



ÍNDICE

1. Tercero Incremento	2
1.1. Definición	2
1.2. Diagrama de Entidad Relación	2
1.3. Diagrama de Clases	3
1.4. Diagramas de Interacción de Objetos	5
1.4.1. Lista de Casos de Usos	5
1.4.2. Descripción de Casos de Usos	5
1.4.3. Especificación de Escenarios	6
1.4.4. Diagramas de Secuencia	9
1.5. Invariantes, Precondiciones y Poscondiciones	12

DISEÑO DETALLADO

1. Tercer Incremento

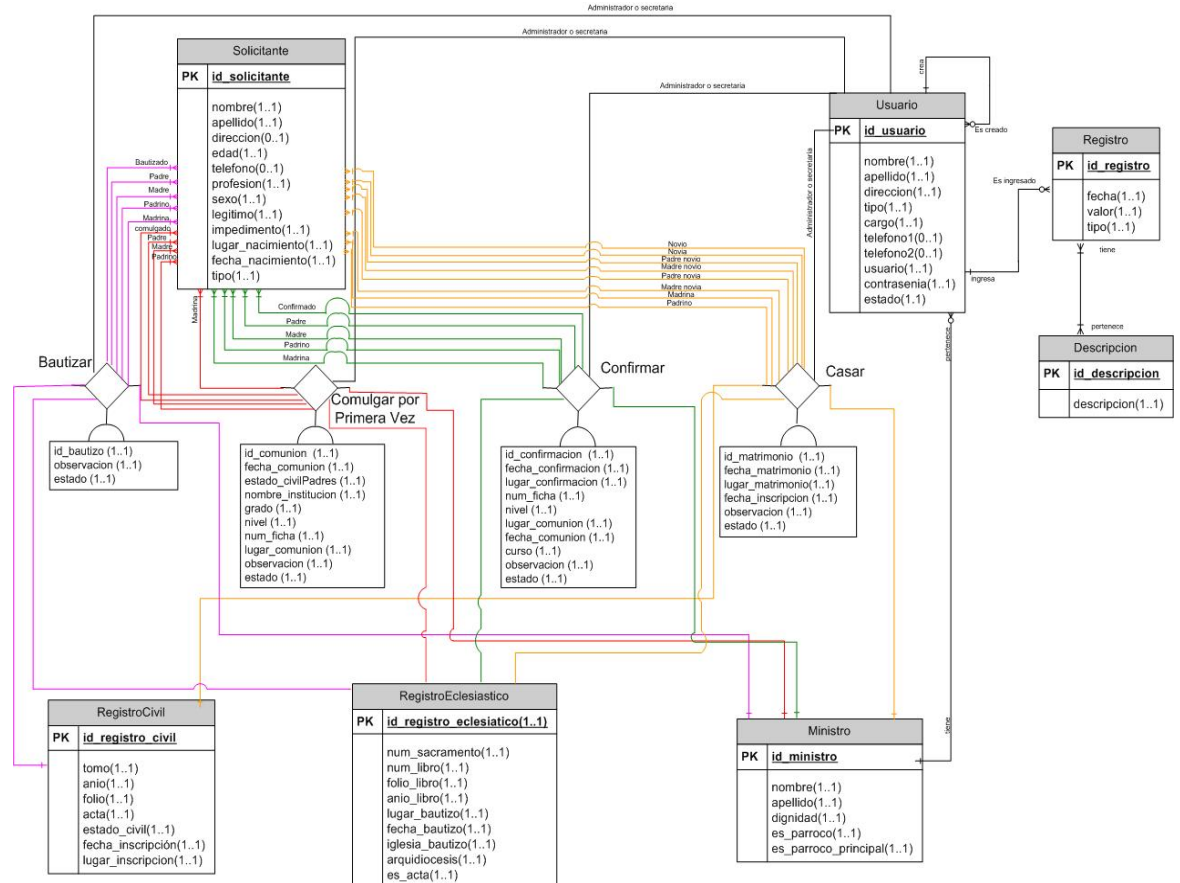
1.1. Definición

Con el tercer incremento del *Sistema de Servicios Parroquiales (SSP)* se implementará la siguiente tarea:

- ◆ Funcionalidad añadida pertenecientes el módulo de documentación esto es, el registro de ingresos y egresos de la parroquia.
- ◆ Modificación y eliminación de usuario perteneciente al módulo de administración.

