

Universidad Autónoma Metropolitana

Unidad Azcapotzalco

Lic. en Ing. en Computación

División de Ciencias Básicas e Ingeniería

Reporte Final

**Sistema de gestión para la recaudación de impuestos
relativos al servicio municipal de agua con base en lo
establecido por Ciudades Digitales.**

Alumnos:

Bergara Chávez Humberto Rodrigo
Carlo Ferrel Giovanni

203305451
203300914

Trimestre 13I
15 Abril de 2013

Asesora de Proyecto Final
M.C. Rafaela Blanca Silva López

Índice de contenido

Tabla de Contenidos

<i>Objetivo General</i>	3
<i>Objetivos Particulares</i>	3
<i>Antecedentes</i>	3
<i>Justificación</i>	4
<i>Desarrollo del Proyecto</i>	6
<i>Caso de uso: Preparar pago de adeudo del contribuyente</i>	8
<i>Caso de uso: Pago de adeudo del contribuyente</i>	9
<i>Caso de uso: Consultar historial de consumos</i>	10
<i>Caso de uso: Consultar contratos</i>	10
<i>Caso de uso: Modificaciones a adeudo</i>	11
<i>Caso de uso: Consultar tarifas</i>	11
<i>Caso de uso: Administrar contratos</i>	11
<i>Caso de uso: Modificar información de contacto</i>	13
<i>Caso de uso: Administrar variables fiscales</i>	13
<i>Caso de uso: Administrar usuarios</i>	13
<i>Caso de uso: Reportes de actividad</i>	14
<i>Diagrama de dominio</i>	15
<i>Diagramas de Navegación</i>	16
<i>Diagramas de clases</i>	19
<i>Estructura de la base de datos</i>	22
<i>Implementación</i>	26
<i>Tecnología utilizada</i>	26
<i>Conclusiones</i>	28
<i>Bibliografía</i>	28
<i>Referencias</i>	29
<i>Anexo 1. Manual del Programador</i>	30
<i>Anexo 2. Guía de Instalación</i>	62
<i>Anexo 3. Manual del Usuario Final</i>	67

Objetivo General

Construir una aplicación web para la gestión de la recaudación de impuestos relativos al servicio de Agua de acuerdo con lo estipulado en el “Código Financiero del Estado de México y Municipios” siguiendo los principios de una ciudad digital.

Objetivos Particulares

- Identificar los problemas de recaudación de impuestos en los municipios del Estado de México que podremos solucionar por medio de la aplicación que desarrollaremos.
- Investigar las características de un sistema para una ciudad digital.
- Dar una propuesta de solución a estos problemas por medio de una aplicación web robusta con características de ciudad digital y que pueda satisfacer las necesidades de recaudación de impuestos.

Antecedentes

En la actualidad, las tecnologías de la información¹ forman parte importante en la vida cotidiana de la sociedad e impactan de forma directa a los sistemas: económico, productivo y social. Esto lleva a una integración cada vez mayor de estas tecnologías en todo tipo de interacciones sociales. En nuestro caso estamos interesados en la interacción entre el gobierno y el ciudadano, esta interacción esta basada en el cumplimiento de derechos y obligaciones por ambas partes de acuerdo a lo estipulado en las leyes.

En particular es de interés para este proyecto aquellos sistemas de información que faciliten al ciudadano el cumplimiento de sus obligaciones y le permitan ejercer el derecho a la información y transparencia por parte del gobierno y por lo tanto mejorar la calidad de vida del ciudadano siguiendo los principios de una ciudad digital².

Algunos casos de éxito de este tipo de sistemas en México los encontramos con municipios como Mérida, San Pedro Garza, Guadalajara entre otros. En todos ellos encontramos modelos a seguir, por el claro avance que han tenido en esta materia, sin embargo lo mas importante a resaltar de estos casos de éxito en interés al proyecto, es la implementación de sistemas robustos, de fácil acceso al ciudadano y al funcionario público y sobre todo sistemas que mantienen su funcionalidad a largo plazo y con

¹ Conjunto de productos derivados de las nuevas herramientas (software y hardware), soportes de la información y canales de comunicación relacionados con el almacenamiento, procesamiento y transmisión digitalizados de la información.

² Un modelo avanzado de comunidad, donde se materializan las nuevas formas de relacionarse con el medio a través de la Sociedad de la Información.

capacidad de crecimiento.

Es claro que este tipo de sistemas son desarrollados en su mayoría como software a la medida³, debido al poco nivel de coordinación que hay entre los gobiernos municipales y estatales y a las características particulares de cada municipio, así como a las administraciones en turno. Por lo que es difícil poner un antecedente directo que se asemeje al proyecto. Más bien nos basaremos en lo establecido en el código financiero del Estado de México y Municipios y siguiendo los lineamientos básicos para una ciudad digital como son: Mejorar la eficiencia de la atención al ciudadano por medio de servicios electrónicos avanzados.

Justificación

Hace alrededor de 15 años ha existido la tendencia por parte de gobiernos locales de “poner la tecnología al servicio de las necesidades cotidianas del ciudadano”, lo cual define el concepto de ciudades digitales. En nuestro país estamos en un momento de cambio para este tipo de servicios, prueba de esto es el apoyo que se ha dado al programa de e-gobierno⁴ durante los últimos años y la cada vez mayor cantidad de municipios que integran tecnologías de información para ofrecer sus servicios.

En la actualidad se hace imprescindible para una delegación o municipio, la utilización de un sistema que facilite el desarrollo de las actividades así como la gestión y seguimiento eficiente de las mismas.

Un sistema informático basado en web, permite a los usuarios involucrados el acceso a la información desde cualquier lugar en el que se encuentre por medio de internet de manera rápida, segura y actualizada. El poseer la información en tiempo y en la forma deseada permite la toma oportuna de decisiones.

El presente proyecto propone facilitar el proceso de recaudación de pagos, relacionados al sistema de agua en la delegación, además de gestionarlos y dar seguimiento a los contribuyentes. El usuario del sistema podrá consultar con algunos pocos datos el estado actual de un contribuyente, historial, realizar pagos, calcular recargos, etc.

El proyecto se propone de forma general, pensando en que podría adaptarse de acuerdo a las necesidades particulares de el cliente final, pero siempre basándose en lo estipulado en la ley. La cual en el caso del Estado de México es la misma para todos los municipios. Por lo que se considera que la adaptación del sistema a un municipio cualquiera será más que nada en cuanto a la parte de la vista, y la administración. Pues en cuanto a lo que se refiere a la lógica de negocio ésta, al estar basada en la ley, no debe tener cambios entre municipios y explica claramente como debe de ser aplicado el impuesto en cada municipio.

³ También llamado en ocasiones software privado, más bien se refiere a software que satisface necesidades particulares de un cliente.

⁴ También llamado gobierno electrónico. Uso de las tecnologías de la información y el conocimiento en los procesos internos de gobierno y en la entrega de los productos y servicios del Estado tanto a los ciudadanos como a la industria.

Por ejemplo:

Artículo 95.- Por los servicios prestados por el Organismo Público Descentralizado, denominado, Comisión del Agua del Estado de México se pagarán los derechos sin exención alguna conforme a lo siguiente:

- I. Por la conexión de la toma para el suministro de agua en bloque proporcionada por la Comisión del Agua del Estado de México, se pagará de acuerdo a la siguiente:

TARIFA				
GRUPO DE MUNICIPIOS				
M3/día	1	2	3	4
Uso doméstico	\$2,559	\$2,405	\$2,252	\$2,100
Uso no doméstico	\$3,199	\$2,892	\$2,686	\$2,531

Para su aplicación se atenderá a la clasificación de municipios contenida en el artículo 140 de este Código.

Extracto del Código Financiero del Estado de México y Municipios.

Los conocimientos necesarios para la elaboración de este proyecto son: programación orientada a objetos, bases de datos, metodología para el análisis y diseño de sistemas de información, bases de datos, ingeniería de software. Los conocimientos requeridos corresponden a la formación de una Licenciatura en Ingeniería en Computación impartida en la UAM. La magnitud del proyecto, también genera la necesidad de un equipo para el desarrollo de esta propuesta, concretamente de un equipo formado por dos alumnos.

Desarrollo del Proyecto

El desarrollo del proyecto se realizó por etapas. En una etapa inicial nos dedicamos a hacer el desarrollo del back-end, es decir los catálogos, administradores y acciones y a la par el desarrollo a grandes rasgos de la vista o front-end con ExtJS. Debido a nuestra falta de experiencia con la integración de esta tecnología, gran parte del tiempo de desarrollo fue dedicado a conocer e integrar ExtJS con la configuración que ya conocíamos del lado del back-end.

Cuando logramos comprender mejor la integración de las tecnologías ocupadas, el desarrollo fue más rápido.

Podríamos decir que empezamos usando una metodología basada en modelado RUP y acabamos aplicando una de desarrollo extremo o similar.

Casos de Uso

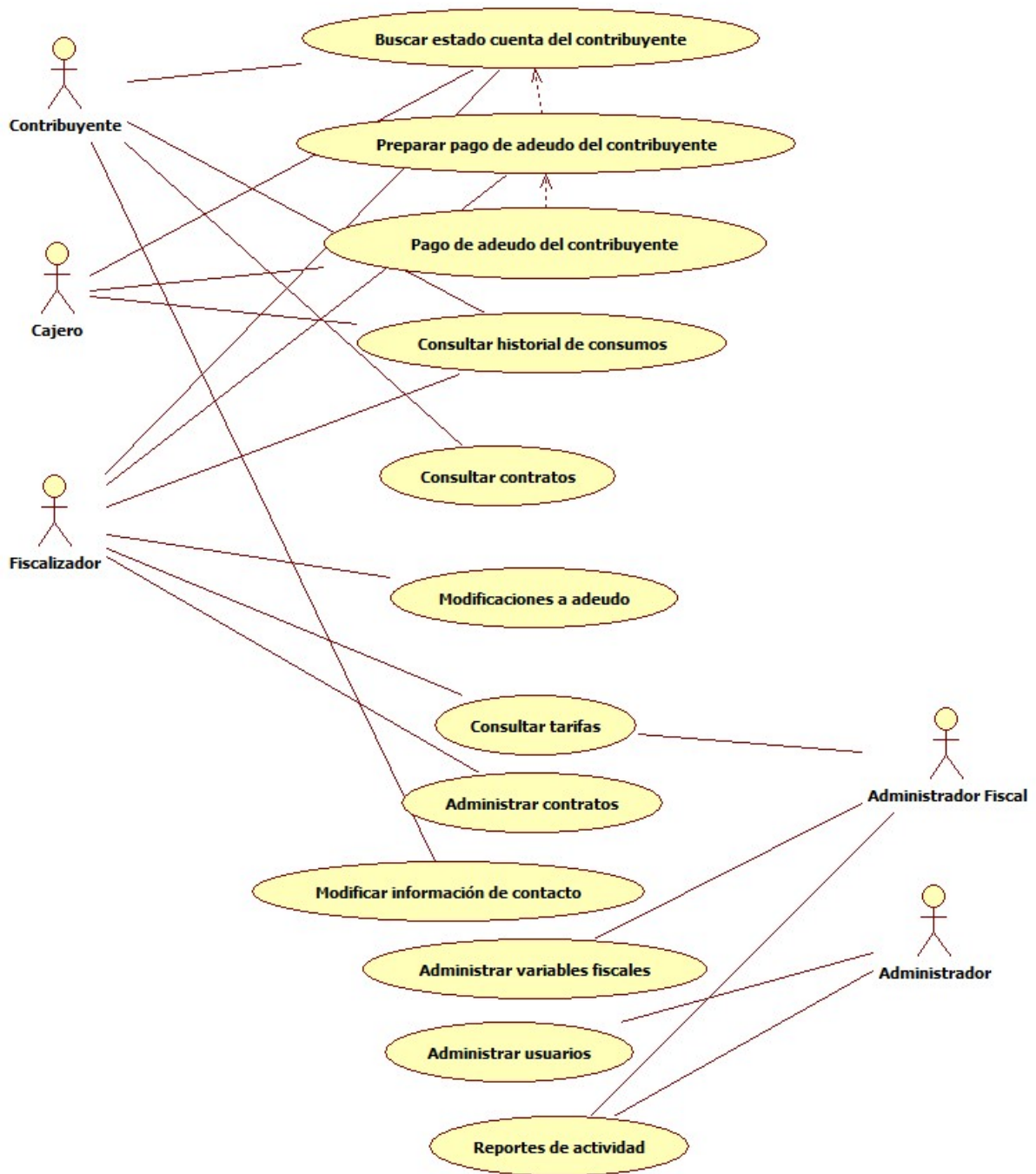


Figura 1: Diagrama de casos de uso. Todos los usuarios pasan por el caso de uso de Autenticación (el cual se omite en el diagrama) para poder hacer uso del sistema.

Caso de uso: Buscar estado cuenta del contribuyente

Roles: Contribuyente, fiscalizador.

Precondiciones: Usuario autenticado que esta buscando la información del estado de cuenta.

Postcondiciones: Se muestra la información relacionada con la cuenta del *contribuyente*.

Flujo básico:

1. El usuario da click en la opción del menú.
2. El sistema muestra la forma para ingresar por alguno de los metodos de busqueda disponibles.
3. El usuario ingresa la información en la forma y da click en el botón de Enviar.
4. El sistema muestra la información general de la cuenta y el listado de adeudos pendientes del *contribuyente* (en caso de haberlos).

Flujos alternos:

1. El *contribuyente* solicita una impresión.
 - a. El *fiscalizador* puede dar click en el icono “imprimir adeudos” para imprimir el resultado de la busqueda de estado de cuenta del contribuyente
2. El estado de cuenta es del *contribuyente* equivocado (error de captura de parametros de busqueda).
 - a. Se repite la operación desde el paso 1 del flujo básico.
3. Los datos ingresados en la búsqueda no regresaron ningun resultado.
 - a. El sistema informa al usuario que la información ingresada no existe en el sistema y que la verifique, dejandolo en la misma pantalla de forma de búsqueda.

Observaciones:

En algun punto el mismo contribuyente podria imprimir el resultado de la busqueda, esto dependiendo de los requisitos de la dependencia que ocupe del sistema.

Caso de uso: Preparar pago de adeudo del contribuyente

Casos de uso relacionados: Buscar estado de cuenta del contribuyente

Roles: Fiscalizador

Precondiciones: Usuario autenticado y dentro del flujo básico del caso de uso Buscar estado de cuenta del *contribuyente*

Postcondiciones: El contribuyente obtiene el formato de pago para el o los adeudos que elige pagar en una sola exhibición.

Flujo básico:

1. El *contribuyente* una vez que vio el listado de adeudos puede informar al *fiscalizador* que va a pagar un determinado número de adeudos.
2. El *fiscalizador* elige cual es el adeudo mas reciente de lo elegidos para pagar y da click en el icono de “pagar” a la derecha de la fila del adeudo elegido.

3. El sistema muestra cual es el monto total de adeudos que debe pagar el contribuyente desde el más antiguo hasta el adeudo elegido.
4. El *contribuyente* confirma que desea pagar los adeudos por la cantidad motrada.
5. El *fiscalizador* da click en la opción “Aceptar”
6. El sistema muestra el formato de pago para impresión

Flujos alternos:

1. El *contribuyente* elige no pagar los adeudos por esa cantidad si no por otra.
 - a. El fiscalizador puede cancelar y realizar los mismos pasos desde el punto 2.

Observaciones:

En algun punto el mismo contribuyente podria imprimir su formato de pago, esto dependiendo de los requisitos de la dependencia que ocupe del sistema.

Caso de uso: Pago de adeudo del contribuyente

Casos de uso relacionados: Preparar pago de adeudo del contribuyente

Roles: Cajero

Precondiciones: El *cajero* esta autenticado en el sistema y el *contribuyente* tiene el formato de pago generado en el caso de uso Preparar pago de adeudo del contribuyente que quiere pagar.

Postcondiciones: El *cajero* recibe el pago correpondiente al formato de pago y el pago queda realizado y registrado en el sistema.

Flujo básico:

1. El *contribuyente* acude con el *cajero* para hacer el pago por la cantidad que se indica en el formato de pago.
2. El *cajero* ingresa los datos de la cuenta del contribuyente para identificar el formato de pago dentro del sistema.
3. El *cajero* cobra al cliente la cantidad indicada en el recibo y da click en el botón de “Pagar”
4. El sistema registra el pago y muestra al *cajero* la confirmación del pago.

Flujos alternos:

1. El *contribuyente* decide no hacer el pago.
 - a. El cajero solamente debe de salir del listado de pagos pendientes del *contribuyente* ingresado y continuar su labor normalmente.
 - b. El sistema hace un corte al final del día de los formatos de pago no realizados para restablecer los adeudos que estaban en proceso de ser pagados.

Observaciones:

Caso de uso: Consultar historial de consumos

Roles: Contribuyente, cajero, fiscalizador

Precondiciones: Usuario autenticado que quiere consultar el historial de consumos del contribuyente.

Postcondiciones: Listado con información del historial del contribuyente. Se puede revisar que consumos se han pagado y cuando.

Flujo básico:

1. El usuario da click en la opción del menú para buscar el historial de consumos.
2. El sistema muestra el listado de consumos en orden de antigüedad

Flujos alternos:

1. El *contribuyente* solicita una impresión.
 - a. El *usuario* puede dar click en el icono “imprimir adeudos” para imprimir el resultado de la búsqueda de estado de cuenta del contribuyente
2. El estado de cuenta es del *contribuyente* equivocado (error de captura de parametros de búsqueda).
 - a. Se repite la operación desde el paso 1 del flujo básico.
3. Los datos ingresados en la búsqueda no regresaron ningun resultado.
 - a. El sistema informa al usuario que la información ingresada no existe en el sistema y que la verifique, dejándolo en la misma pantalla de forma de búsqueda.

Observaciones:

En algun punto el mismo contribuyente podria imprimir el resultado de la búsqueda, esto dependiendo de los requisitos de la dependencia que ocupe del sistema.

Caso de uso: Consultar contratos

Roles: Contribuyente

Precondiciones: Usuario autenticado y que quiere realizar consulta de los contratos donde es el titular de la cuenta.

Postcondiciones: El usuario puede ver la información de los contratos que estan a su nombre.

Flujo básico:

1. El usuario da click en la opción del menú para consultar sus contratos.
2. El sistema muestra un listado de contratos a nombre del usuario que esta en sesión.
3. El usuario puede ver el detalle de los contratos a su nombre al dar click en el icono de información de cada contrato dentro del listado.

Flujos alternos:

Observaciones:

El sistema puede incluir mas funcionalidades para el contribuyente para administrar sus contratos dependiendo de los requisitos finales de la dependencia que ocupe el sistema.

Caso de uso: Modificaciones a adeudo

Roles: Fiscalizador

Precondiciones: Usuario autenticado y con autorización para hacer modificaciones a un adeudo dentro de los consumos de un contribuyente.

Postcondiciones: El usuario realiza la modificación al consumo para ajustar su calculo de acuerdo a lo autorizado.

