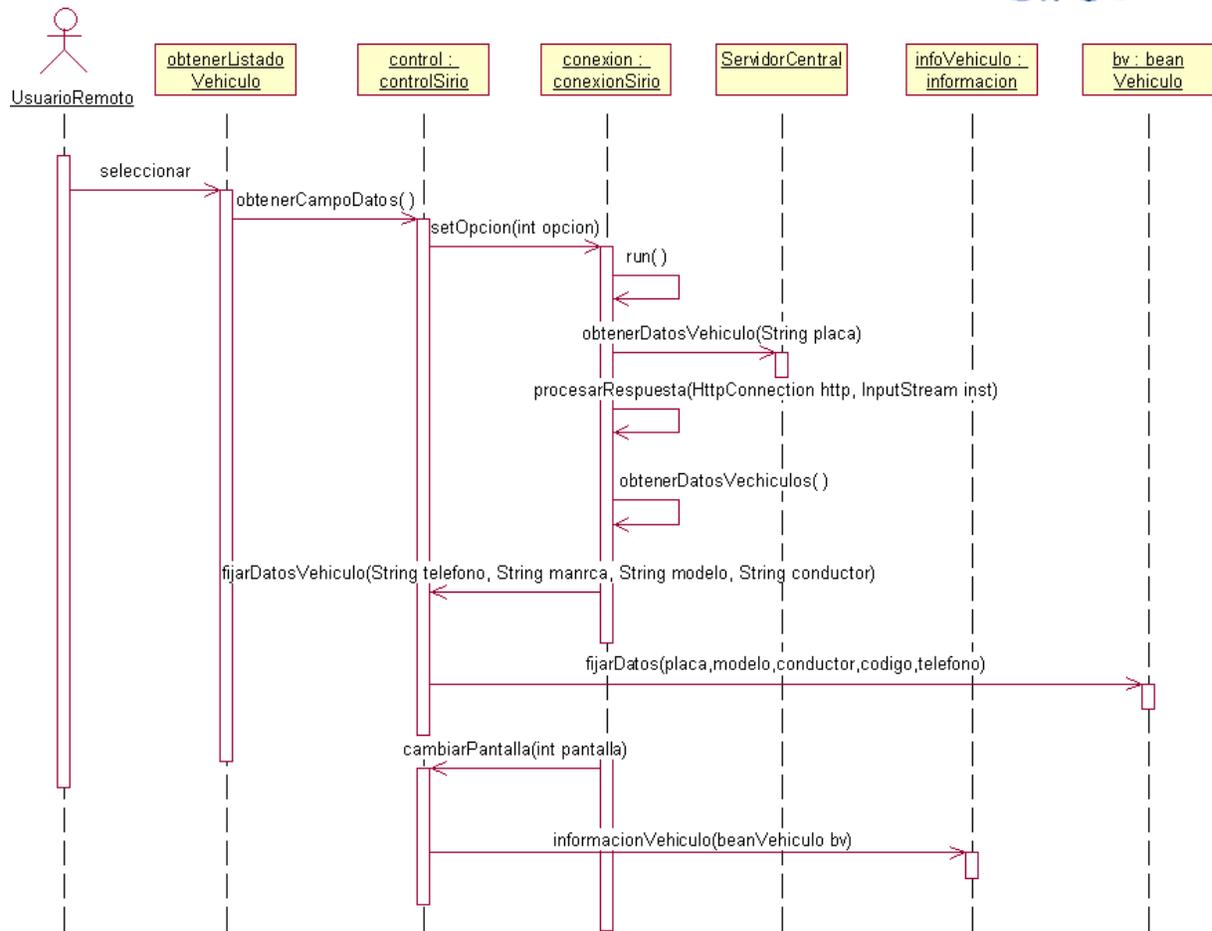


Figura 10. Diagrama de Secuencia para obtención exitosa de la lista de vehículos

1.3.6 Obtener Posición Actual

1.3.6.1 Obtención de posición actual exitosa

En la Figura 11 se ilustra el diagrama de secuencia utilizado cuando se obtiene exitosamente la posición actual del vehículo.