1.2. Diagrama de Entidad – Relación

El Modelo Entidad-Relación del *Sistema de Servicios Parroquiales (SSP)* es el siguiente:



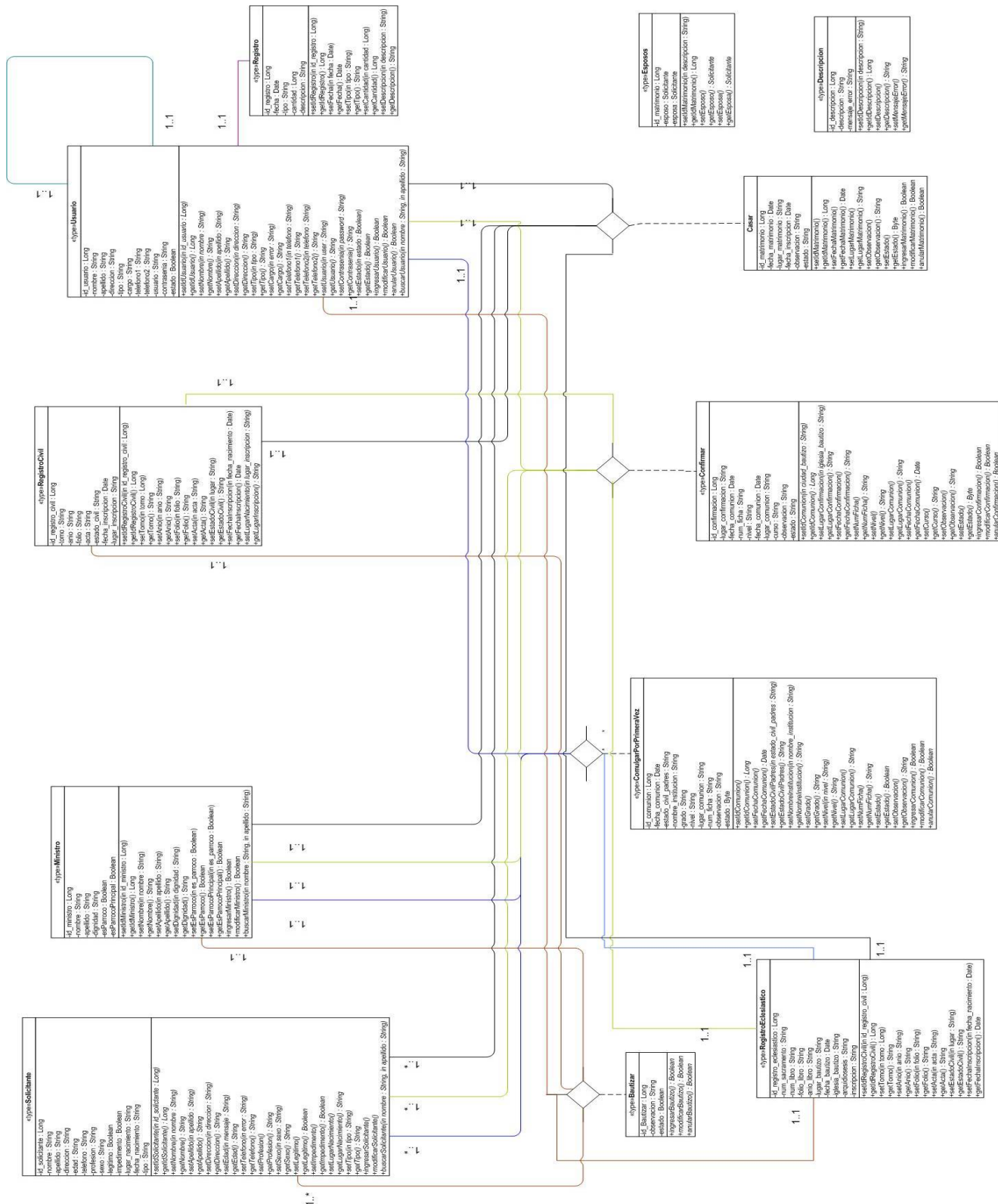


1.3. Diagrama de Clases

Las clases que van a intervenir en el Tercer Incremento son las siguientes:

- ◆ Registro,
- ◆ Usuario.