Flujo básico:

1. El *fiscalizador* ingresa a la opción de menú para modificar consumo.
2. El sistema muestra los consumos pendientes de pagar.
3. El *fiscalizador* escoge el consumo a modificar y edita el campo adecuado y da click en el botón de guardar.
4. El sistema realiza el ajuste y registra el cambio.

Flujos alternos:

1. El usuario comete un error en la modificación del consumo.
 - a. El usuario puede repetir el proceso desde el punto 3 del flujo básico.

Observaciones:

Caso de uso: Consultar tarifas

Roles: Administrador fiscal, fiscalizador

Precondiciones: Usuario autenticado y que necesita revisar las tarifas vigentes o de años anteriores.

Postcondiciones: El usuario puede ver las tarifas vigentes o de años anteriores.

Flujo básico:

1. El usuario da click en la opción del menú consultar tarifas
2. El sistema muestra las tarifas vigentes.

Flujos alternos:

1. El usuario quiere consultar las tarifas de años anteriores.
 - a. El usuario puede acceder a la opción del menú para consultar tarifas por año.
 - b. El sistema muestra una forma con un combobox para elegir el año.
 - c. El usuario escoge el año y da click en "Aceptar"
 - d. El sistema muestra las tarifas correspondientes al año seleccionando.
2. El usuario quiere imprimir el listado de tarifas.
 - a. El usuario puede dar click en el botón imprimir tarifas de la tarifas listadas.

Observaciones:

Caso de uso: Administrar contratos

Roles: Fiscalizador

Precondiciones: Usuario autenticado y con la necesidad de Agregar, modificar, suspender o eliminar contratos.

Postcondiciones: El usuario agrego, modifiko, suspendio o elimino un contrato del sistema.

Flujo básico:

1. El usuario da click en la opción de menú para Agregar, modificar, suspender o eliminar un contrato.
 - a. Agregar
 - i. El sistema muestra una forma con los campos necesarios a llenar para crear el nuevo contrato.
 - ii. El usuario llena la forma y da click en “Enviar”
 - b. Modificar
 - i. El sistema muestra una forma de búsqueda para ingresar los datos del contrato que se quiere modificar.
 - ii. El usuario ingresa los datos del contrato que quiere editar y da click en el botón “Buscar”.
 - iii. El sistema muestra los datos del contrato en una forma con campos editables.
 - iv. El usuario hace los cambios al contrato y da click en la opción de “Guardar”
 - c. Suspender
 - i. El sistema muestra una forma de búsqueda para ingresar los datos del contrato que se quiere suspender.
 - ii. El usuario ingresa los datos del contrato que se quiere suspender y da click en el botón buscar.
 - iii. El sistema muestra la información del contrato.
 - iv. El usuario da click en el botón “Suspender”.
 - v. El sistema muestra una forma para ingresar la razón de la suspensión.
 - vi. El usuario captura la razón y da click en “Continuar”
 - vii. El sistema cambia el estado del contrato a suspendido.
 - d. Eliminar
 - i. El sistema muestra una forma de búsqueda para ingresar los datos del contrato que se quiere eliminar.
 - ii. El usuario ingresa los datos del contrato que se quiere eliminar y da click en el botón buscar.
 - iii. El sistema muestra la información del contrato.
 - iv. El usuario da click en el botón “Eliminar”.
 - v. El sistema muestra una forma para ingresar la razón de eliminación.
 - vi. El usuario captura la razón y da click en “Continuar”
 - vii. El sistema cambia el estado del contrato a eliminado.

Flujos alternos:

1. El usuario ya no quiere crear un nuevo contrato.
 - a. El usuario puede dar click en la opción cancelar o simplemente cambiar a otra opción del menú.
2. El usuario se equivoca en la actualización del contrato.
 - a. El usuario puede continuar con la modificación de contrato y continuar

haciendo cambios hasta que de click en otra opción del menú.

Observaciones:

Otras opciones a considerar en funcionamiento con la suspensión del contrato son la activación del contrato y bajo que criterios se suspende/activa un contrato.

Caso de uso: Modificar información de contacto

Roles: Contribuyente

Precondiciones: Usuario autenticado y que quiere modificar sus datos de contacto.

Postcondiciones: Usuario pudo modificar sus datos de contacto.

Flujo básico:

1. El usuario da click en la opción información de contacto del menú.
2. El sistema muestra en pantalla la información actual que tiene el usuario.
3. El usuario da click en actualizar información.
4. El sistema muestra la forma editable para modificar los campos disponibles.
5. El usuario modifica los campos necesarios y da click en “Guardar”
6. El sistema actualiza la información de la forma y muestra la pantalla del paso 2.

Flujos alternos:

1. El usuario ya no quiere modificar su información de contacto.
 - a. El usuario puede cancelar la operación dando click en el botón de “Cancelar”

Observaciones:

Caso de uso: Administrar variables fiscales

Roles: Administrador Fiscal

Precondiciones: Usuario autenticado y con la necesidad de actualizar la variables fiscales por algun ajuste a la ley.

Postcondiciones: El usuario logra actualizar las variable fiscales.

Flujo básico:

1. El usuario da click en la variables fiscal que desee administrar del menú.
2. El sistema muestra un listado de los valores actuales de la variable fiscal seleccionada.
3. El usuario puede modificar las variables fiscales como un catálogo con operaciones de agregar, modificar, eliminar.
4. El sistema guarda la información modificada de las variables fiscales en el sistema.

Flujos alternos:

Observaciones:

Caso de uso: Administrar usuarios

Roles: Adminisitrador del sistema

Precondiciones: Usuario autenticado que quiere administrar a los usuarios del sistema.

Postcondiciones: Se han agregado, actualizado o eliminado usuarios del sistema.

Flujo básico:

1. El *usuario* escoge de las opciones del menú Administración la opción usuarios.

2. El sistema muestra el listado de usuarios actualmente activos en el sistema.
3. El usuario puede agregar, editar, eliminar.
 - a. Agregar
 - i. Al dar click en el botón agregar una nueva fila de usuario aparecera para poder ser editada en el listado de usuarios del sistema.
 - ii. El usuario llena los campos necesarios del nuevo usuario del sistema y da click en “Guardar”.
 - iii. El sistema registra al nuevo usuario.
 - b. Editar
 - i. El usuario puede dar doble click sobre un usuario en el listado de usuarios para editar la información del usuario y posteriormente dar click en “Guardar” para guardar los cambios.
 - ii. El sistema registra los cambios al usuario.
 - c. Eliminar
 - i. El usuario puede elegir un usuario de la lista dando click sobre el y posteriormente click en el botón de “Eliminar”.
 - ii. El sistema elimina al usuario del sistema.

Flujos alternos:

1. El usuario decide dejar de administrar los cambios en el listado de usuarios.
2. El usuario puede dar cancelar o cambiar a otra opción del menú para dejar la sección de administrar usuarios del sistema.

Observaciones:

Caso de uso: Reportes de actividad

Roles: Administrador, AdministradorFiscal

Precondiciones: Usuario autenticado y que necesita conocer reportes del sistema.

Postcondiciones: El usuario puede ver reportes del sistema.

Flujo básico:

1. El usuario selecciona el tipo de reporte del sistema del menú.
2. El sistema muestra el reporte en pantalla.

Flujos alternos:

1. El usuario quiere imprimir el reporte.
 - a. El usuario puede dar click en la opción imprimir reporte una vez que el reporte seleccionado aparezca en pantalla.

Observaciones:

Los reportes que se muestran para administrador y administrador fiscal son distintos debido a los permisos pero su funcionamiento es el mismo.

Diagrama de dominio

Este diagrama describe los principales objetos del negocio participantes en el sistema.

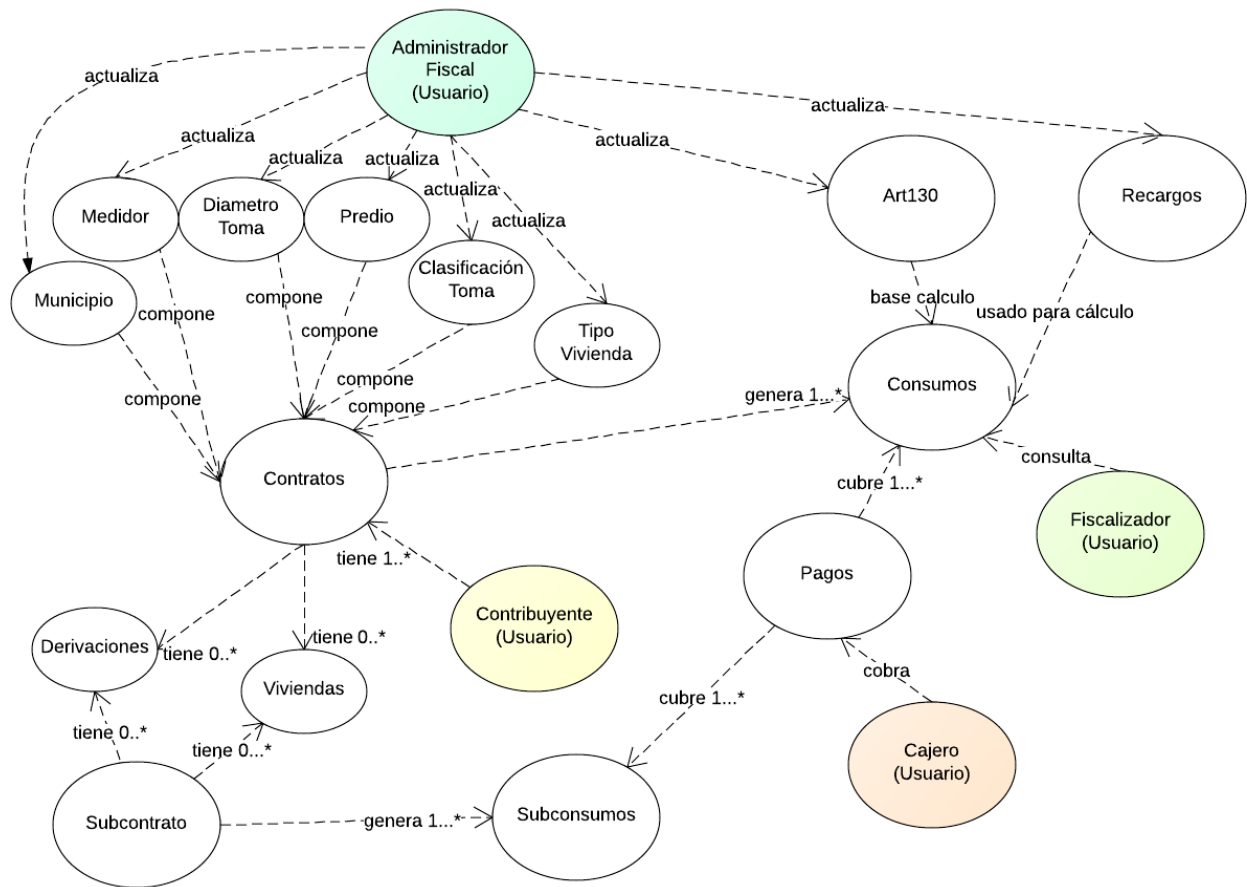


Figura 2: Diagrama de dominio

Diagramas de Navegación

Los diagramas de navegación están separados por Perfil, salvo el de Login que es común a todos.

Diagrama de navegación de sección **Principal**

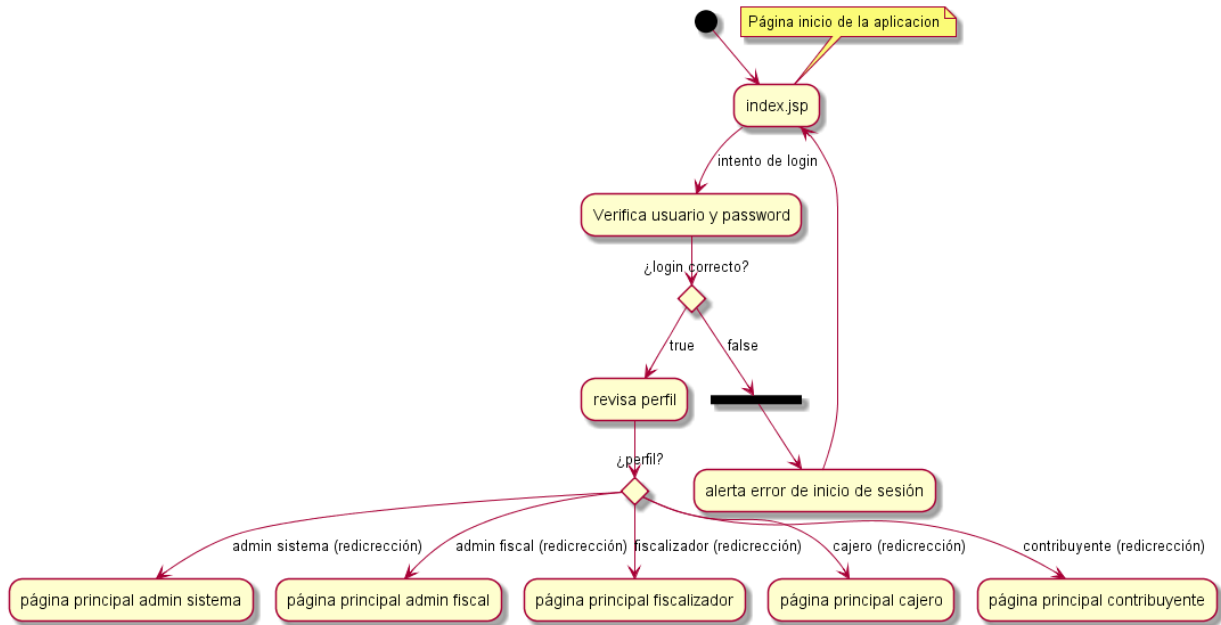


Figura 3: Diagrama de navegación Principal

Diagrama de navegación de sección **Administrador Fiscal**

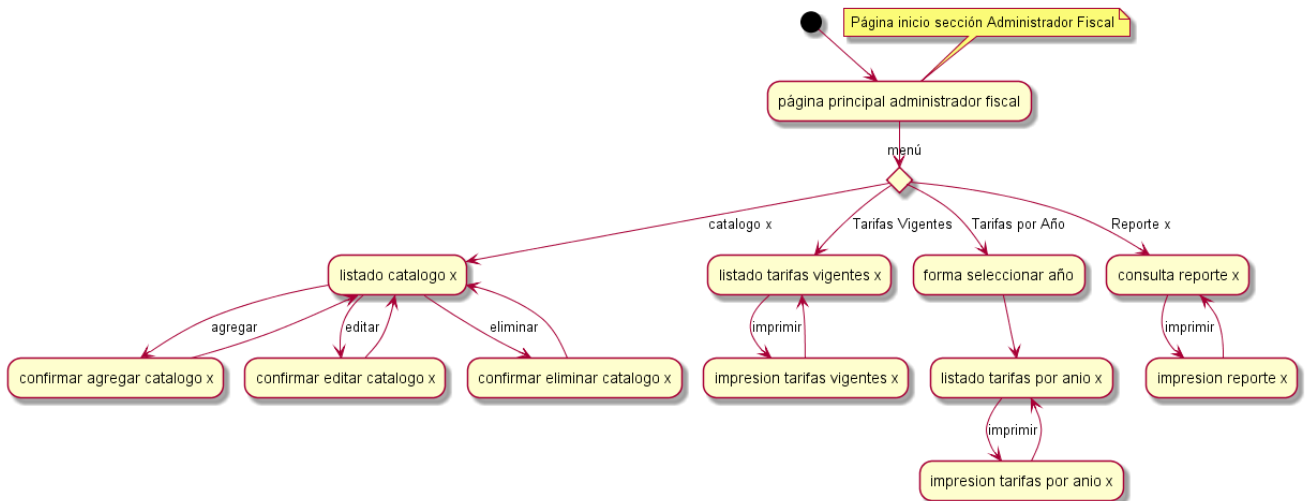


Figura 4: Diagrama de navegación Admnistrador Fiscal

Diagrama de navegación de sección **Fiscalizador**

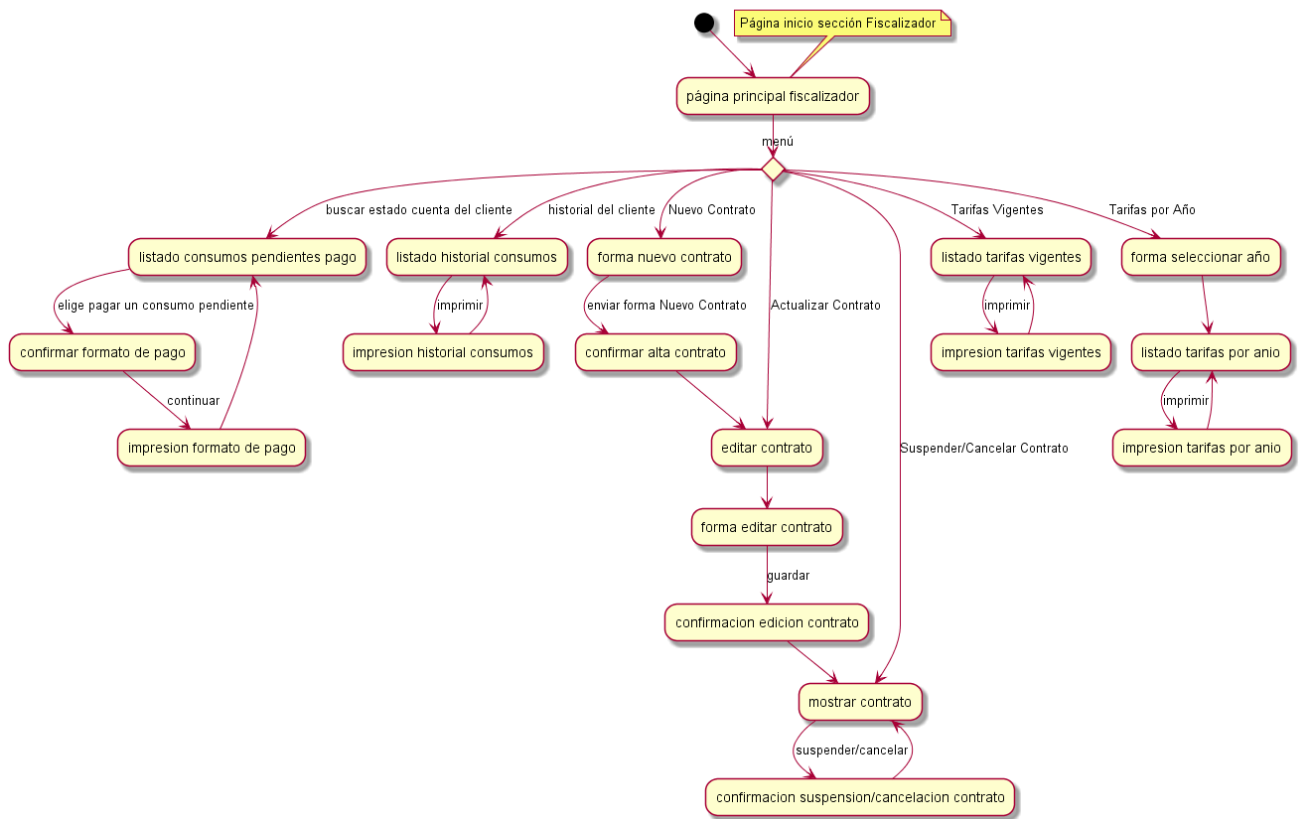


Figura 5: Diagrama de navegación Fiscalizador
Diagrama de navegación de sección **Cajero**