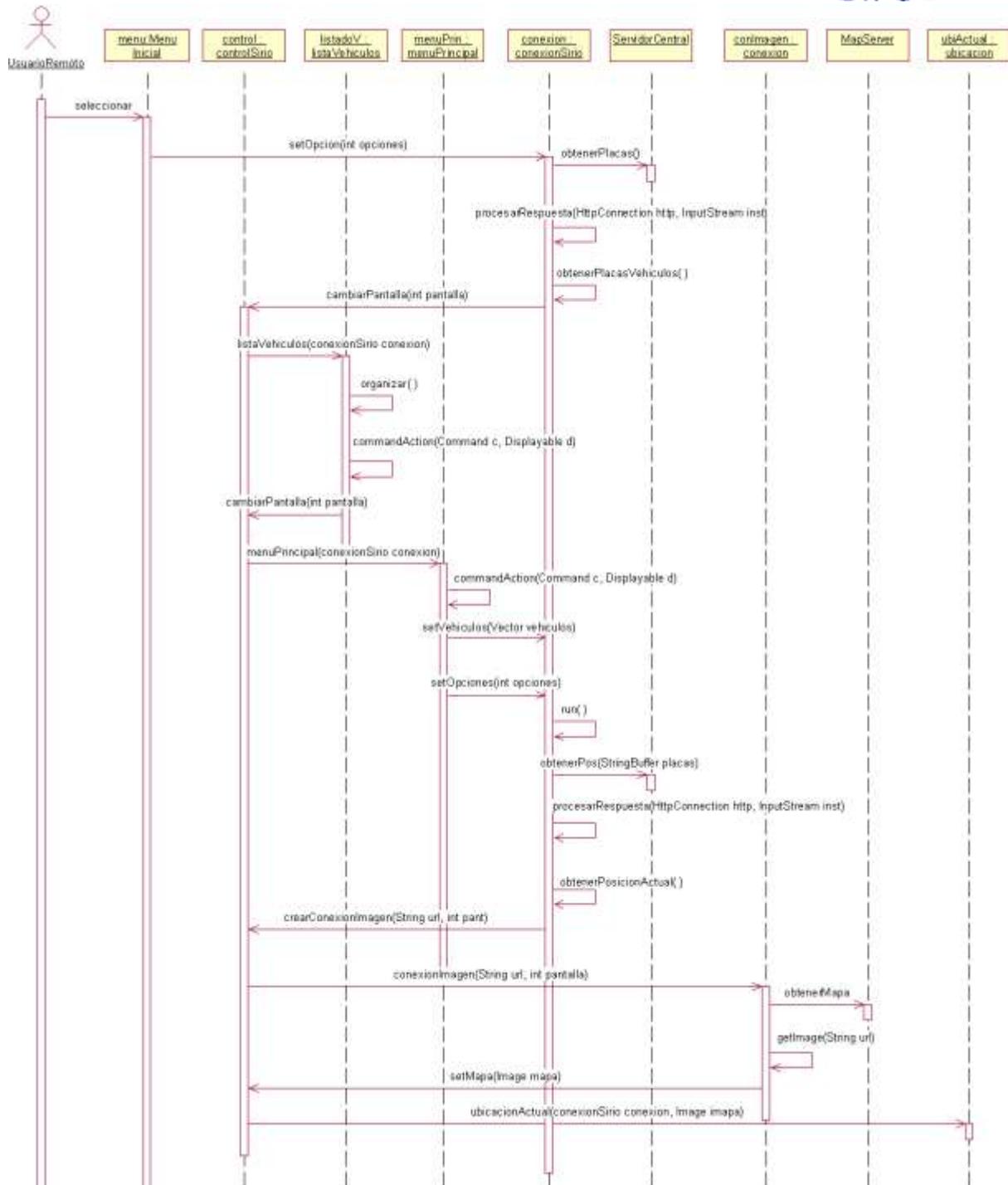


Figura 11. Diagrama de Secuencia para la obtención exitosa de la posición actual del vehículo

1.3.7 Enviar Mensajes SMS

1.3.7.1 Envío de Mensajes

En la Figura 12 se ilustra el diagrama de secuencia utilizado cuando se envía mensajes SMS a los destinos seleccionados.