A continuación se presenta el Modelo de Clases del Módulo de Documentación:



1.4. Diagramas de Interacción de Objetos

1.4.1 Listas de Casos de Usos

A continuación se listarán los casos de usos ha ser implementados:

1.4 Diagrama de Interacción de Objetos

1.4.1 Lista de Casos de Usos

Módulo de Documentación funcionalidad añadida:

1. Ingreso de Registro.

Módulo de Administración:

2. Modificar Usuario.
3. Anular Usuario.

1.4.2 Descripción de Casos de Usos

Nombre de caso de uso:	1. Ingresar Registro
Actores participantes:	Inicializado por el usuario (contador)
Flujo de eventos:	<ol style="list-style-type: none"> 1. Escoger la opción Ingresar registro. 2. Ingresar los datos respectivos para el ingreso del registro. 3. Hacer clic en guardar para que los datos se almacenen en la base de datos.
Condición de entrada:	<ul style="list-style-type: none"> • El usuario debe estar ingresado en la base como contador
Condición de salida:	<ol style="list-style-type: none"> 1.1. Ingreso exitoso de registro, 1.2. Ingreso no exitoso debido a que ya existe. 1.3. Ingreso no exitoso de registro debido a errores internos.
Requerimientos de calidad:	<ul style="list-style-type: none"> • Este procesamiento debe ser instantáneo.

Nombre de caso de uso:	2. Modificar usuario
Actores participantes:	Inicializado por el usuario (administrador)
Flujo de eventos:	<ol style="list-style-type: none"> 1. Escoger la opción Modificar usuario. 2. Ingresar datos modificados del usuario existente. 3. Hacer clic en aceptar para que los datos se actualicen en la base de datos.
Condición de entrada:	<ul style="list-style-type: none"> • El usuario debe estar registrado en la Base de Datos como administrador.
Condición de salida:	<p>Modificación exitosa de usuario.</p> <p>Modificación no exitosa de usuario debido a errores internos.</p>
Requerimientos de calidad:	<ul style="list-style-type: none"> • Este procesamiento debe ser instantáneo.

Nombre de caso de uso:	3. Anular usuario
Actores participantes:	Inicializado por el usuario (administrador)
Flujo de eventos:	<ol style="list-style-type: none"> 1. Escoger la opción Anular usuario. 2. Cambiar el estado del usuario de habilitado a deshabilitado. 3. Hacer clic en aceptar para que los datos se actualicen en la base de datos.
Condición de entrada:	<ul style="list-style-type: none"> • El usuario debe estar registrado en la Base de Datos como administrador.
Condición de salida:	<ol style="list-style-type: none"> 4.1. Anulación exitosa del usuario, 4.2. Anulación no exitosa del usuario debido a errores internos.
Requerimientos de calidad:	<ul style="list-style-type: none"> • Este procesamiento debe ser instantáneo.

1.4.3 Especificación de Escenarios

Nombre de Caso de uso:	1. Ingresar registro
Nombre de Escenario:	1.1. Ingreso exitoso de registro
Instancias de actores participantes:	contador:Usuario conexion:ConexionDB
Flujo de eventos:	<ol style="list-style-type: none"> 1. El contador escoge la opción registro de contabilidad, 2. El contador ingresa los datos pertinentes al registro, 3. El usuario da clic en aceptar para que los datos se ingresen en la base. 4. conexion abre la base de datos y le da el manejador de la base a la clase registro, 5. Se ingresa el registro de contabilidad en la base, 6. Se genera un mensaje de éxito de registro de la contabilidad.