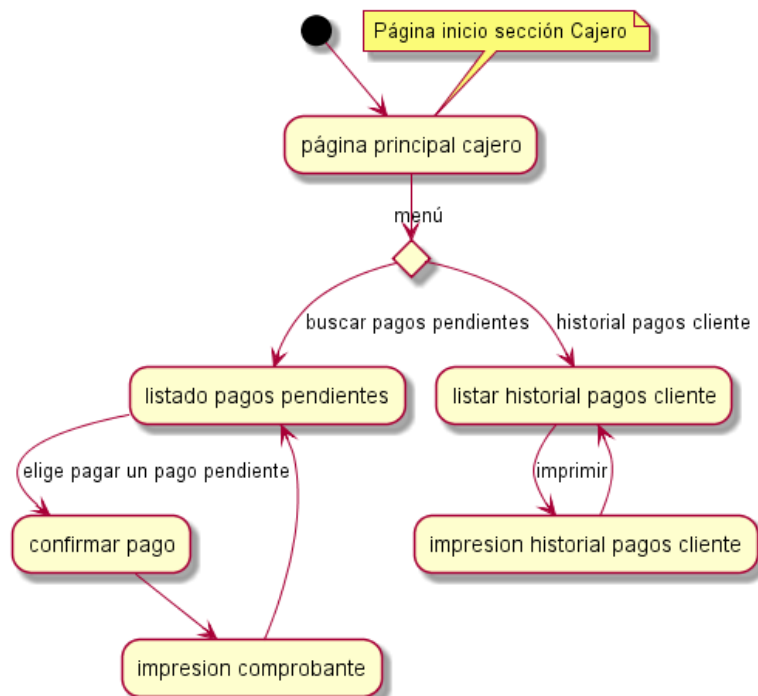


Figura 6: Diagrama de navegación Cajero

Diagrama de navegación de sección **Contribuyente**

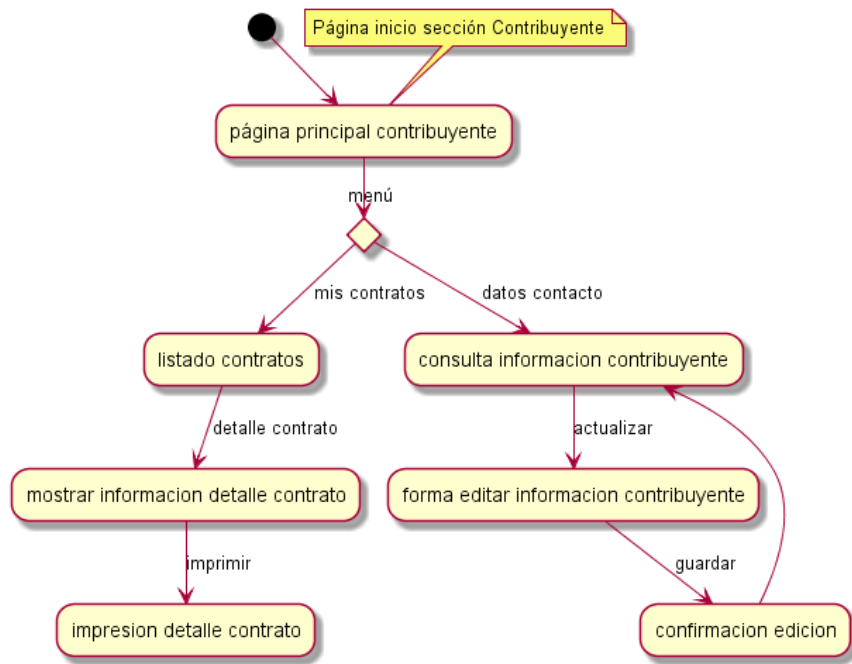


Figura 7: Diagrama de navegación Contribuyente

Diagramas de clases

Las clases en sus respectivos paquetes son las siguientes:

- **mx.uam.azc.sgrima**

- InicializadorCatalogos

- **mx.uam.azc.sgrima.actions**

(Los actions son parte del modelo MVC de struts y se ocupan de fungir como controladores)

- AdminFiscalAction
- AdminFiscalArt130Action
- AdminFiscalCatalogosAction
- CajeroAction
- ContribAction
- FiscalAction
- LoginAction
- PrincipalAction
- SysAdminAction
- SysadminPerfilesAction
- SysadminUsuariosAction

- **mx.uam.azc.sgrima.busines**

(Son las interfaces para los administradores que contralan el acceso al modelo del sistema)

- *AdministradorActualizaciones*
- *AdministradorArt130*
- *AdministradorArt130DomConMedidor*
- *AdministradorArt130DomSinMedidor*
- *AdministradorArt130NoDomConMedidor*
- *AdministradorArt130NoDomSinMedidor*
- *AdministradorCalculosConsumoDCM*
- *AdministradorCfnSgrima*
- *AdministradorClasificacionToma*
- *AdministradorConsumo*
- *AdministradorConsumoPago*
- *AdministradorContratoDetalle*
- *AdministradorContratoMaster*
- *AdministradorDiametroToma*
- *AdministradorGrupoMunicipio*
- *AdministradorLogDefault*
- *AdministradorMedidor*
- *AdministradorMunicipio*
- *AdministradorPago*
- *AdministradorPagoSubconsumo*
- *AdministradorPerfil*
- *AdministradorPredio*
- *AdministradorRecargos*
- *AdministradorSalariosMinimos*
- *AdministradorSubconsumo*
- *AdministradorSubconsumoPago*
- *AdministradorSubcontrato*
- *AdministradorSubcontratoDetalle*
- *AdministradorTipoVivienda*
- *AdministradorUsuario*
- *AdministradorUsuarioExtended*
- *CalculosConsumoBO*
- *ConsumosBO*
- *PagosBO*

- **mx.uam.azc.sgrima.business.local**

(Implementación de las interfaces del modelo)

- AdministradorActualizacionesImpl
- AdministradorArt130DomConMedidorImpl
- AdministradorArt130DomSinMedidorImpl
- AdministradorArt130Impl
- AdministradorArt130NoDomConMedidorImpl
- AdministradorArt130NoDomSinMedidorImpl
- AdministradorCalculosConsumoDCMImpl
- AdministradorCfnSgrimaImpl
- AdministradorClasificacionTomaImpl
- AdministradorConsumoImpl
- AdministradorConsumoPagImpl
- AdministradorContratoDetalleImpl
- AdministradorContratoMasterImpl
- AdministradorDiametroTomaImpl
- AdministradorGrupoMunicipioImpl
- AdministradorLogDefaultImpl
- AdministradorMedidorImpl
- AdministradorMunicipioImpl
- AdministradorPagoImpl
- AdministradorPagoSubconsumoImpl

- AdministradorPerfilImpl
- AdministradorPredioImpl
- AdministradorRecargosImpl
- AdministradorSalariosMinimosImpl
- AdministradorSubconsumoImpl
- AdministradorSubconsumoPagoImpl
- AdministradorSubcontratoDetalleImpl
- AdministradorSubcontratoImpl
- AdministradorTipoViviendaImpl
- AdministradorUsuarioExtendedImpl
- AdministradorUsuariosImpl
- CalculosConsumoBOImpl
- ConsumosBOImpl
- PagosBOImpl

- **mx.uam.azc.sgrima.data**

(Clases que modelan la Base de datos y otros objetos necesarios en el sistema)

- ActualizacionDTO
- Art130DomConMedidorDTO
- Art130DomSinMedidorDTO
- Art130NoDomConMedidorDTO
- Art130NoDomSinMedidorDTO
- CalculosConsumoDTO
- CalculosSubconsumoDTO
- CfnSgrimaDTO
- ClasificacionTomaDTO
- ConsumoDTO
- ConsumoPagoDTO
- ConsumoPorPagarDTO
- ContratoDetalleDTO
- ContratoMasterDTO
- DiametroTomaDTO
- GrupoMunicipioDTO
- LogDefaultDTO
- MedidorDTO
- MunicipioDTO
- PagoDTO
- PagoSubconsumoDTO
- PerfilDTO
- PredioDTO
- RecargoDTO
- SalarioMinimoDTO
- SubconsumoDTO
- SubconsumoPagoDTO
- SubconsumoPorPagarDTO
- SubcontratoDetalleDTO
- SubcontratoDTO
- TipoViviendaDTO
- UsuarioDTO
- UsuarioExtendedDTO

- **mx.uam.azc.sgrima.interceptors**

(Estas clases sirven para validar la sesión de cada tipo de usuario en cada petición)

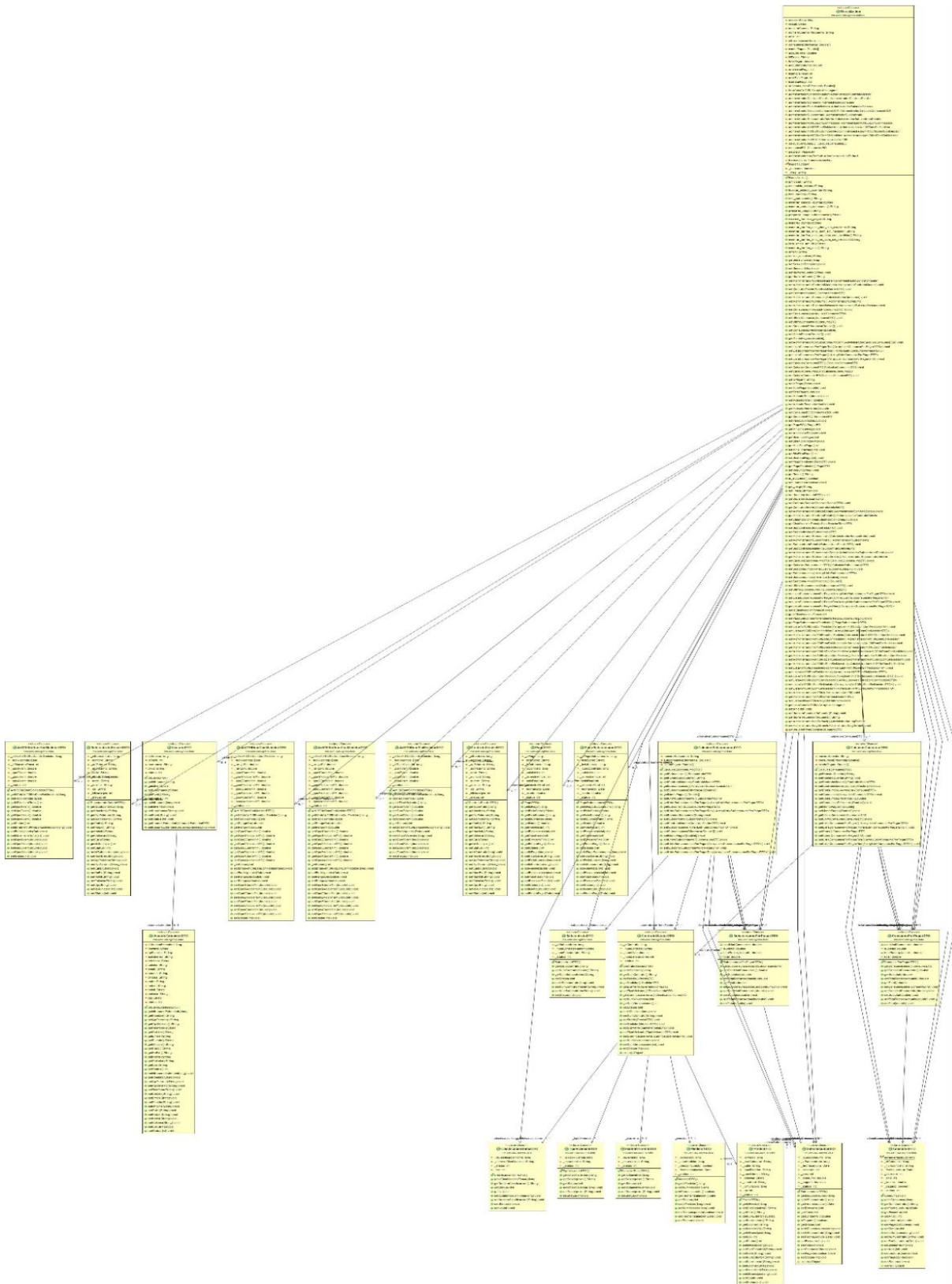
- IntecerptorAutenticacionAdminFiscal
- InterceptorAuntenticacionAdmin
- InterceptorAutenticacionCajero
- InterceptorAutenticacionContribuyente
- InterceptorAutenticacionFiscal

- **mx.uam.azc.sgrima.util**

(Clases auxiliares)

- *ConstantesPDF*
- *ConstantesSistema*
- GenerarPDF
- ToStringDTOs
- Transaction
- TransactionsMap

Debido al gran tamaño del diagrama de clases, solo se muestra una parte de este, el resto del diagrama se encuentra en el documento "Manual del Programador" que se adjunta con este reporte.



Estructura de la base de datos

A continuación, se describen cada una de las tablas de la base de datos de Sgrima.

Nombre de la tabla	Descripción
art_130_dom_con_medidor	Define número de salarios mínimos que se deben cobrar basándose en el rango de m3 consumidos y en el grupo de municipio al que pertenece(el municipio) para las tomas domésticas con medidor.
art_130_dom_sin_medidor	Define el número de salarios mínimos que se cobran basándose en el grupo de municipio al que pertenece(el municipio) para tomas domesticas sin medidor.
art_130_no_dom_con_medidor	Define número de salarios mínimos que se deben cobrar basándose en el rango de m3 consumidos y en el grupo de municipio al que pertenece(el municipio) para las tomas no domésticas con medidor.
art_130_no_dom_sin_medidor	Define el número de salarios mínimos que se cobran basándose en el grupo de municipio al que pertenece(el municipio) para tomas no domésticas sin medidor.
cfn_sgrima	Información de la configuración actual del sistema.
clasificaciones_toma	Indica el tipo de contrato de la vivienda y por lo tanto cómo será tratado el contrato y la forma de hacer los cálculos en el sistema.
consumos	Lleva registro de la fecha en que se toma el consumo de una vivienda o toma, para las que tienen medidor. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado
consumos_pagos	Lleva registro de la relacion de consumos y a que pago corresponden. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado. La fecha hace referencia a la ultima transaccion
contrato_detalle	Guarda informacion de facturacion para viviendas o derivaciones que están relacionadas con un contrato_master.
contrato_master	Guarda información detallada del contrato para realizar cálculos.
diametros_toma	Los diferentes tipos de toma(diferenciados por el tamaño

	de la toma) manejados por el sistema.
grupos_municipios	Detalla la información de grupos de municipios.
log_default	Almacena la actividad de los usuarios a lo largo de la aplicación.
medidores	Detalla la información relacionada con un medidor y su estado.
municipios	Detalle de la información de cada municipio.
pagos	Información sobre pagos_subconsumos realizados de un subcontrato. Un pago cubre por lo menos 1 consumo. Status: 1 = 2 = pendiente 3 = pagado
pagos_subconsumos	Lleva registro de la relación de pagos_subconsumos y a que pago corresponden. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado. La fecha hace referencia a la última transacción
perfiles	Los diferentes perfiles de usuario que están involucrados en el sistema.
predios	Información del predio, el predio está asociado a con un contrato_master.
recargos	indica la tasa a cobrar por recargos cuando se genera un atraso.
salarios_minimos	indica el valor del del salario mínimo para la zona.
subconsumos	Llevar control de los consumos cuando se hace división de pagos como en el caso de Mixto con Medidor. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado
subconsumos_pagos	Lleva registro de la relación de subconsumos y a que pago corresponden. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado. La fecha hace referencia a la ultima transaccion
subcontrato_detalle	Información complementaria del subcontrato para realizar cálculos.

subcontratos	Guarda información detallada del subcontrato para realizar cálculos.
tipos_vivienda	Indica el tipo de vivienda el cual se usa para el cálculo del consumo.
usuarios	Información de usuarios del sistema.
usuarios_extended	Información complementaria para los usuarios del sistema.

En la figura 14, se muestra la estructura completa de la base de datos. Por cuestiones de espacio la imagen fue escalada al espacio de la presente página. La estructura y la imagen se adjuntan como parte del entregable.

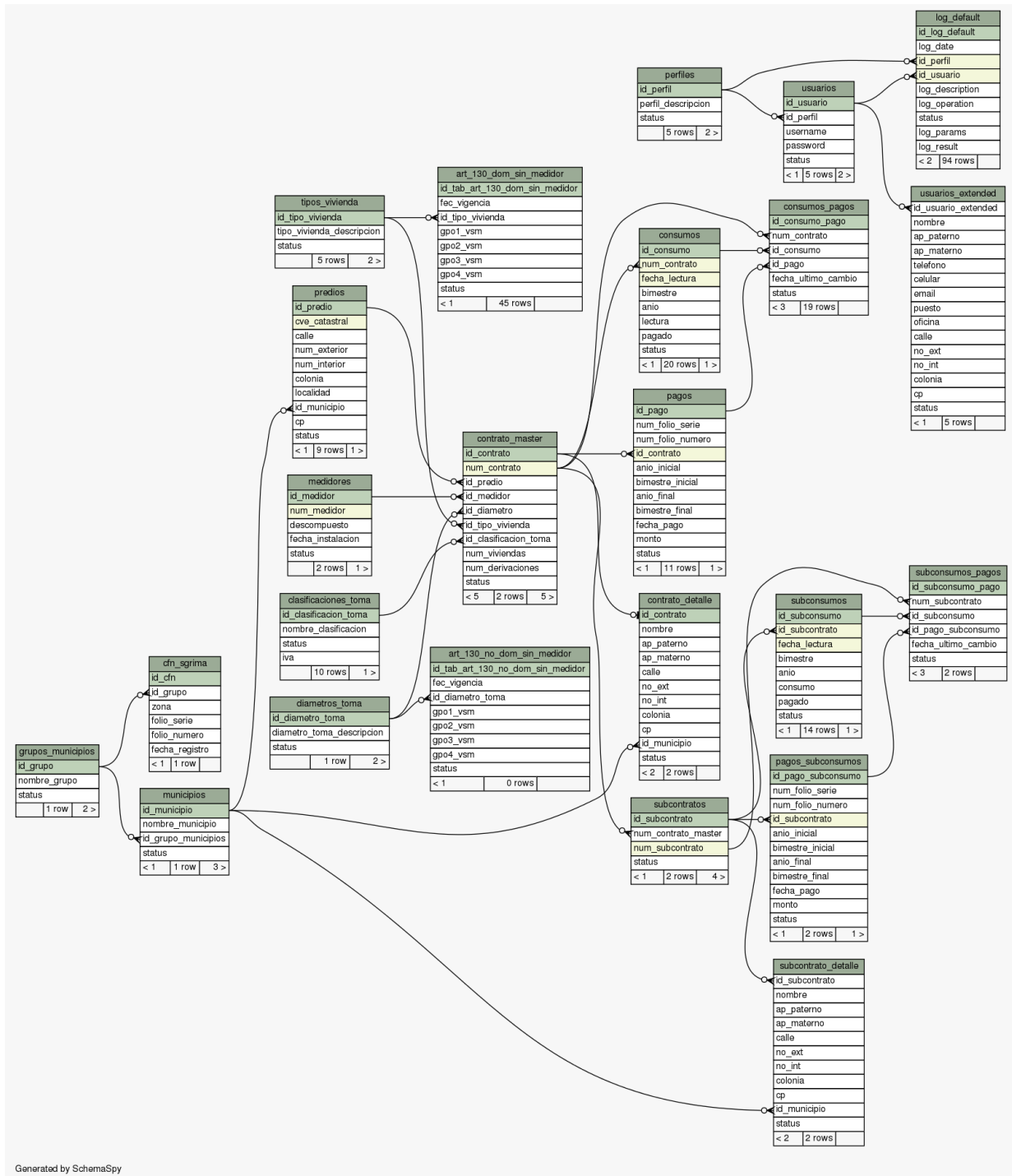


Figura 14: Estructura de la base de datos de la aplicación

Implementación

Tecnología utilizada

Software. Para la realización del presente proyecto se decidió utilizar solamente software libre y tratando de usar las versiones más recientes.