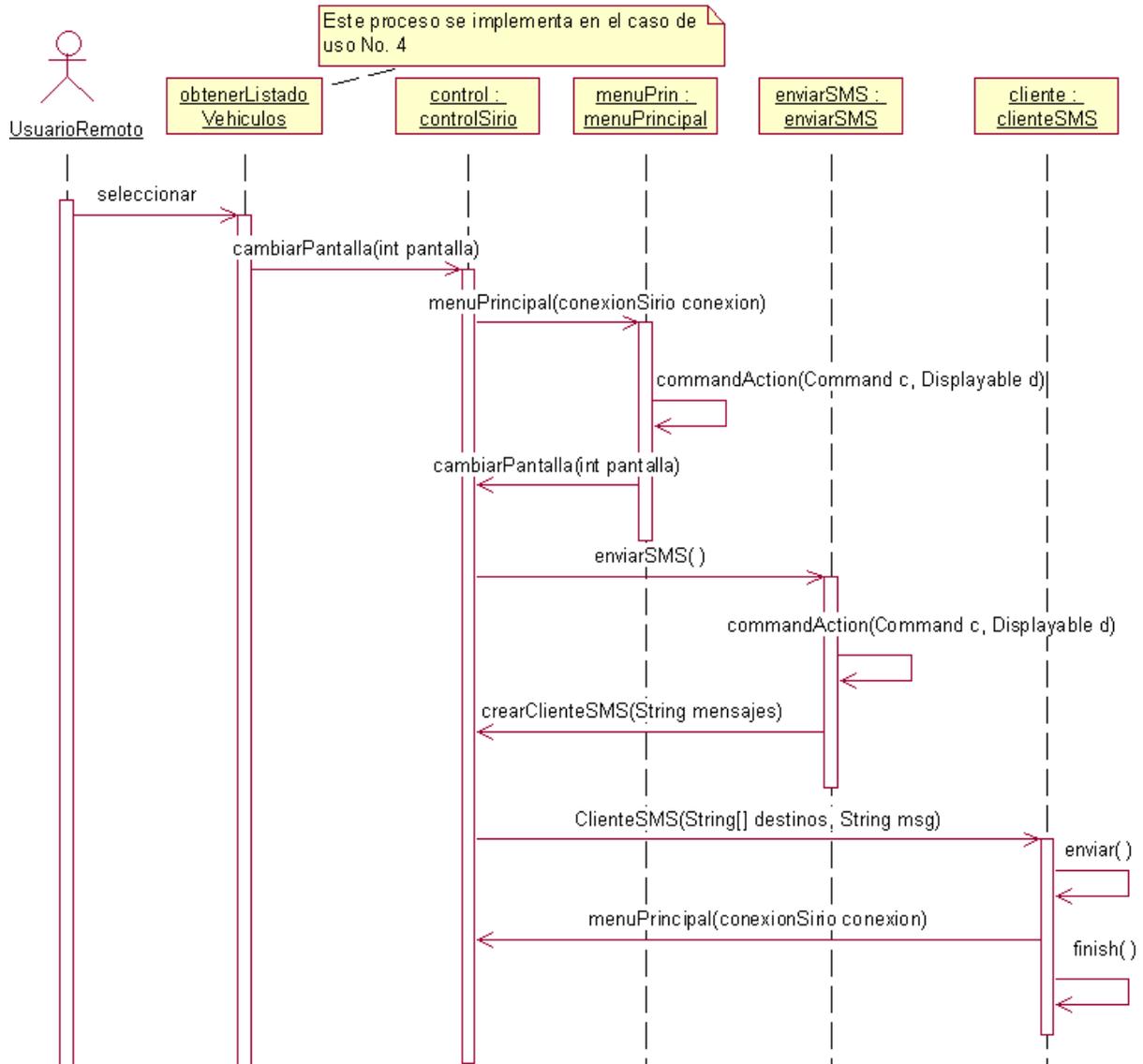


Figura 12. Diagrama de Secuencia para el envío de mensajes SMS

TABLA DE CONTENIDO

1	IMPLEMENTACION DETALLADA UR	1
1.1	Diagrama de Casos de Uso	1
1.1.1	Caso de Uso No. 1: Validar	2
1.1.2	Caso de Uso No. 2: cambiarContraseña	3
1.1.3	Caso de Uso No. 3: cambiarDatos	6
1.1.4	Caso de Uso No. 4: obtenerListadoVehiculos	8
1.1.5	Caso de Uso No. 5: obtenerDatosVehiculos	9
1.1.6	Caso de Uso No. 6: obtenerPosicion	11
1.1.7	Caso de Uso No. 7: enviarMensajes	14
1.1.8	Caso de Uso No. 8: gestionarUsuario	16
1.2	Diagrama de Clases	17
1.3	Diagramas de Secuencia	19
1.3.1	Validación de Usuario	19
1.3.1.1	Validación Correcta	19
1.3.1.2	Validación Incorrecta	20
1.3.2	Cambiar Contraseña	20
1.3.2.1	Cambio de contraseña exitoso	20
1.3.2.2	Cambio de contraseña fallido	21
1.3.3	Cambiar Datos Usuario	22
1.3.3.1	Cambio de datos de usuario exitoso	22
1.3.3.2	Cambio de datos de usuario fallido	23
1.3.4	Obtener Listado de Vehículos	23
1.3.5	Obtener Datos Vehículos	25
1.3.5.1	Obtención de datos de vehículos exitoso	25
1.3.6	Obtener Posición Actual	26

1.3.6.1	Obtención de posición actual exitosa	26
1.3.7	Enviar Mensajes SMS	28
1.3.7.1	Envío de Mensajes	28

INDICE DE FIGURAS

Figura 1. Diagrama de Casos de Usos del UR	1
Figura 2. Diagrama de Clases del UR	18
Figura 3. Diagrama de Secuencia de Validación Correcta de Usuario.	19