Nombre de Caso de uso:	1. Ingresar registro
Nombre de Escenario:	1.2. Ingreso no exitoso de registro debido a que ya existe.
Instancias de actores participantes:	contador:Usuario conexion:ConexionDB
Flujo de eventos:	<ol style="list-style-type: none"> 1. El contador escoge la opción registro de contabilidad, 2. El contador ingresa los datos pertinentes al registro, 3. El usuario da clic en aceptar para que los datos se ingresen en la base. 4. conexion abre la base de datos y le da el manejador de la base a la clase registro, 5. Se genera un mensaje de error de registro de la contabilidad debido a que ya existe.

Nombre de Caso de uso:	1. Ingresar registro
-------------------------------	-----------------------------

Nombre de Escenario:	1.3. Ingreso no exitoso de registro debido a errores internos.
Instancias de actores participantes:	contador:Usuario conexion:ConexionDB
Flujo de eventos:	<ol style="list-style-type: none"> 1. El contador escoge la opción registro de contabilidad, 2. El contador ingresa los datos pertinentes al registro, 3. El usuario da clic en aceptar para que los datos se ingresen en la base. 4. conexion abre la base de datos y le da el manejador de la base a la clase registro, 5. Se genera un mensaje no exitoso de registro de la contabilidad debido a errores internos.

Nombre de Caso de uso:	2. Modificar Usuario
Nombre de Escenario:	2.1. Modificación exitosa de usuario
Instancias de actores participantes:	administrador:Usuario conexion:ConexionBD
Flujo de eventos:	<ol style="list-style-type: none"> 1. El administrador escoge la opción de modificación de usuario, 2. El administrador ingresa el nombre y el apellido del usuario a modificar. 3. conexion abre la base de datos y le da el manejador de la base a la clase Usuario, 4. la clase Usuario busca una lista de posibles nombres en la Base, 5. Se presenta en una interfaz una lista de usuarios que coinciden con el nombre y apellido a buscar. 6. El administrador escoge el usuario a modificar. 7. Se presenta en una interfaz los datos del usuario a modificar. 8. El administrador modifica los datos pertinentes del usuario del sistema a modificar. 9. El administrador da clic en aceptar para que los datos se actualicen en la base. 10. la clase Usuario actualiza los datos en la Base, 11. la clase Usuario cierra la Base, 12. Mostrar un mensaje de modificación exitosa.

Nombre de Caso de uso:	2. Modificar Usuario
Nombre de Escenario:	2.2. Modificación no exitosa de Usuario debido a errores internos.
Instancias de actores participantes:	administrador:Usuario conexion:ConexionBD
Flujo de eventos:	<ol style="list-style-type: none"> 1. El administrador escoge la opción de modificación de usuario, 2. El administrador ingresa el nombre y el apellido del usuario a modificar. 3. conexion abre la base de datos y le da el manejador de la base a la clase Usuario, 4. la clase Usuario busca una lista de posibles

	<p>nombres en la Base,</p> <ol style="list-style-type: none"> Se presenta en una interfaz una lista de usuarios que coinciden con el nombre y apellido a buscar. El administrador escoge el usuario a modificar. Se presenta en una interfaz los datos del usuario a modificar. El administrador modifica los datos pertinentes del usuario del sistema a modificar. El administrador da clic en aceptar para que los datos se actualicen en la base. existe un error en la clase Usuario para modificar, la clase usuario controla el error, Mostrar un mensaje de modificación no exitosa debido a errores internos.
--	--