Debido a la naturaleza del proyecto y que tiene las aplicaciones web, algunas tecnologías de software se utilizaron para generar la vista del sistema, otras para implementar la lógica de negocio y funcionalidad del lado del servidor y otras como intercambio de datos entre capas de la aplicación.

Tecnologías para generar la vista (UI cliente)

- JSP
Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.
- EXTJS 4
Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM.
- Javascript
Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,3 basado en prototipos, imperativo, débilmente tipado y dinámico.

Tecnologías para lógica de negocio y funcionalidad del lado del servidor

- Java
Java es una plataforma con mucho soporte para el desarrollo de aplicaciones web. El soporte no refiere ni se limita solo a la cantidad de información que existe sobre ella en la red, sino que muchas empresas y organizaciones aportan bibliotecas de código abierto desarrolladas bajo esta plataforma, que facilitan el desarrollo de proyectos.
- Hibernate
Es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

- **Spring**
Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.2
- **Struts**
Es una herramienta de soporte para el desarrollo de aplicaciones bajo el patrón modelo-vista-controlador (MVC) bajo la plataforma J2EE. Struts se desarrollaba como parte del proyecto Jakarta de la Fundación Apache, pero actualmente es un proyecto independiente conocido como Apache Struts.
- **Apache Tomcat**
También conocido como Jakarta Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.
- **PostgreSQL 8.4**
Es un sistema gestor de base de datos relacional orientado a objetos.

Tecnologías para intercambio de datos

- **JSON**
Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Hardware. Para el desarrollo de este proyecto se dispuso de dos computadoras cuyas características se listan a continuación:

Componente	Computadora 1	Computadora 2
Procesador	AMD Turion 64x2	Core i3
Memoria RAM	4 GB	4 GB
Disco Duro	512 GB	256 GB
Sistema Operativo	Ubuntu 11.10	Ubuntu 11.10

Conclusiones

De acuerdo a los objetivos planteados en el proyecto, llegamos a la conclusión de que aun es necesario un avance en los sistemas municipales que ayude en la gestión de sus actividades de recaudación de impuestos.

Lo ideal es que exista un estandar en cuanto a esta actividad y el objetivo de este proyecto es justamente proponer un sistema estandarizado de cobro de impuestos con la misma base de información.

Por lo tanto consideramos que el objetivo del proyecto se cumple al ser este un sistema cuya base es la ley que aplica para todos los municipios.

Bibliografía

<http://www.cddiputados.gob.mx/POLEMEX/leyes/Ley08.html>

Visitado el 26 de Octubre de 2009

<http://java.sun.com/j2ee/overview.html>

Visitado el 26 de Octubre de 2009

<http://uml.sourceforge.net/>

Visitado el 26 de Octubre de 2009

<http://www.eclipse.org/>

Visitado el 26 de Octubre de 2009

<http://www.mysql.com/>

Visitado el 26 de Octubre de 2009

<http://struts.apache.org/>

Visitado el 26 de Octubre de 2009

<http://tomcat.apache.org/>

Visitado el 26 de Octubre de 2009

<http://www.opensymphony.com/quartz/>

Visitado el 4 de Noviembre de 2009

<https://www.hibernate.org/>

Visitado el 4 de Noviembre de 2009

<http://www.springsource.org/>

Visitado el 4 de Noviembre de 2009

<http://creativecommons.org/>

Visitado el 4 de Noviembre de 2009

<http://logging.apache.org/log4j/1.2/index.html>

Visitado el 4 de Noviembre de 2009

http://jasperforge.org/website/jasperreportswebsite/trunk/index.html?group_id=252

Visitado el 4 de Noviembre de 2009

<http://www.merida.gob.mx/>

Visitado el 26 de Octubre de 2009

<http://www.guadalajara.gob.mx/>

Visitado el 26 de Octubre de 2009

<http://www.sanpedro.gob.mx/portal.asp>

Visitado el 26 de Octubre de 2009

Referencias

Hibernate

<http://www.onjava.com/pub/a/onjava/2005/09/21/what-is-hibernate.html>

Spring

<http://www.springsource.org/spring-framework>

Struts

<http://www.onjava.com/pub/a/onjava/2005/11/02/what-is-struts.html>

JSON

<http://en.wikipedia.org/wiki/JSON>

Apache Tomcat

http://www.omii.ac.uk/docs/2.3.1/reference/tomcat/what_is_tomcat_.htm

PostgreSQL

<http://en.wikipedia.org/wiki/PostgreSQL>

ExtJS

http://en.wikipedia.org/wiki/Ext_JS

JavaScript

<http://javascript.about.com/od/reference/p/javascript.htm>

JSP

http://en.wikipedia.org/wiki/JavaServer_Pages

Casos de uso

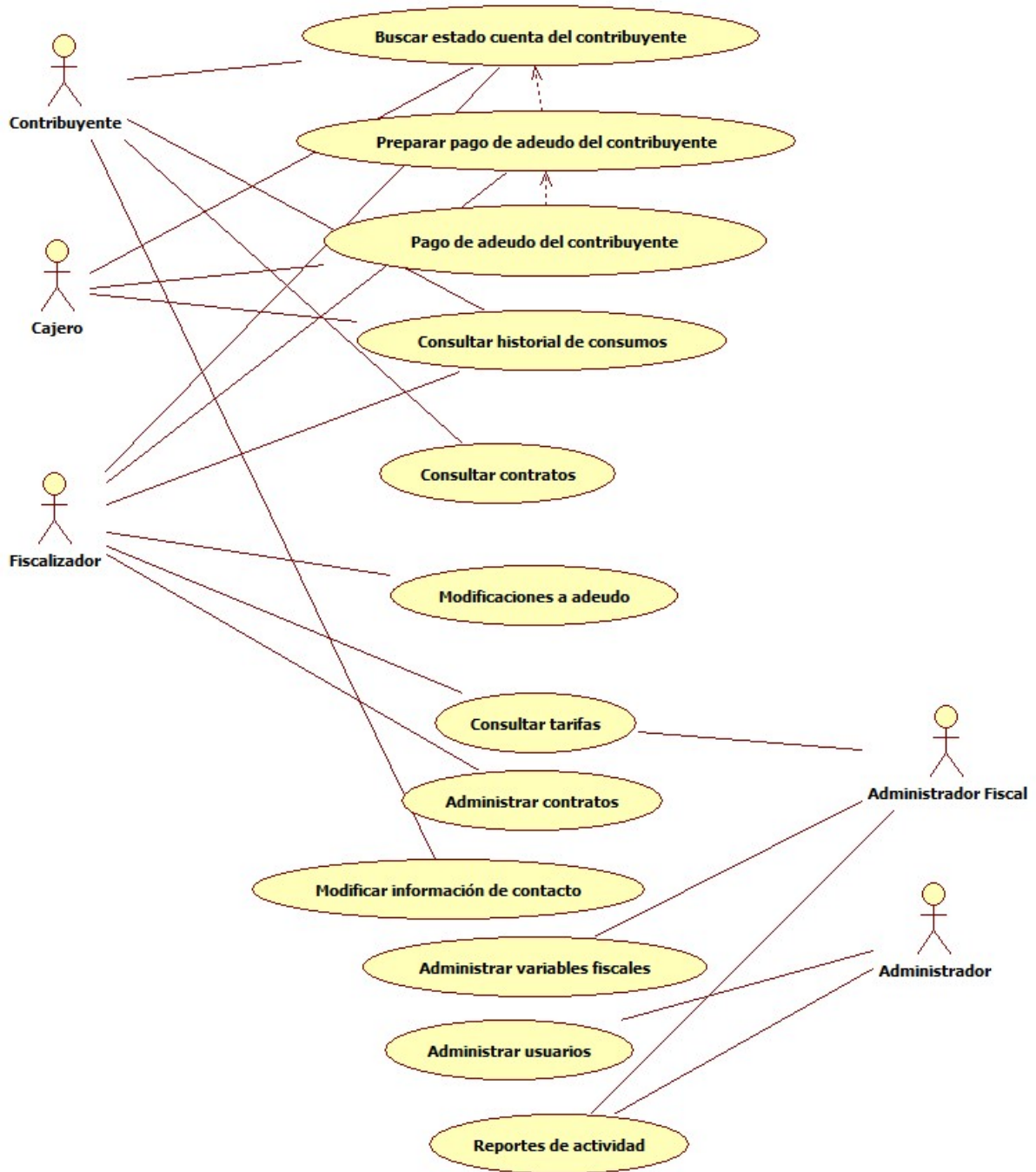


Figura 1: Todos los usuarios pasan por el caso de uso de Autenticación (el cual se omite en el diagrama) para poder hacer uso del sistema.

Caso de uso: Buscar estado cuenta del contribuyente

Roles: Contribuyente, fiscalizador.

Precondiciones: Usuario autenticado que esta buscando la información del estado de cuenta.

Postcondiciones: Se muestra la información relacionada con la cuenta del *contribuyente*.

Flujo básico:

5. El usuario da click en la opción del menú.
6. El sistema muestra la forma para ingresar por alguno de los metodos de busqueda disponibles.
7. El usuario ingresa la información en la forma y da click en el botón de Enviar.
8. El sistema muestra la información general de la cuenta y el listado de adeudos pendientes del *contribuyente* (en caso de haberlos).

Flujos alternos:

4. El *contribuyente* solicita una impresión.
 - a. El *fiscalizador* puede dar click en el icono “imprimir adeudos” para imprimir el resultado de la busqueda de estado de cuenta del contribuyente
5. El estado de cuenta es del *contribuyente* equivocado (error de captura de parametros de busqueda).
 - a. Se repite la operación desde el paso 1 del flujo básico.
6. Los datos ingresados en la búsqueda no regresaron ningun resultado.
 - a. El sistema informa al usuario que la información ingresada no existe en el sistema y que la verifique, dejandolo en la misma pantalla de forma de búsqueda.

Observaciones:

En algun punto el mismo contribuyente podria imprimir el resultado de la busqueda, esto dependiendo de los requisitos de la dependencia que ocupe del sistema.

Caso de uso: Preparar pago de adeudo del contribuyente

Casos de uso relacionados: Buscar estado de cuenta del contribuyente

Roles: Fiscalizador

Precondiciones: Usuario autenticado y dentro del flujo básico del caso de uso Buscar estado de cuenta del *contribuyente*

Postcondiciones: El contribuyente obtiene el formato de pago para el o los adeudos que elige pagar en una sola exhibición.

Flujo básico:

7. El *contribuyente* una vez que vio el listado de adeudos puede informar al *fiscalizador* que va a pagar un determinado número de adeudos.
8. El *fiscalizador* elige cual es el adeudo mas reciente de lo elegidos para pagar y da click en el icono de “pagar” a la derecha de la fila del adeudo elegido.
9. El sistema muestra cual es el monto total de adeudos que debe pagar el contribuyente desde el más antiguo hasta el adeudo elegido.

10. El *contribuyente* confirma que desea pagar los adeudos por la cantidad motrada.
11. El *fiscalizador* da click en la opción “Aceptar”
12. El sistema muestra el formato de pago para impresión

Flujos alternos:

2. El *contribuyente* elige no pagar los adeudos por esa cantidad si no por otra.
 - a. El *fiscalizador* puede cancelar y realizar los mismos pasos desde el punto 2.

Observaciones:

En algun punto el mismo contribuyente podria imprimir su formato de pago, esto dependiendo de los requisitos de la dependencia que ocupe del sistema.

Caso de uso: Pago de adeudo del contribuyente

Casos de uso relacionados: Preparar pago de adeudo del contribuyente

Roles: Cajero

Precondiciones: El *cajero* esta autenticado en el sistema y el *contribuyente* tiene el formato de pago generado en el caso de uso Preparar pago de adeudo del contribuyente que quiere pagar.

Postcondiciones: El *cajero* recibe el pago correspondiente al formato de pago y el pago queda realizado y registrado en el sistema.

Flujo básico:

5. El *contribuyente* acude con el *cajero* para hacer el pago por la cantidad que se indica en el formato de pago.
6. El *cajero* ingresa los datos de la cuenta del contribuyente para identificar el formato de pago dentro del sistema.
7. El *cajero* cobra al cliente la cantidad indicada en el recibo y da click en el botón de “Pagar”
8. El sistema registra el pago y muestra al *cajero* la confirmación del pago.

Flujos alternos:

2. El *contribuyente* decide no hacer el pago.
 - a. El *cajero* solamente debe de salir del listado de pagos pendientes del *contribuyente* ingresado y continuar su labor normalmente.
 - b. El sistema hace un corte al final del día de los formatos de pago no realizados para restablecer los adeudos que estaban en proceso de ser pagados.

Observaciones:

Caso de uso: Consultar historial de consumos

Roles: Contribuyente, cajero, fiscalizador

Precondiciones: Usuario autenticado que quiere consultar el historial de consumos del contribuyente.

Postcondiciones: Listado con información del historial del contribuyente. Se puede revisar que

consumos se han pagado y cuando.

Flujo básico:

3. El usuario da click en la opción del menu para buscar el historial de consumos.
4. El sistema muestra el listado de consumos en orden de antigüedad

Flujos alternos:

4. El *contribuyente* solicita una impresión.
 - a. El *usuario* puede dar click en el icono “imprimir adeudos” para imprimir el resultado de la búsqueda de estado de cuenta del contribuyente
5. El estado de cuenta es del *contribuyente* equivocado (error de captura de parametros de búsqueda).
 - a. Se repite la operación desde el paso 1 del flujo básico.
6. Los datos ingresados en la búsqueda no regresaron ningun resultado.
 - a. El sistema informa al usuario que la información ingresada no existe en el sistema y que la verifique, dejandolo en la misma pantalla de forma de búsqueda.

Observaciones:

En algun punto el mismo contribuyente podria imprimir el resultado de la búsqueda, esto dependiendo de los requisitos de la dependencia que ocupe del sistema.

Caso de uso: Consultar contratos

Roles: Contribuyente

Precondiciones: Usuario autenticado y que quiere realizar consulta de los contratos donde es el titular de la cuenta.

Postcondiciones: El usuario puede ver la información de los contratos que estan a su nombre.

Flujo básico:

4. El usuario da click en la opción del menú para consultar sus contratos.
5. El sistema muestra un listado de contratos a nombre del usuario que esta en sesión.
6. El usuario puede ver el detalle de los contratos a su nombre al dar click en el icono de información de cada contrato dentro del listado.

Flujos alternos:

Observaciones:

El sistema puede incluir mas funcionalidades para el contribuyente para administrar sus contratos dependiendo de los requisitos finales de la dependencia que ocupe el sistema.

Caso de uso: Modificaciones a adeudo

Roles: Fiscalizador

Precondiciones: Usuario autenticado y con autorización para hacer modificaciones a un adeudo dentro de los consumos de un contribuyente.

Postcondiciones: El usuario realiza la modificación al consumo para ajustar su calculo de acuerdo a lo autorizado.

Flujo básico:

5. El *fiscalizador* ingresa a la opción de menú para modificar consumo.
6. El sistema muestra los consumos pendientes de pagar.
7. El *fiscalizador* escoge el consumo a modificar y edita el campo adecuado y da click en el botón de guardar.
8. El sistema realiza el ajuste y registra el cambio.

Flujos alternos:

2. El usuario comete un error en la modificación del consumo.
 - a. El usuario puede repetir el proceso desde el punto 3 del flujo básico.

Observaciones:**Caso de uso: Consultar tarifas**

Roles: Administrador fiscal, fiscalizador

Precondiciones: Usuario autenticado y que necesita revisar las tarifas vigentes o de años anteriores.

Postcondiciones: El usuario puede ver las tarifas vigentes o de años anteriores.

Flujo básico:

3. El usuario da click en la opción del menú consultar tarifas
4. El sistema muestra las tarifas vigentes.

Flujos alternos:

3. El usuario quiere consultar las tarifas de años anteriores.
 - a. El usuario puede acceder a la opción del menú para consultar tarifas por año.
 - b. El sistema muestra una forma con un combobox para elegir el año.
 - c. El usuario escoge el año y da click en "Aceptar"
 - d. El sistema muestra las tarifas correspondientes al año seleccionando.
4. El usuario quiere imprimir el listado de tarifas.
 - a. El usuario puede dar click en el botón imprimir tarifas de las tarifas listadas.

Observaciones:**Caso de uso: Administrar contratos**

Roles: Fiscalizador

Precondiciones: Usuario autenticado y con la necesidad de Agregar, modificar, suspender o eliminar contratos.

Postcondiciones: El usuario agrega, modifica, suspende o elimina un contrato del sistema.

Flujo básico:

2. El usuario da click en la opción de menú para Agregar, modificar, suspender o eliminar un contrato.
 - a. Agregar
 - i. El sistema muestra una forma con los campos necesarios a llenar para crear el nuevo contrato.