Figura 4. Diagrama de Secuencia para la Validación de Usuario incorrecta	20
Figura 5. Diagrama de Secuencia de Cambio de Contraseña exitoso	21
Figura 6. Diagrama de Secuencia de Cambio de Contraseña fallido	22
Figura 7. Diagrama de Secuencia para el cambio de datos del usuario exitoso	23
Figura 8. Diagrama de Secuencia de Actualización Datos de Usuario fallido	24
Figura 9. Diagrama de Secuencia para obtención exitosa de la lista de vehículos	25
Figura 10. Diagrama de Secuencia para obtención exitosa de la lista de vehículos	26
Figura 11. Diagrama de Secuencia para la obtención exitosa de la posición actual del vehículo	27
Figura 12. Diagrama de Secuencia para el envío de mensajes SMS	28

SIRIO

Sistema de Localización Operado bajo red Celular



María Isabel Caicedo Castro
Rubén Darío Rincón Giraldo

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Línea: Gestión Integrada de Sistemas de Telecomunicaciones
Área de Sistemas de Comunicaciones Móviles e Inalámbricas
Popayán, Enero 2006

SIRIO

Sistema de Localización Operado bajo red Celular



ANEXO B

IMPLEMENTACIÓN DETALLADA DEL UR

María Isabel Caicedo Castro
Rubén Darío Rincón Giraldo

Director: Víctor Manuel Quintero Flórez

Universidad del Cauca

Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telecomunicaciones
Línea: Gestión Integrada de Sistemas de Telecomunicaciones
Área de Sistemas de Comunicaciones Móviles e Inalámbricas
Popayán, Enero 2006