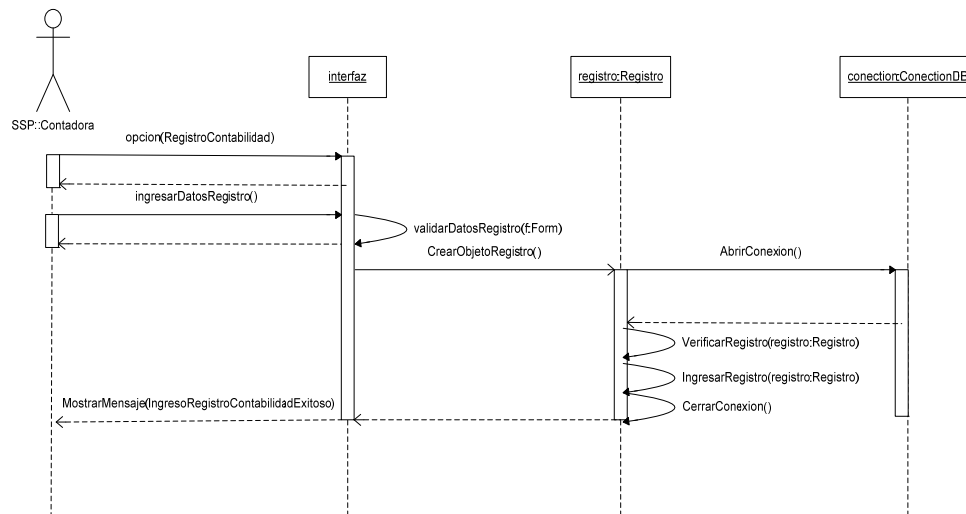
Nombre de Caso de uso:	3. Anular Usuario
Nombre de Escenario:	3.1. Anulación exitosa de usuario
Instancias de actores participantes:	administrador:Usuario conexion:ConexionBD
Flujo de eventos:	<ol style="list-style-type: none"> El administrador escoge la opción de modificación de usuario, El administrador ingresa el nombre y el apellido del usuario a modificar. conexion abre la base de datos y le da el manejador de la base a la clase Usuario, la clase Usuario busca una lista de posibles nombres en la Base, Se presenta en una interfaz una lista de usuarios que coinciden con el nombre y apellido a buscar. El administrador escoge el usuario a anular. Se presenta un mensaje de advertencia antes de anular. la clase usuario anula al usuario correspondiente, la clase Usuario cierra la Base Mostrar un mensaje de anulación exitosa.

Nombre de Caso de uso:	3. Anular Usuario
Nombre de Escenario:	3.2. Anulación no exitosa de usuario debido a errores internos
Instancias de actores participantes:	administrador:Usuario conexion:ConexionBD
Flujo de eventos:	<ol style="list-style-type: none"> El administrador escoge la opción de modificación de usuario, El administrador ingresa el nombre y el apellido del usuario a modificar. conexion abre la base de datos y le da el manejador de la base a la clase Usuario, la clase Usuario busca una lista de posibles nombres en la Base, Se presenta en una interfaz una lista de usuarios que coinciden con el nombre y apellido a buscar.

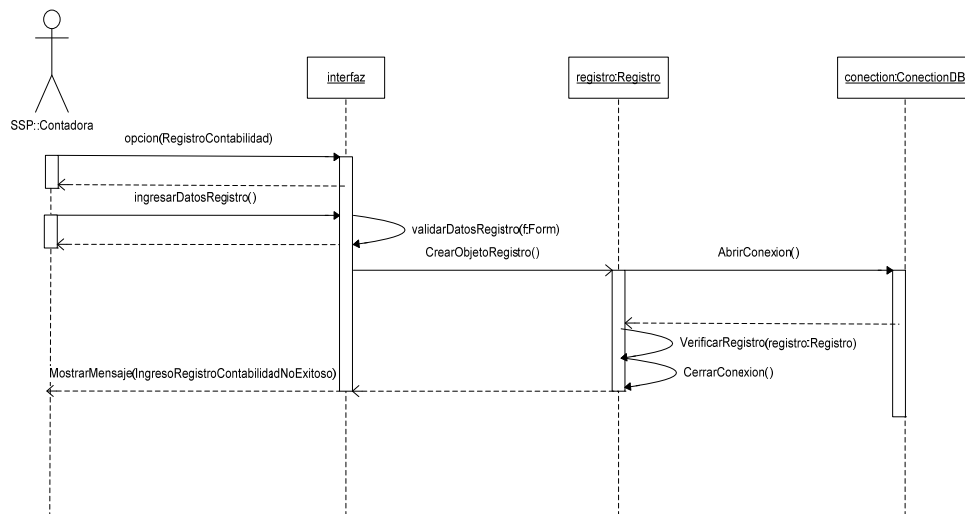
	<ol style="list-style-type: none"> 6. El administrador escoge el usuario a anular. 7. Se presenta un mensaje de advertencia antes de anular. 8. la clase usuario al anular al usuario correspondiente, se presenta un error, 9. la Clase Usuario controla el error, 10. la clase Usuario cierra la Base 11. Mostrar un mensaje de anulación no exitosa debido a errores internos.
--	---