- ii. El usuario llena la forma y da click en “Enviar”
- b. Modificar
 - i. El sistema muestra una forma de búsqueda para ingresar los datos del contrato que se quiere modificar.
 - ii. El usuario ingresa los datos del contrato que quiere editar y da click en el botón “Buscar”.
 - iii. El sistema muestra los datos del contrato en una forma con campos editables.
 - iv. El usuario hace los cambios al contrato y da click en la opción de “Guardar”
- c. Suspender
 - i. El sistema muestra una forma de búsqueda para ingresar los datos del contrato que se quiere suspender.
 - ii. El usuario ingresa los datos del contrato que se quiere suspender y da click en el botón buscar.
 - iii. El sistema muestra la información del contrato.
 - iv. El usuario da click en el botón “Suspender”.
 - v. El sistema muestra una forma para ingresar la razón de la suspensión.
 - vi. El usuario captura la razón y da click en “Continuar”
 - vii. El sistema cambia el estado del contrato a suspendido.
- d. Eliminar
 - i. El sistema muestra una forma de búsqueda para ingresar los datos del contrato que se quiere eliminar.
 - ii. El usuario ingresa los datos del contrato que se quiere eliminar y da click en el botón buscar.
 - iii. El sistema muestra la información del contrato.
 - iv. El usuario da click en el botón “Eliminar”.
 - v. El sistema muestra una forma para ingresar la razón de eliminación.
 - vi. El usuario captura la razón y da click en “Continuar”
 - vii. El sistema cambia el estado del contrato a eliminado.

Flujos alternos:

- 3. El usuario ya no quiere crear un nuevo contrato.
 - a. El usuario puede dar click en la opción cancelar o simplemente cambiar a otra opción del menú.
- 4. El usuario se equivoca en la actualización del contrato.
 - a. El usuario puede continuar con la modificación de contrato y continuar haciendo cambios hasta que de click en otra opción del menú.

Observaciones:

Otras opciones a considerar en funcionamiento con la suspensión del contrato son la activación del contrato y bajo que criterios se suspende/activa un contrato.

Caso de uso: Modificar información de contacto

Roles: Contribuyente

Precondiciones: Usuario autenticado y que quiere modificar sus datos de contacto.

Postcondiciones: Usuario pudo modificar sus datos de contacto.

Flujo básico:

7. El usuario da click en la opción información de contacto del menú.
8. El sistema muestra en pantalla la información actual que tiene el usuario.
9. El usuario da click en actualizar información.
10. El sistema muestra la forma editable para modificar los campos disponibles.
11. El usuario modifica los campos necesarios y da click en “Guardar”
12. El sistema actualiza la información de la forma y muestra la pantalla del paso 2.

Flujos alternos:

2. El usuario ya no quiere modificar su información de contacto.
 - a. El usuario puede cancelar la operación dando click en el botón de “Cancelar”

Observaciones:

Caso de uso: Administrar variables fiscales

Roles: Administrador Fiscal

Precondiciones: Usuario autenticado y con la necesidad de actualizar la variables fiscales por algun ajuste a la ley.

Postcondiciones: El usuario logra actualizar las variable fiscales.

Flujo básico:

5. El usuario da click en la variables fiscal que desee administrar del menú.
6. El sistema muestra un listado de los valores actuales de la variable fiscal seleccionada.
7. El usuario puede modificar las variables fiscales como un catálogo con operaciones de agregar, modificar, eliminar.
8. El sistema guarda la información modificada de las variables fiscales en el sistema.

Flujos alternos:

Observaciones:

Caso de uso: Administrar usuarios

Roles: Adminisitrador del sistema

Precondiciones: Usuario autenticado que quiere administrar a los usuarios del sistema.

Postcondiciones: Se han agregado, actualizado o eliminado usuarios del sistema.

Flujo básico:

4. El *usuario* escoge de las opciones del menú Administración la opción usuarios.
5. El sistema muestra el listado de usuarios actualmente activos en el sistema.
6. El usuario puede agregar, editar, eliminar.
 - a. Agregar
 - i. Al dar click en el botón agregar una nueva fila de usuario aparecera para poder ser editada en el listado de usuarios del sistema.
 - ii. El usuario llena los campos necesarios del nuevo usuario del sistema y da click en “Guardar”.
 - iii. El sistema registra al nuevo usuario.

- b. Editar
 - i. El usuario puede dar doble click sobre un usuario en el listado de usuarios para editar la información del usuario y posteriormente dar click en “Guardar” para guardar los cambios.
 - ii. El sistema registra los cambios al usuario.
- c. Eliminar
 - i. El usuario puede elegir un usuario de la lista dando click sobre el y posteriormente click en el botón de “Eliminar”.
 - ii. El sistema elimina al usuario del sistema.

Flujos alternos:

- 3. El usuario decide dejar de administrar los cambios en el listado de usuarios.
- 4. El usuario puede dar cancelar o cambiar a otra opción del menú para dejar la sección de administrar usuarios del sistema.

Observaciones:

Caso de uso: Reportes de actividad

Roles: Administrador, AdministradorFiscal

Precondiciones: Usuario autenticado y que necesita conocer reportes del sistema.

Postcondiciones: El usuario puede ver reportes del sistema.

Flujo básico:

- 3. El usuario selecciona el tipo de reporte del sistema del menú.
- 4. El sistema muestra el reporte en pantalla.

Flujos alternos:

- 2. El usuario quiere imprimir el reporte.
 - a. El usuario puede dar click en la opción imprimir reporte una vez que el reporte seleccionado aparezca en pantalla.

Observaciones:

Los reportes que se muestran para administrador y administrador fiscal son distintos debido a los permisos pero su funcionamiento es el mismo.

Diagrama de dominio

Este diagrama describe los principales objetos del negocio participantes en el sistema.

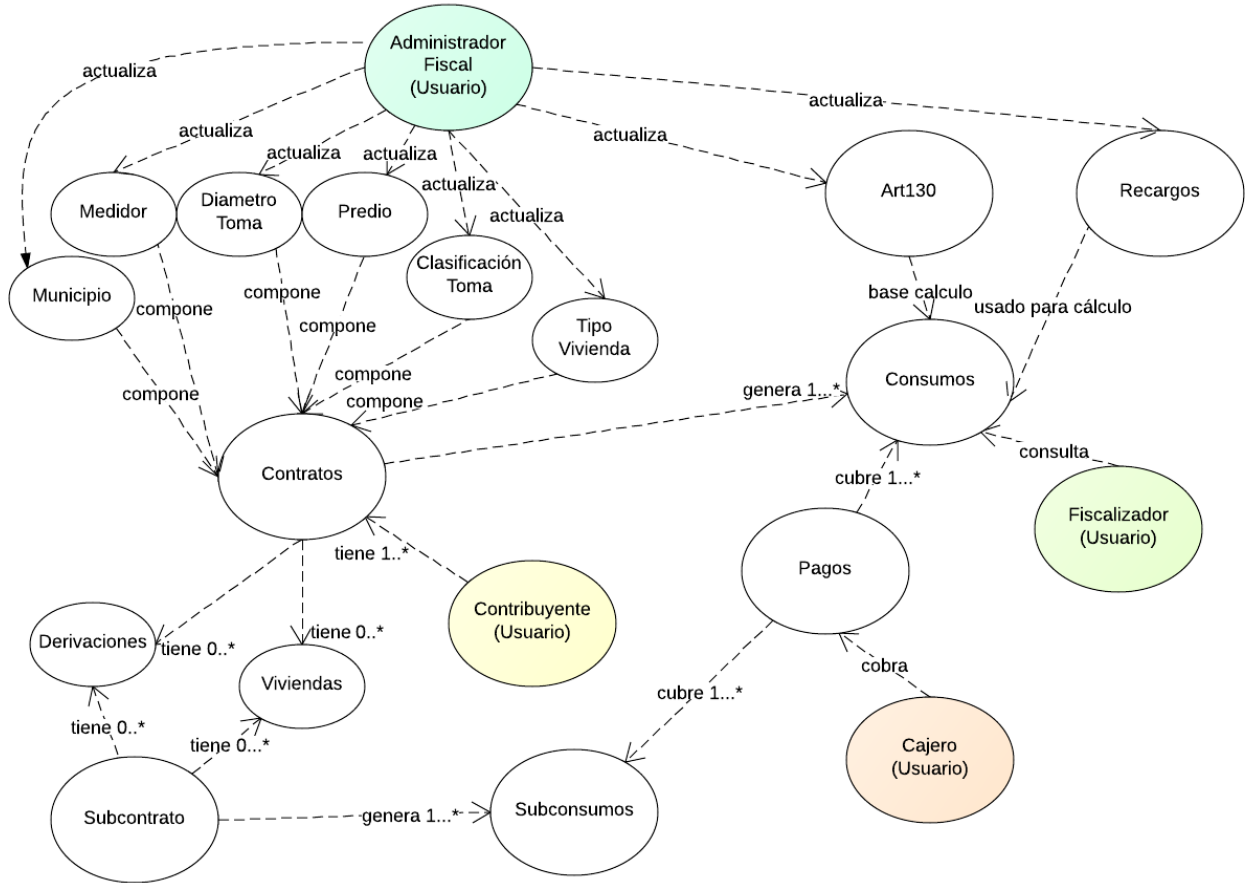


Figura 2: Diagrama de dominio

Diagramas de Navegación

Los diagramas de navegación están separados por Perfil, salvo el de Login que es común a todos.

Diagrama de navegación de sección **Principal**

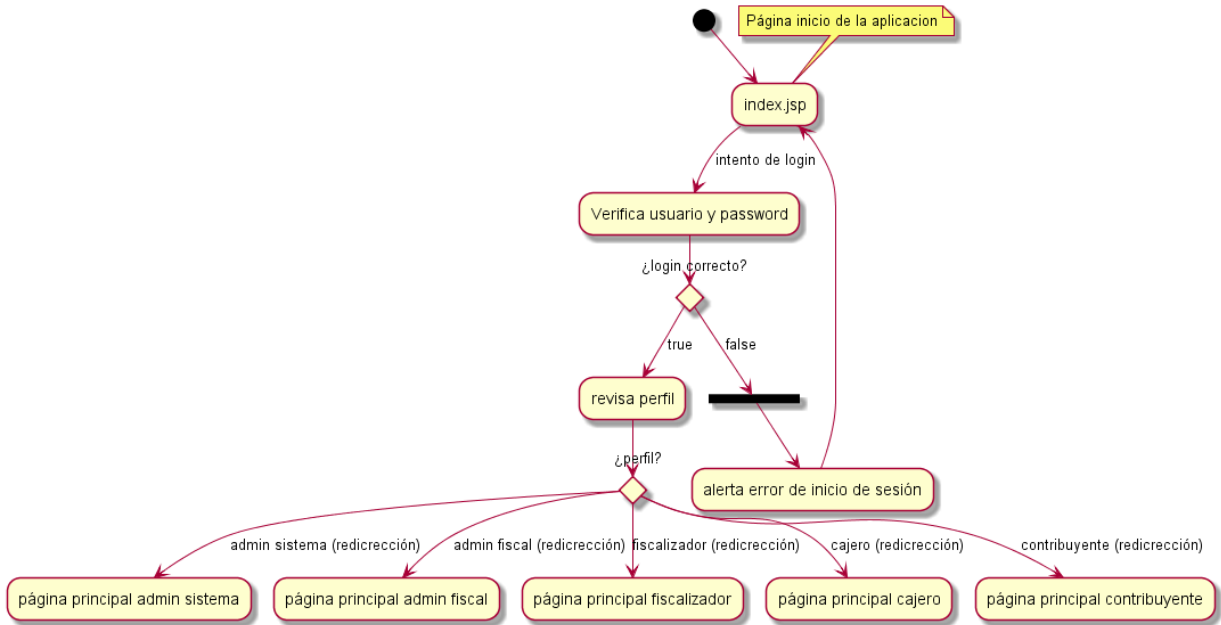


Figura 3: Diagrama de navegación Principal

Diagrama de navegación de sección **Administrador Fiscal**

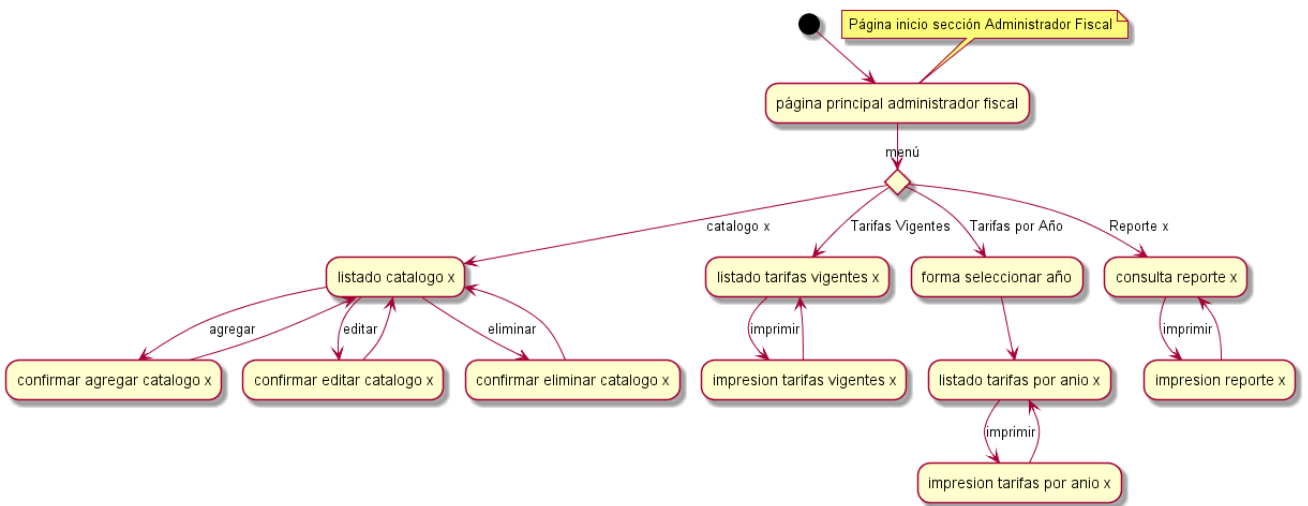


Figura 4: Diagrama de navegación Administrador Fiscal

Diagrama de navegación de sección **Fiscalizador**

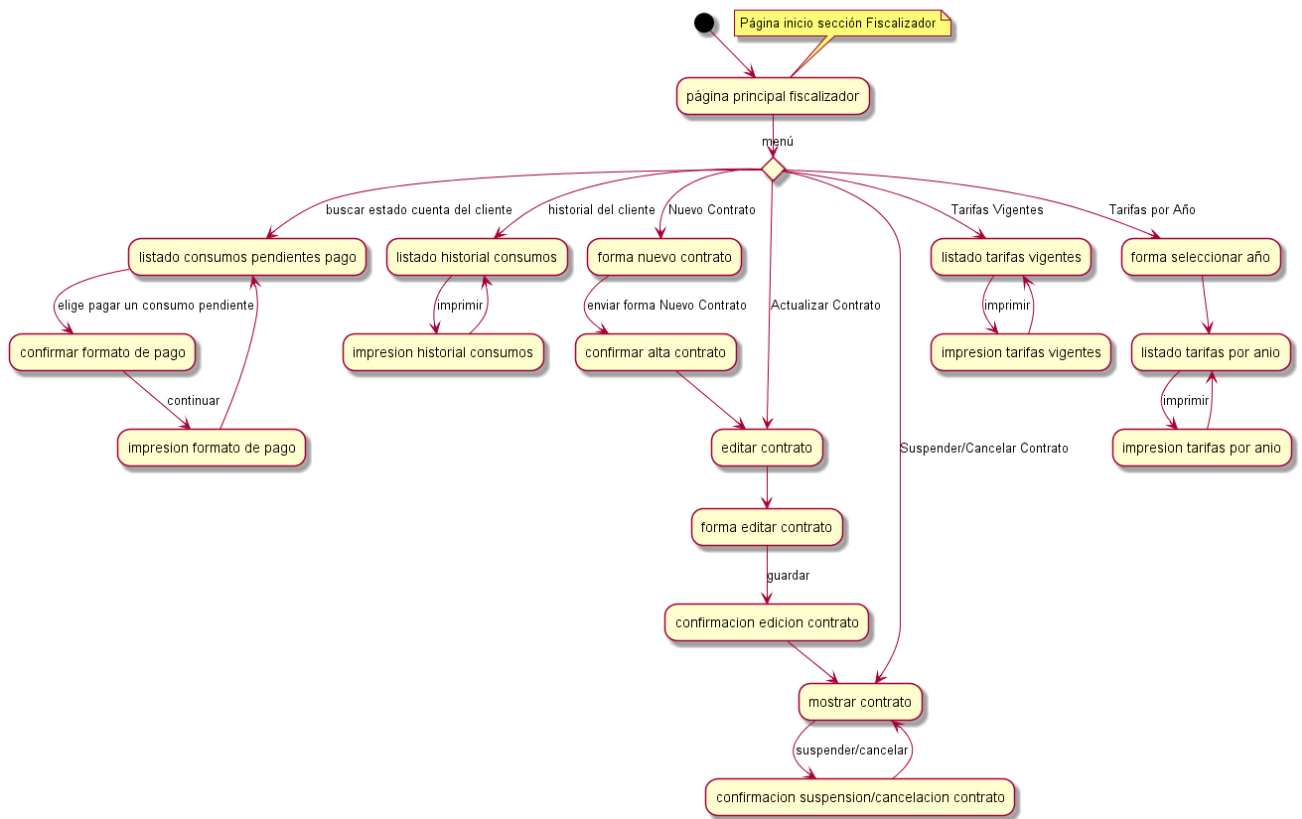


Figura 5: Diagrama de navegación Fiscalizador

Diagrama de navegación de sección **Cajero**

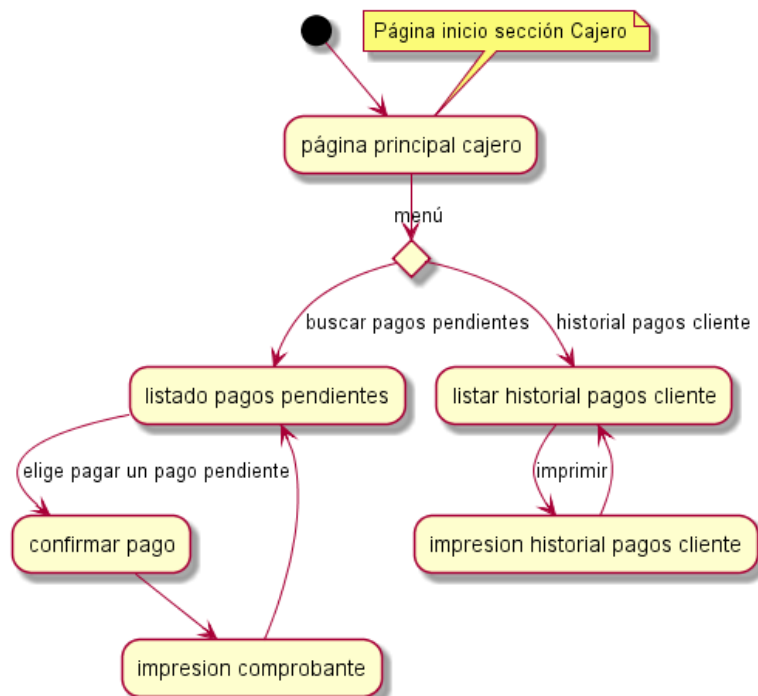


Figura 6: Diagrama de navegación Cajero

Diagrama de navegación de sección **Contribuyente**

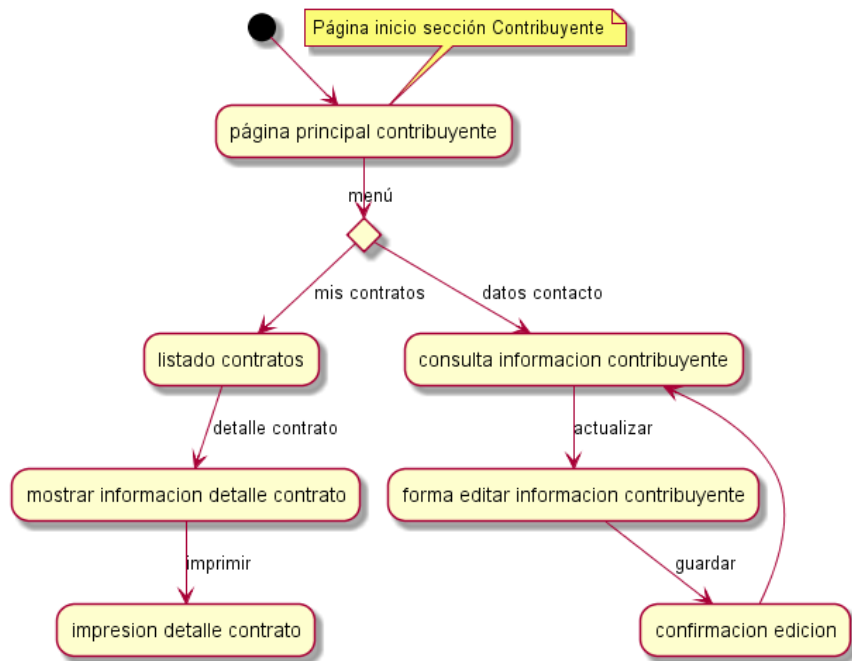


Figura 7: Diagrama de navegación Contribuyente

Diagramas de clases

Las clases en sus respectivos paquetes son las siguiente
Se da una descripción de las clases mas importantes.

- **mx.uam.azc.sgrima**

- InicializadorCatalogos

- **mx.uam.azc.sgrima.actions**

(Los actions son parte del modelo MVC de struts y se ocupan de fungir como controladores)

- AdminFiscalAction
- AdminFiscalArt130Action
- AdminFiscalCatalogosAction
- CajeroAction
- ContribAction
- FiscalAction
- LoginAction
- PrincipalAction
- SysAdminAction
- SysadminPerfilesAction
- SysadminUsuariosAction

- **mx.uam.azc.sgrima.busines**

(Son las interfaces para los administradores que contralan el acceso al modelo del sistema)

- *AdministradorActualizaciones*
- *AdministradorArt130*
- *AdministradorArt130DomConMedidor*
- *AdministradorArt130DomSinMedidor*
- *AdministradorArt130NoDomConMedidor*
- *AdministradorArt130NoDomSinMedidor*
- *AdministradorCalculosConsumoDCM*
- *AdministradorCfnSgrima*
- *AdministradorClasificacionToma*
- *AdministradorConsumo*
- *AdministradorConsumoPago*
- *AdministradorContratoDetalle*
- *AdministradorContratoMaster*
- *AdministradorDiametroToma*
- *AdministradorGrupoMunicipio*
- *AdministradorLogDefault*
- *AdministradorMedidor*
- *AdministradorMunicipio*
- *AdministradorPago*
- *AdministradorPagoSubconsumo*
- *AdministradorPerfil*
- *AdministradorPredio*
- *AdministradorRecargos*
- *AdministradorSalariosMinimos*
- *AdministradorSubconsumo*
- *AdministradorSubconsumoPago*
- *AdministradorSubcontrato*
- *AdministradorSubcontratoDetalle*
- *AdministradorTipoVivienda*
- *AdministradorUsuario*
- *AdministradorUsuarioExtended*
- *CalculosConsumoBO*
- *ConsumosBO*
- *PagosBO*

- **mx.uam.azc.sgrima.business.local**

(Implementación de las interfaces del modelo)

- AdministradorActualizacionesImpl
- AdministradorArt130DomConMedidorImpl
- AdministradorArt130DomSinMedidorImpl
- AdministradorArt130Impl
- AdministradorArt130NoDomConMedidorImpl
- AdministradorArt130NoDomSinMedidorImpl
- AdministradorCalculosConsumoDCMImpl
- AdministradorCfnSgrimaImpl
- AdministradorClasificacionTomaImpl
- AdministradorConsumoImpl
- AdministradorConsumoPagoImpl
- AdministradorContratoDetalleImpl
- AdministradorContratoMasterImpl
- AdministradorDiametroTomaImpl
- AdministradorGrupoMunicipioImpl
- AdministradorLogDefaultImpl

- AdministradorMedidorImpl
- AdministradorMunicipioImpl
- AdministradorPagoImpl
- AdministradorPagoSubconsumoImpl
- AdministradorPerfilImpl
- AdministradorPredioImpl
- AdministradorRecargosImpl
- AdministradorSalariosMinimosImpl
- AdministradorSubconsumoImpl
- AdministradorSubconsumoPagoImpl
- AdministradorSubcontratoDetalleImpl
- AdministradorSubcontratoImpl
- AdministradorTipoViviendaImpl
- AdministradorUsuarioExtendedImpl
- AdministradorUsuariosImpl
- CalculosConsumoBOImpl
- ConsumosBOImpl
- PagosBOImpl

- **mx.uam.azc.sgrima.data**

(Clases que modelan la Base de datos y otros objetos necesarios en el sistema)

- ActualizacionDTO
- Art130DomConMedidorDTO
- Art130DomSinMedidorDTO
- Art130NoDomConMedidorDTO
- Art130NoDomSinMedidorDTO
- CalculosConsumoDTO
- CalculosSubconsumoDTO
- CfnSgrimaDTO
- ClasificacionTomaDTO
- ConsumoDTO
- ConsumoPagoDTO
- ConsumoPorPagarDTO
- ContratoDetalleDTO
- ContratoMasterDTO
- DiametroTomaDTO
- GrupoMunicipioDTO
- LogDefaultDTO
- MedidorDTO
- MunicipioDTO
- PagoDTO
- PagoSubconsumoDTO
- PerfilDTO
- PredioDTO
- RecargoDTO
- SalarioMinimoDTO
- SubconsumoDTO
- SubconsumoPagoDTO
- SubconsumoPorPagarDTO
- SubcontratoDetalleDTO
- SubcontratoDTO
- TipoViviendaDTO
- UsuarioDTO
- UsuarioExtendedDTO

- **mx.uam.azc.sgrima.interceptors**

(Estas clases sirven para validar la sesión de cada tipo de usuario en cada petición)

- IntecceptorAutenticacionAdminFiscal
- IntecceptorAuntenticacionAdmin
- IntecceptorAutenticacionCajero
- InterceptorAutenticacionContribuyente
- InterceptorAutenticacionFiscal

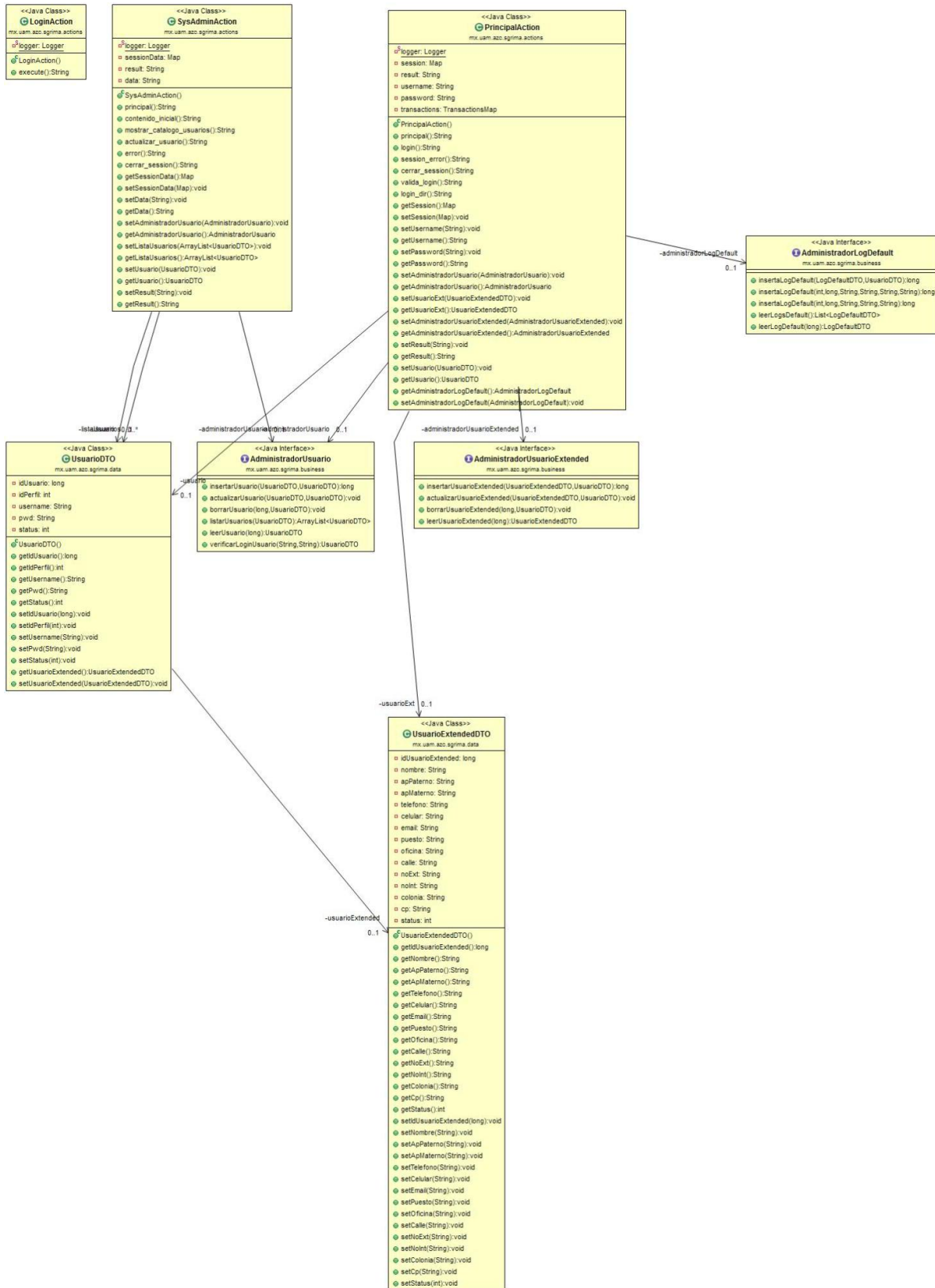
- **mx.uam.azc.sgrima.util**

(Clases auxiliares)

- ConstantesPDF
- ConstantesSistema
- GenerarPDF
- ToStringDTOs
- Transaction
- TransactionsMap