1.4.4 Diagramas de Secuencia

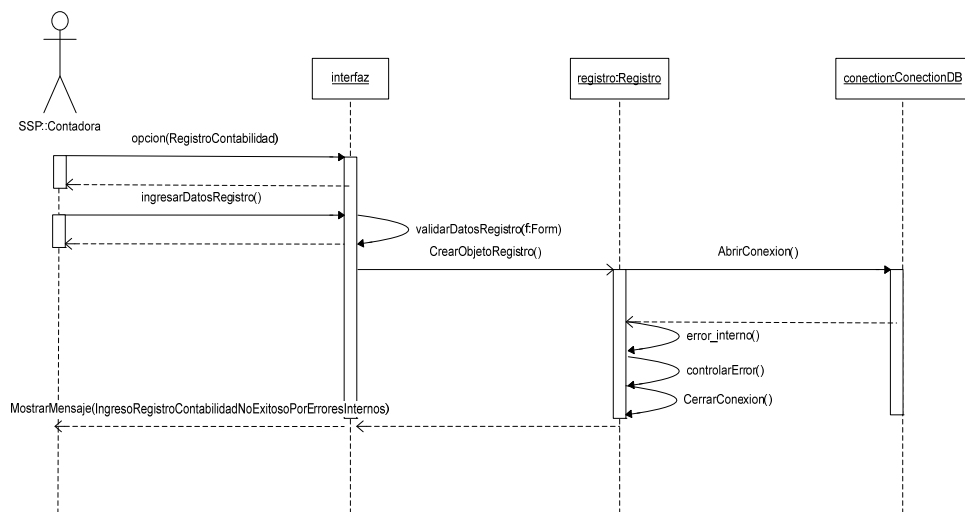
1.1. Ingreso exitoso de registro



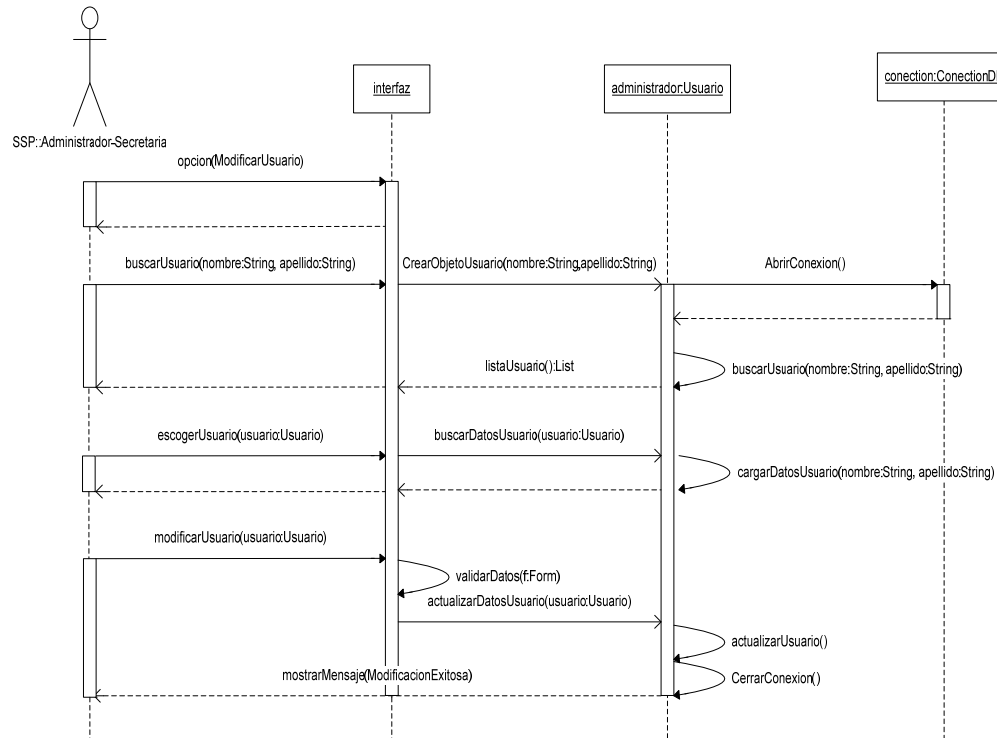
1.2. Ingreso no exitoso de registro debido a que ya existe



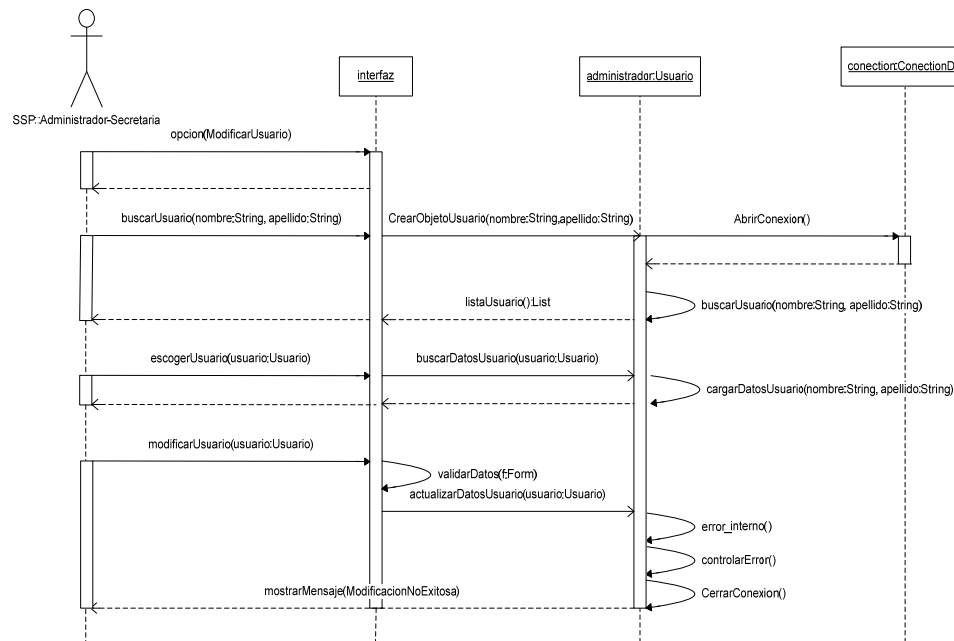
1.3. Ingreso no exitoso de registro debido a errores internos