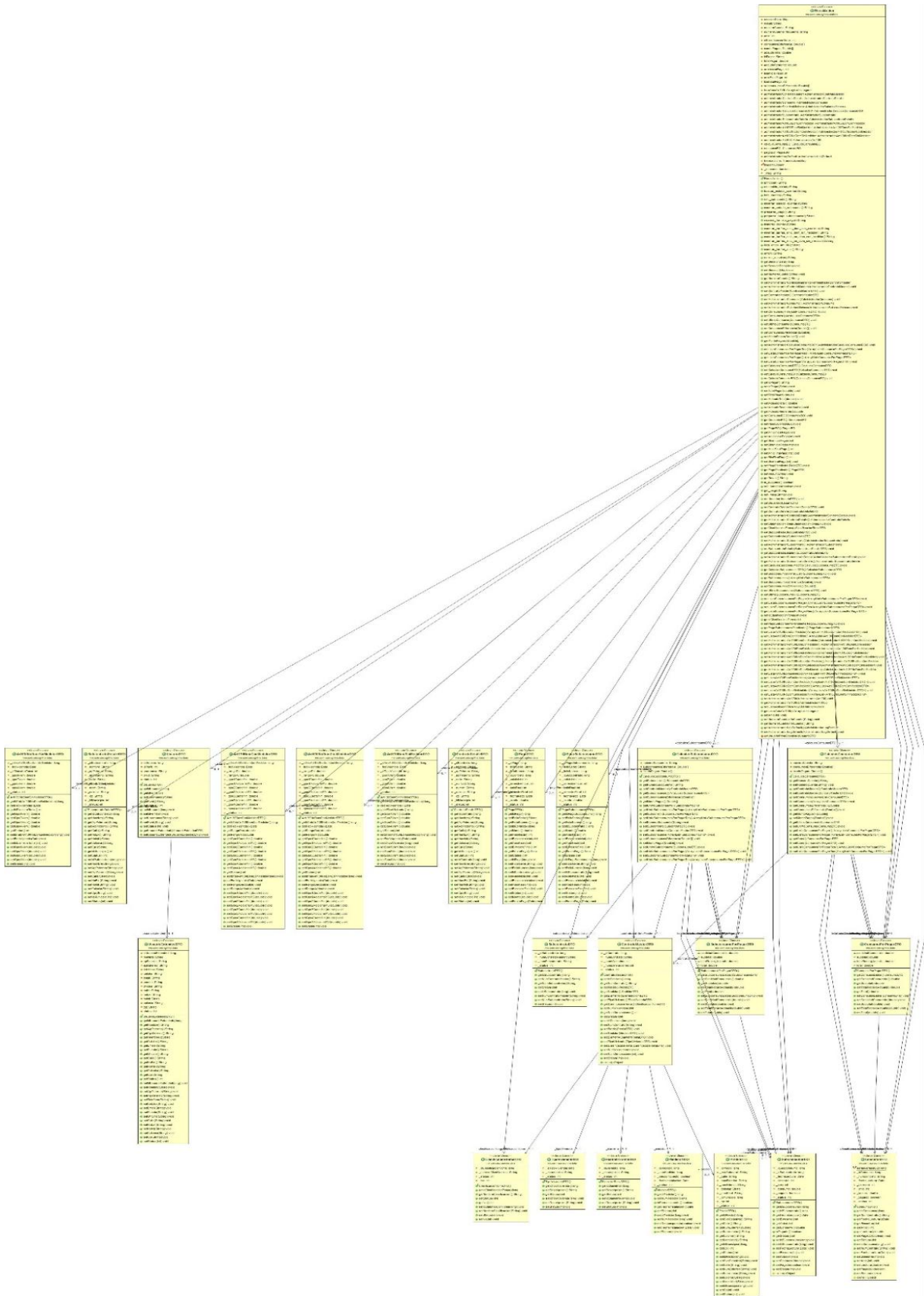
Diagrama de clases para Administrador

Figura 8



Diagramas de clases para Administrador Fiscal

Figura 9



Diagramas de Secuencia

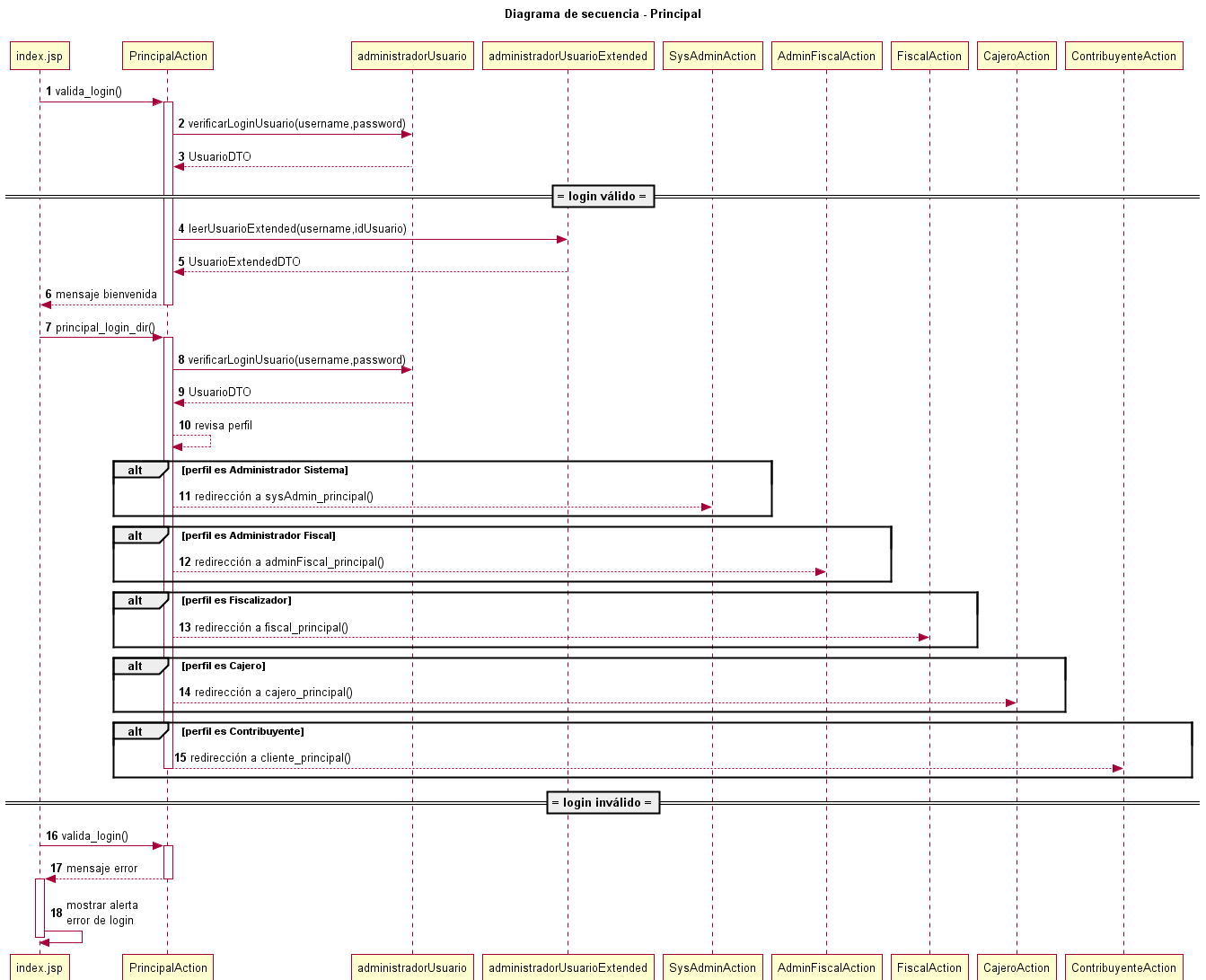


Figura 14: Diagrama de Secuencia Principal

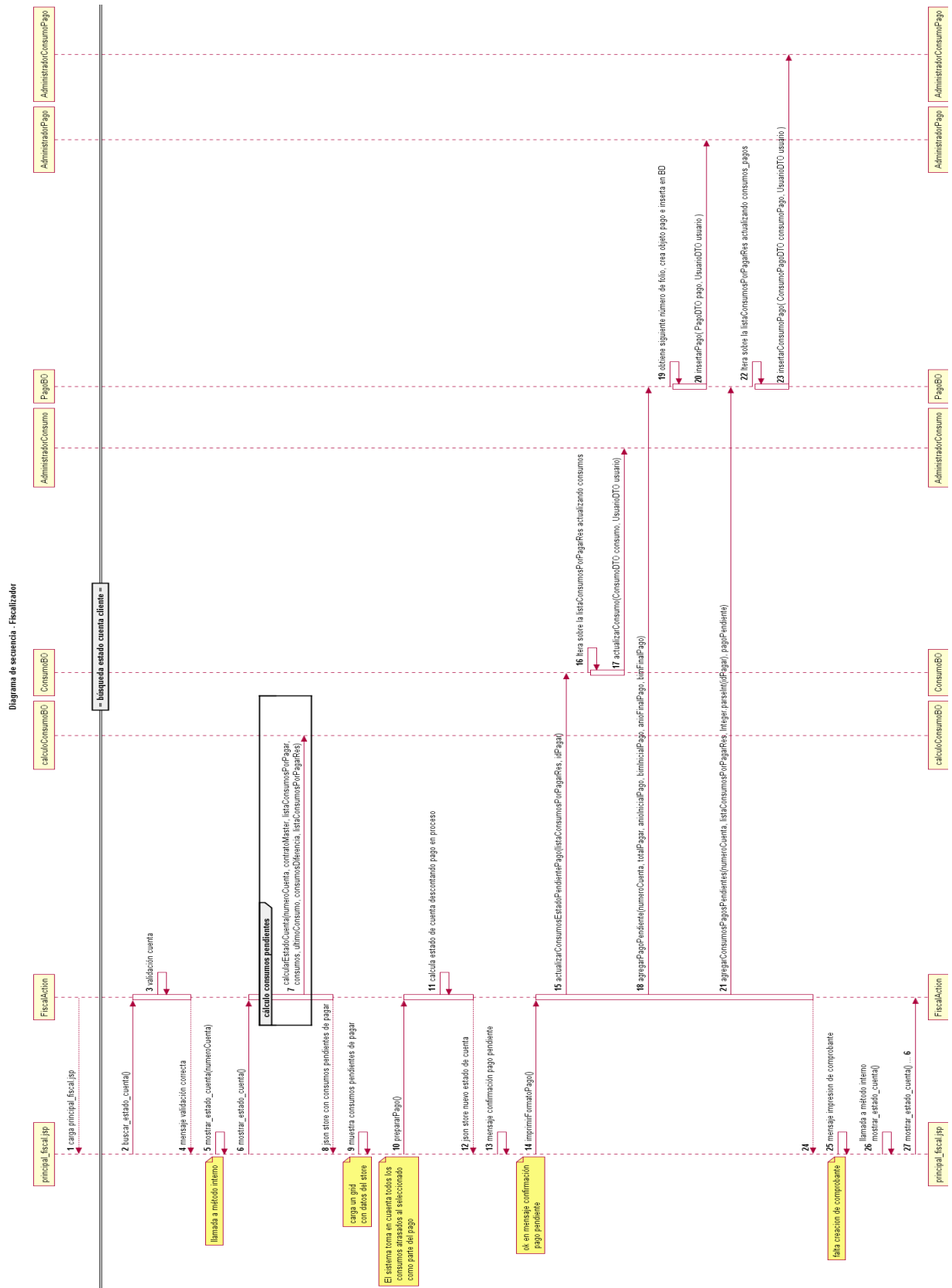


Figura 15: Diagrama Secuencia Fiscalizador

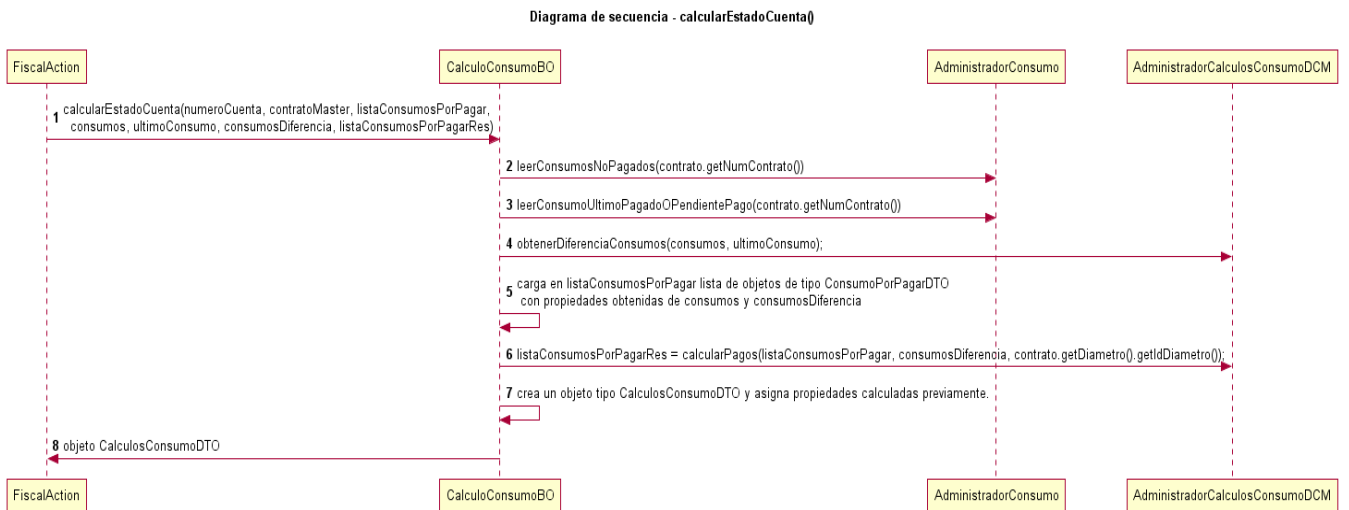


Figura 16: Diagrama Secuencia calculoEstadoCuenta

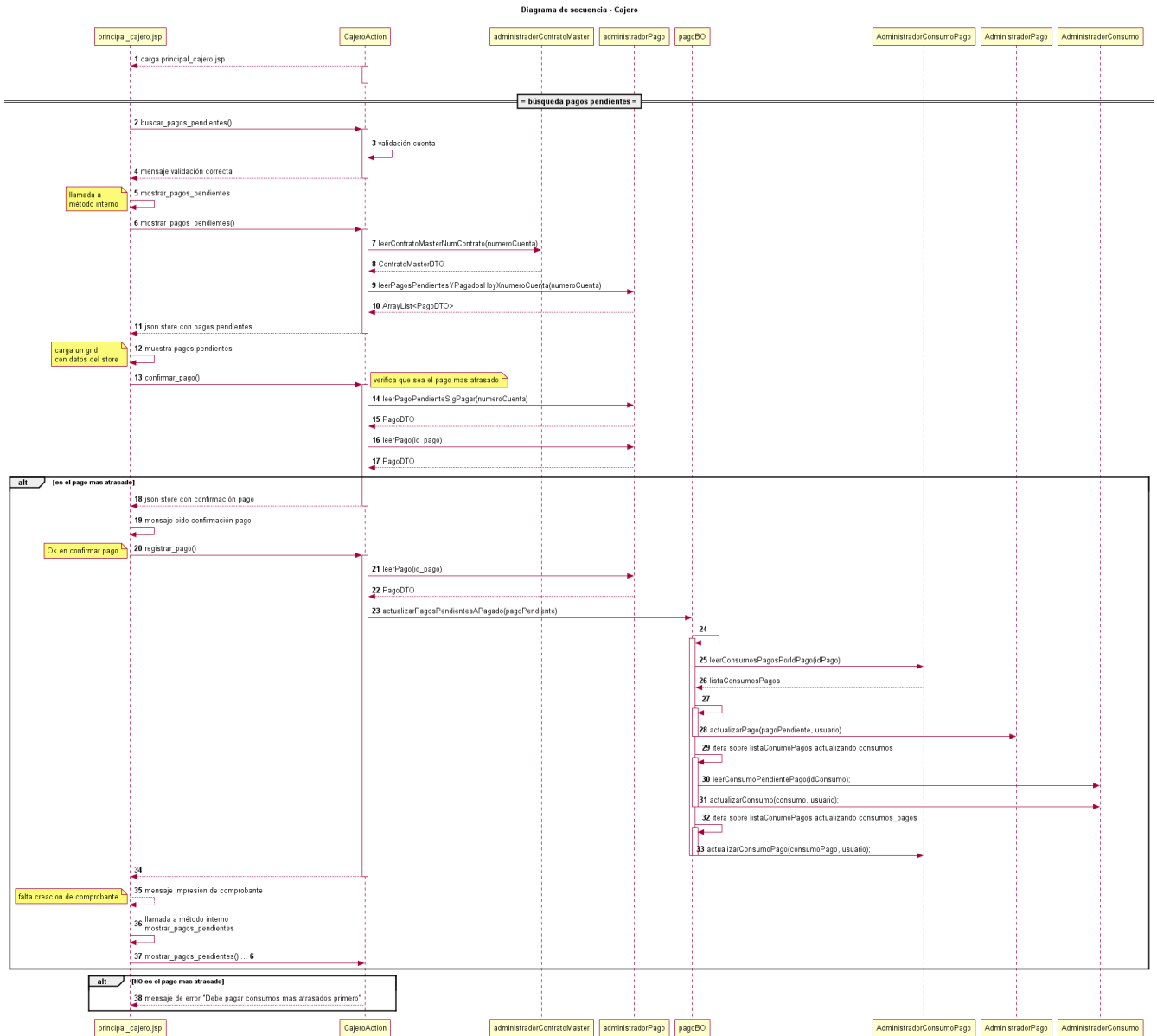


Figura 17: Diagrama Secuencia Cajero

Estructura de la base de datos

A continuación, se describen cada una de las tablas de la base de datos de Sgrima.

Nombre de la tabla	Descripción
art_130_dom_con_medidor	Define número de salarios mínimos que se deben cobrar basándose en el rango de m3 consumidos y en el grupo de municipio al que pertenece(el municipio) para las tomas domésticas con medidor.
art_130_dom_sin_medidor	Define el número de salarios mínimos que se cobran basándose en el grupo de municipio al que pertenece(el municipio) para tomas domesticas sin medidor.
art_130_no_dom_con_medidor	Define número de salarios mínimos que se deben cobrar basándose en el rango de m3 consumidos y en el grupo de municipio al que pertenece(el municipio) para las tomas no domésticas con medidor.
art_130_no_dom_sin_medidor	Define el número de salarios mínimos que se cobran basándose en el grupo de municipio al que pertenece(el municipio) para tomas no domésticas sin medidor.
cfn_sgrima	Información de la configuración actual del sistema.
clasificaciones_toma	Indica el tipo de contrato de la vivienda y por lo tanto cómo será tratado el contrato y la forma de hacer los cálculos en el sistema.
consumos	Lleva registro de la fecha en que se toma el consumo de una vivienda o toma, para las que tienen medidor. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado
consumos_pagos	Lleva registro de la relacion de consumos y a que pago corresponden. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado. La fecha hace referencia a la ultima transaccion
contrato_detalle	Guarda informacion de facturacion para viviendas o derivaciones que están relacionadas con un contrato_master.
contrato_master	Guarda información detallada del contrato para realizar cálculos.

diametros_toma	Los diferentes tipos de toma(diferenciados por el tamaño de la toma) manejados por el sistema.
grupos_municipios	Detalla la información de grupos de municipios.
log_default	Almacena la actividad de los usuarios a lo largo de la aplicación.
medidores	Detalla la información relacionada con un medidor y su estado.
municipios	Detalle de la información de cada municipio.
pagos	Información sobre pagos_subconsumos realizados de un subcontrato. Un pago cubre por lo menos 1 consumo. Status: 1 = 2 = pendiente 3 = pagado
pagos_subconsumos	Lleva registro de la relación de pagos_subconsumos y a que pago corresponden. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado. La fecha hace referencia a la última transacción
perfiles	Los diferentes perfiles de usuario que están involucrados en el sistema.
predios	Información del predio, el predio está asociado a con un contrato_master.
recargos	indica la tasa a cobrar por recargos cuando se genera un atraso.
salarios_minimos	indica el valor del del salario mínimo para la zona.
subconsumos	Llevar control de los consumos cuando se hace división de pagos como en el caso de Mixto con Medidor. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado
subconsumos_pagos	Lleva registro de la relación de subconsumos y a que pago corresponden. status: 1 = pendiente de pago 2 = proceso de pago 3 = pagado. La fecha hace referencia a la ultima transaccion
subcontrato_detalle	Información complementaria del subcontrato para realizar

	cálculos.
subcontratos	Guarda información detallada del subcontrato para realizar cálculos.
tipos_vivienda	Indica el tipo de vivienda el cual se usa para el cálculo del consumo.
usuarios	Información de usuarios del sistema.
usuarios_extended	Información complementaria para los usuarios del sistema.

En la figura 18, se muestra la estructura completa de la base de datos. Por cuestiones de espacio la imagen fue escalada al espacio de la presente página. La estructura y la imagen se adjuntan como parte del entregable.

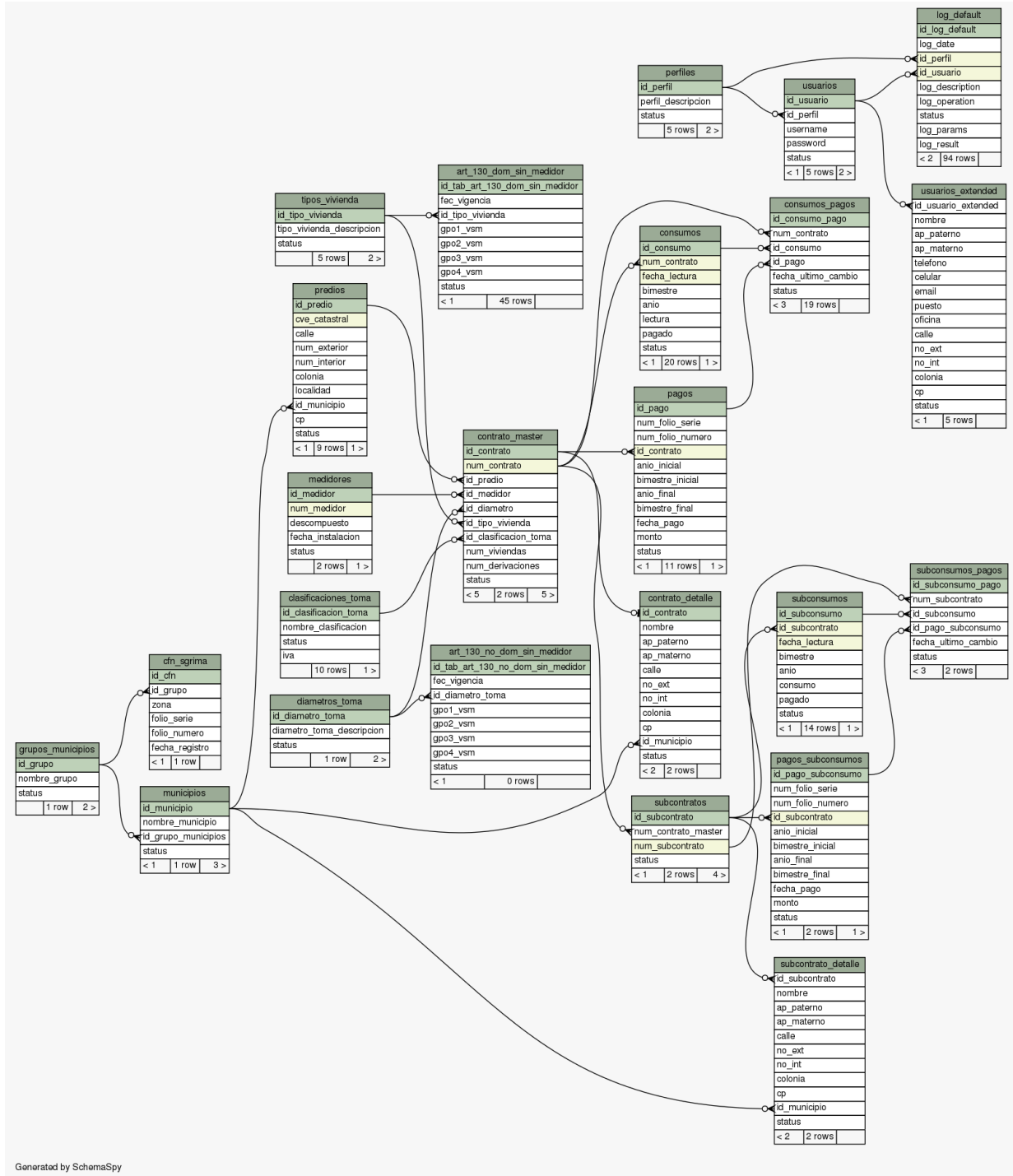


Figura 18: Estructura de la base de datos de la aplicación

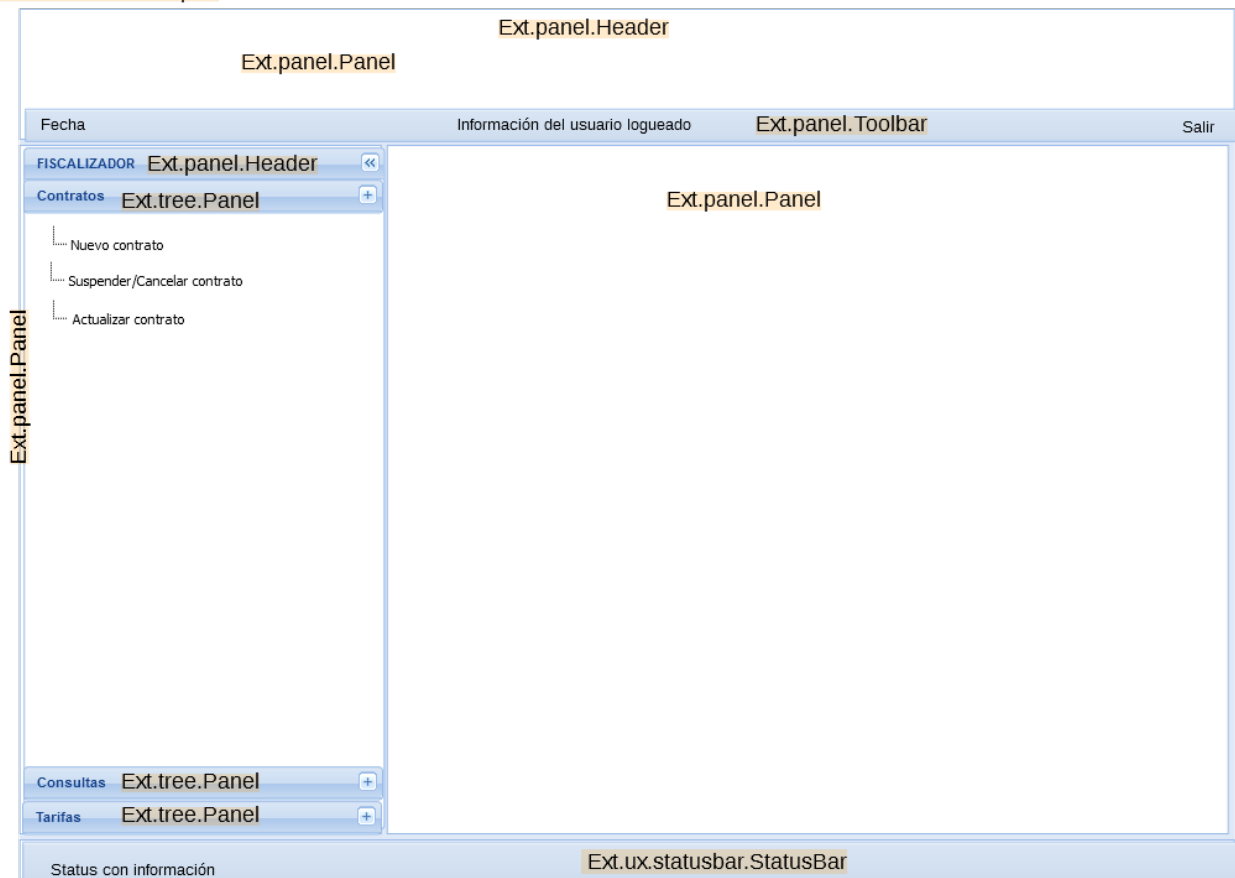
Front-end con ExtJS

Por parte de la vista se ocupo ExtJS 4 (version 4.1.3) para el desarrollo.