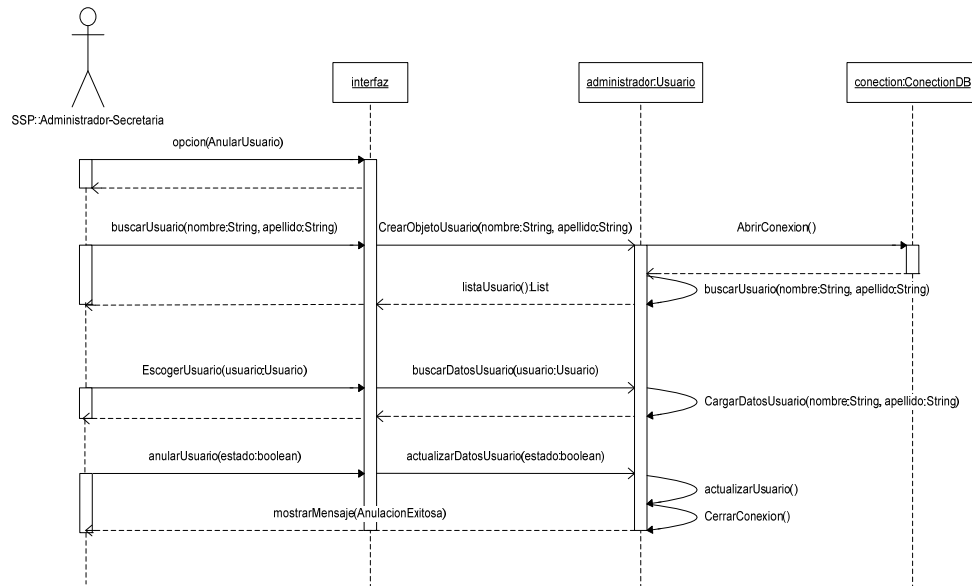
2.1. Modificación exitosa de usuario



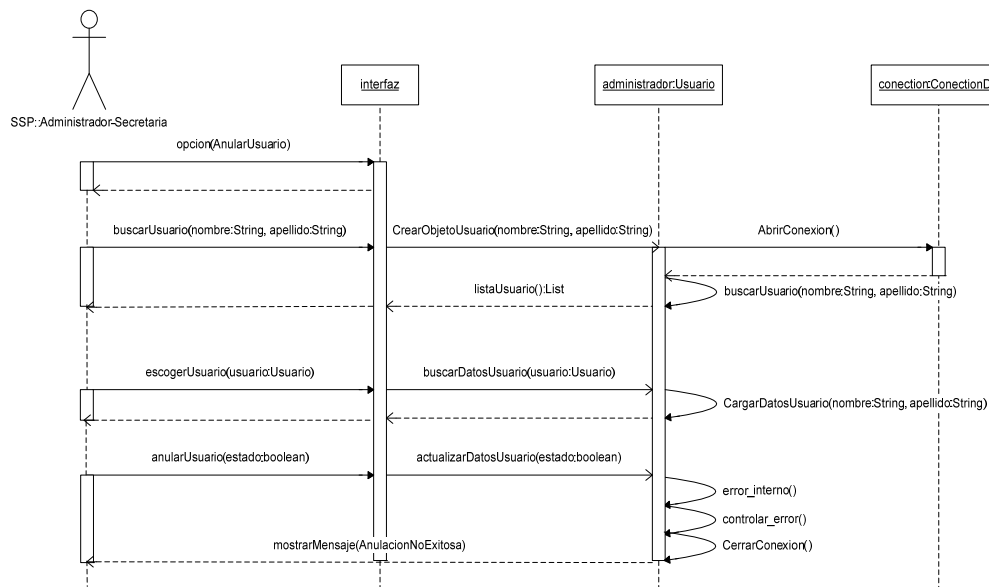
2.2. Modificación no exitosa de Usuario debido a errores internos.



3.1. Anulación exitosa de usuario



3.2. Anulación no exitosa de usuario debido a errores internos



1.5. Invariantes, Precondiciones y Poscondiciones

Este *Sistema de Servicios Parroquiales (SSP)* tendrá las siguientes invariantes de **clases**:

Invariante	Clase	Observación
id_usuario	Usuario.	El id_usuario tiene que ser del tipo administrador o secretaria.
valor	Registro	El valor debe ser mayor a cero para ser ingresado el registro
tipo	Registro	El tipo debe ser ingreso y egresos internos o externos.
anio	RegistroCivil RegistroEclesiástico	El año de ambos registros debe ser menor igual al año actual.

Este *Sistema de Servicios Parroquiales (SSP)* tendrá las siguientes invariantes de **función**:

Invariante	Función	Observación
tipo de la Clase Registro	ingresarRegistro() de Registro	tipo= = ingreso tipo = = egreso interno tipo = = egreso externo
estado de la Clase Usuario	anularUsuario() de Usuario	estado = 0 estado=1

Este *Sistema de Servicios Parroquiales (SSP)* tendrá las siguientes poscondición y precondition de **función**:

PreCondición	Función	PosCondición
estado de la Clase Usuario se encuentre como verdadero.	anularUsuario() de Usuario.	estado de la Clase Usuario sea falso.