ExtJS maneja componentes que interactuan con el back-end por medio de peticion ajax para pedir información. Los componentes se van acomodando dentro de otros componentes que van tomando control de su funcionamiento.

La figura 19 se muestra el diseño de pantalla con los principales componentes. Este diseño de pantalla es la que comparten todos los perfiles del sistema y que conforma su principal area de trabajo.

Ext.Container.Viewport



Para todos los perfiles se sigue la siguiente estructura de archivos:

- | | |
|-------------------------------|--|
| principal_[nombre_perfil].jsp | Genera el area de trabajo como en la imagen de arriba e importa las funciones y modelos que necesitan. |
| functions_[nombre_perfil].js | Funciones que hacen peticiones a back-end |
| models_[nombre_perfil].js | Modelos para reflejar datos regresados por las peticiones a back-end |

Implementacion

Tecnologia utilizada

Software. Para la realización del presente proyecto se decidió utilizar solamente software libre y tratando de usar las versiones más recientes.

Debido a la naturaleza del proyecto y que tiene las aplicaciones web, algunas tecnologías de software se utilizaron para generar la vista del sistema, otras para implementar la lógica de negocio y funcionalidad del lado del servidor y otras como intercambio de datos entre capas de la aplicación.

Tecnologías para generar la vista (UI cliente)

- JSP
Es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.
- ExtJS
Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM.
- Javascript
Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos,³ basado en prototipos, imperativo, débilmente tipado y dinámico.

Tecnologías para lógica de negocio y funcionalidad del lado del servidor

- Java
Java es una plataforma con mucho soporte para el desarrollo de aplicaciones web. El soporte no refiere ni se limita solo a la cantidad de información que existe sobre ella en la red, sino que muchas empresas y organizaciones aportan bibliotecas de código abierto desarrolladas bajo esta plataforma, que facilitan el desarrollo de proyectos.
- Hibernate
Es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.
- Spring

Spring es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java.2

- **Struts**
Es una herramienta de soporte para el desarrollo de aplicaciones bajo el patrón modelo-vista-controlador (MVC) bajo la plataforma J2EE. Struts se desarrollaba como parte del proyecto Jakarta de la Fundación Apache, pero actualmente es un proyecto independiente conocido como Apache Struts.
- **Apache Tomcat**
También conocido como Jakarta Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems.
- **PostgreSQL 8.4**
Es un sistema gestor de base de datos relacional orientado a objetos.

Tecnologías para intercambio de datos

- **JSON**
Acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

Hardware. Para el desarrollo de este proyecto se dispuso de dos computadoras cuyas características se listan a continuación:

Componente	Computadora 1	Computadora 2
Procesador	AMD Turion 64x2	Core i3
Memoria RAM	4 GB	4 GB
Disco Duro	512 GB	256 GB
Sistema Operativo	Ubuntu 11.10	Ubuntu 11.10

Referencias

Hibernate

<http://www.onjava.com/pub/a/onjava/2005/09/21/what-is-hibernate.html>

Spring

<http://www.springsource.org/spring-framework>

Struts

<http://www.onjava.com/pub/a/onjava/2005/11/02/what-is-struts.html>

JSON

<http://en.wikipedia.org/wiki/JSON>

Apache Tomcat

http://www.omii.ac.uk/docs/2.3.1/reference/tomcat/what_is_tomcat_.htm

PostgreSQL

<http://en.wikipedia.org/wiki/PostgreSQL>

ExtJS

http://en.wikipedia.org/wiki/Ext_JS

JavaScript

<http://javascript.about.com/od/reference/p/javascript.htm>

JSP

http://en.wikipedia.org/wiki/JavaServer_Pages

Anexo 2. Guía de Instalación

(En sistemas Ubuntu GNU/Linux)

Convenciones seguidas en el presente documento

[software_dir] = Se refiere a la carpeta ‘programas/’ dentro de este entregable.

[codigo_dir] = Se refiere a la carpeta llamada ‘condigoFuente/Sgrima’ dentro de este disco.

[war_dir] = Se refiere a la carpeta llamada ‘war/’ dentro de este disco.

[bd_dir] = Se refiere a la carpeta llamada ‘baseDeDatos’ dentro de este disco.

[java_dir] = Se refiere al path absoluto del directorio generado al extraer el contenido del archivo ‘jdk-6u26-linux-i586.bin’ en el paso X inciso Y.

[tomcat_dir] = Se refiere al path absoluto del directorio generado al extraer el contenido del archivo ‘’ en el paso X inciso Y.

[sql_file] = Se refiere al archivo scriptInicializacionSgrima12Marzo2013.sql que se encuentra en el directorio ‘baseDeDatos’ dentro de este disco.

Instalacion de software necesario para el funcionamiento de Sgrima

Instalacion de Java

- a. Descargar Java JDK 6
 - i. Entramos a la liga
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
 - ii. Hacemos click en descargar Java Plataform (JDK)
 - iii. Esto lleva a una pagina con diferentes versiones de instalacion del JDK de Java.
 - iv. Hacemos click en la version que mas nos convenga dependiendo de nuestro sistema operativo y arquitectura, pero asegurando de que sea 'Self Extracting Installer', para poder personalizar el directorio de instalacion.
 - v. Una vez realizado lo anterior, se tendra un archivo con un nombre parecido al siguiente: 'jdk6u43linuxi586.bin'.
Nota: Una alternativa es utilizar el archivo que se encuentra en [software_dir].
- b. Extraer en carpeta destino.
 - i. Abrir una terminal y navegamos a traves de los directorios con ayuda del comando 'cd' hasta el folder en donde se quiere instalar Java.
 - ii. Escribimos el comando 'sh' dejamos un espacio y arrastramos a la terminal el archivo 'jdk6u43linuxi586.bin' para que la ruta absoluta de este sea insertada automaticamente y presionamos la tecla enter.
 - iii. Como resultado, se generara un directorio con el nombre 'jdk.1.6.0_43' (varia segun la version) dentro del directorio seleccionado en el inciso i.
- c. Configurar JAVA_HOME
 - i. Abrimos una terminal e introducimos los siguientes comandos:
cd
gedit .profile [enter]
 - ii. Una vez el archivo navegamos al final de este y agregamos las siguientes 4 lineas:
JAVA_HOME = [JAVA_DIR] [enter]
export JAVA_HOME [enter]
PATH=\$PATH:\$JAVA_HOME/bin [enter]
export PATH [enter]
 - iii. Una vez completado el paso anterior, guardamos y cerramos el archivo.
- d. Comprobar instalacion
 - i. Reiniciamos la sesion en GNU/Linux
 - ii. Abrimos una terminal e introducimos lo siguiente:
\$JAVA_HOME/bin/java -version [enter]
Si la instacion fue correcta debe el resultado debe ser parecido a lo siguiente:
Java version "1.6.0-43"

Java(TM) SE Runtime Environment (build 1.6.0_26-b03)
Java HotSpot(TM) Server VM (build 20.1-b02, mixed mode)

Instalacion de Tomcat 6

- e. Descargar Tomcat 6
 - i. Entramos a la liga: <http://apache.webxcreen.org/tomcat/tomcat-6/v6.0.36/bin/apache-tomcat-6.0.36.zip>
 - ii. Hacemos click en el archivo '[apache-tomcat-6.0.36.zip](#)' para descargarlo
Nota: Como alternativa, podemis utilizar el archivo que se encuentra en [SOFTWARE_DIR]
- f. Extraer el archivo en la carpeta destino
 - i. Descomprimos el achivo en el directorio donde queremos instalarlo.
 - ii. Posteriormente, se generara un nuevo directorio con nombre '[apache-tomcat-6.0.36.zip](#)' dentro del directorio que escogimos para su instalacion.
- g. Configurar TOMCAT_HOME
 - i. Abrimos una terminal e introducimos los siguientes comandos:
cd [enter]
gedit .profile [enter]
 - ii. Una vez abierto el archivo navegamos al final de este y agregamos las siguientes 4 lineas:
TOMCAT_HOME = [TOMCAT_DIR] [enter]
export TOMCAT_HOME [enter]
 - iii. Una vez completado el paso anterior, guardamos y cerramos el archivo.
- h. Comprobar instalacion de Tomcat
 - i. Reiniciar nuestra sesion en GNU/Linux
 - ii. Abrimos una terminal y tecleamos lo siguiente:
\$TOMCAT_HOME/bin/./version.sh

Si el resultado es parecido al siguiente, la instalacion fue correcta (se omite parte de la salida al comando anterior ya que se muestra informacion detallada sobre la plataforma en la que fue instalada):

```
sing CATALINA_BASE: [TOMCAT_DIR]
Using CATALINA_HOME: [[TOMCAT_DIR]
...
Server version: Apache Tomcat/6.0.36
JVM Version: 1.6.0_36-b05
...
JVM Vendor: Sun Microsystems Inc.
```


Instalacion de postgresQL

- i. Instalar PostgreSQL al menos la versión 8.4 y según tu sistema operativo
La instalación de PostgreSQL no es trivial y cambia de distribución en distribución, toma en cuenta que los pasos para instalar PostgreSQL pueden ser diferentes para tu distribución.
 - i. Para instalarlo, abrimos una terminal y tecleamos lo siguiente:
sudo apt-get install postgresql postgresql-client postgresql-contrib libpq-dev pgadmin3 [enter]
 - ii. Ahora cambiaremos la contraseña del usuario administrador tecleando lo siguiente (reemplaza la palabra password por la contraseña que desees usar) :
sudo su postgres -c psql [enter]
ALTER USER postgres WITH PASSWORD 'password'; [enter]
\q; [enter]
Esto altera la contraseña dentro de la base de datos.
 - iii. Ahora se necesita realizar lo mismo para el usuario de GNU/Linux *postgres* tecleamos lo siguiente:
sudo passwd -d postgres [enter]
sudo su postgres -c passwd [enter]
Aparecera el prompt solicitando la contraseña y su confirmacion.
 - j. Verificar conexión a postgres
 - i. Para comprobar que PostgreSQL fue instalado de forma correcta, basta introducir lo siguiente en una terminal.
psql --version [enter]
Si el resultado es parecido al siguiente entonces la instalación fue correcta.
psql (PostgreSQL) 8.4.8
9. Instalación de la base de datos Sgrima para la carga Inicial de la aplicación.
- a. Para poder cargar la base de datos primero es necesario loguearse en el sistema como el usuario postgres.
 - i. Abrimos una terminal y tecleamos lo siguiente:
su - postgres [enter]
Nota: Introducir la contraseña que se fijó en el paso 3 inciso III
 - b. Cargar base de datos
 - i. Para cargar la base de datos inicial de Sgrima, tecleamos lo siguiente:
psql sgrima < [SQL_FILE] [enter]
Realizamos el paso 4 inciso
 - c. Verificar que la carga fue correcta
 - i. Para poder comprobar que la base de datos está correctamente instalada, abrimos una terminal e introducimos lo siguiente:

Realizar el paso 4 inciso I, en caso de no estar logueados en una terminal como el usuario postgres.

```
psql -s sgrima [enter]
```

La

- ii. Una vez logueados en una terminal como postgres, introducimos lo siguiente:

```
psql -s sgrima [enter]
```

La terminal devolvera un prompt parecido al siguiente:

```
link#
```

- d. Ejecutar query en tabla de sgrima para comprobar correcta instalacion.
 - i. En el prompt que devuelve el sistema (sgrima#) introducimos lo siguiente:
select * from perfiles; [enter]

Si el resultado es parecido al siguiente, nuestra base de datos esta bien instalada.

id_perfil	perfil_descripcion	status
1	Administrador Sistema	1
2	Administrador Fiscal	1
3	Fiscalizador	1
4	Cajero	1
5	Contribuyente	1

(5 rows)

Hasta aqui, se ha concluido la instalacion del software que Sgrima utiliza. En la siguiente seccion modificaremos todo lo relacionado con la configuracion de inicio del sistema.

Anexo 3. Manual del Usuario Final

Login

Todo los usuarios deben ingresar su nombre y contraseña en la pantalla inicial del sistema



The screenshot shows the SGRIMA login interface. At the top left is the SGRIMA logo, which consists of a blue water drop icon followed by the text 'SGRIMA' in green and blue. Below the logo, the date 'domingo, 14 de abril de 2013' is displayed. The main area contains a 'Login al sistema' dialog box with two input fields: 'Nombre de Usuario:' and 'Contraseña:'. Below these fields is an 'Enviar' button. At the bottom left of the window, the word 'Listo' is visible.

Perfil Administrador del sistema

Una vez que se haya validado al usuario, se mostrará su area de trabajo.



The screenshot shows the SGRIMA administrator profile screen. At the top left is the SGRIMA logo. Below it, the date 'domingo, 14 de abril de 2013' is displayed. The user's role 'Administrador' is shown in the top right corner, along with a 'Salir' button. The main area is titled 'Contenido inicial o central ADMINISTRADOR DEL SISTEMA'. On the left side, there is a navigation menu with 'Administración' and 'Usuario' options. At the bottom left, there is a placeholder text: 'Información adicional dependiendo de la opción de menú seleccionada.' At the bottom center, the word 'Listo' is visible.

En la opción de administración de usuarios se puede:

10. agregar nuevos usuarios
11. editar (doble click sobre la fila a modificar) rápidamente su perfil, nombre de usuario o contraseña
12. eliminar al usuario

Usuarios				
Borrar		Actualizar		
Id	Perfil	Usuario	Contraseña	Status
1	1	admin	admin	1
2	2		adminfiscal	1
3	3		fiscal	1
4	4	cajero	cajero	1
5	5	contrib	contrib	1

Perfil Administrador Fiscal

Una vez que se haya validado al usuario, se mostrara su area de trabajo.

The screenshot shows the SGRIMA web application interface. At the top left is the SGRIMA logo, which consists of a blue water drop icon followed by the text 'SGRIMA' in green and blue. Below the logo, the date 'domingo, 14 de abril de 2013' is displayed on the left, and 'Administrador Fiscal' is displayed on the right. A 'Salir' button is located in the top right corner. The main content area is titled 'CONTENIDO INICIAL PARA ADMINISTRADOR FISCAL'. On the left side, there is a 'Catálogos' menu with a tree view containing the following items: 'Clasificaciones Toma', 'Diametros Toma', 'Grupos de Municipios', 'Municipios', and 'Tipo Vivienda'. Below the menu, there are sections for 'Tarifas' and 'Reportes', each with a plus sign icon. At the bottom of the left sidebar, there is a note: 'Información adicional dependiendo de la opción de menú seleccionada.' The main content area is currently empty. At the bottom right of the interface, the word 'Listo' is displayed.

Dentro de la opción de catálogos se pueden tener para cada catálogo las opciones de agregar, modificar o eliminar dentro del listado de elementos.

The screenshot shows a table titled 'Clasificaciones Toma'. At the top of the table, there are three buttons: 'Nuevo', 'Borrar', and 'Actualizar'. The table has four columns: 'Id', 'Nombre de Clasificación', 'Iva', and 'Status'. The data rows are as follows:

Id	Nombre de Clasificación	Iva	Status
1	domestico con medidor	0	1
2	domestico mixto con medidor	0	1
3	domestico sin medidor	0	1
4	domestico mixto sin medidor	0	1
5	no domestico con medidor	1	1
6	no domestico con medidor con derivaciones	1	1
7	no domestico sin medidor	1	1
8	no domestico sin medidor con derivaciones	1	1
9	predio baldio	0	1
10	prueba clasificacion	0	1

Below the table, there are two buttons: 'Guardar' and 'Cancelar'.

La opción de tarifas permite consultar las tarifas vigentes, así como las tarifas de años anteriores.

Perfil Fiscalizador

Una vez que se haya validado al usuario, se mostrara su area de trabajo.













El Fiscalizador podra realizar las siguientes tareas:

Buscar estado de cuenta del cliente (Menú Consultas)

Se muestra un caja de texto donde debe ingresar los datos de la cuenta y dar click en el boton enviar.

Se muestra los datos de la cuenta que se esta consultando así como el listado de adeudos pendientes. Desde esta pantalla, se puede consultar la lista para imprimir y darle una copia al contribuyente o generar el formato de pago para que el contribuyente pueda hacer el pago.

Información del Cliente									
Número Cuenta:	0870100203010000				Clasificación Toma:	domestico con medidor			
Contribuyente:	Juan Pérez Pérez				Dirección:	Esperanza 100 centro 57800			
Consumos pendientes de pago									
									Imprimir adeudos <input type="button" value="Generar PDF"/>
Fecha Lectura	Bimestre	Año	Lectura	Pagado	Consumo	Subtotal	Recargos y Actualizaciones	Total a pagar	
2011-12-31	6	2011	800	NO	60	180.4024	230.5407	410.94	
2011-10-31	5	2011	740	NO	25	96.1255	126.4464	222.57	
2011-08-31	4	2011	715	NO	26.5	101.7813	137.6601	239.44	
2011-06-30	3	2011	688.5	NO	48.5	182.2625	253.1881	435.45	
2011-04-30	2	2011	640	NO	39.5	156.4115	223.0957	379.51	
2011-02-28	1	2011	600.5	NO	40.5	160.7718	235.275	396.05	
2010-12-31	6	2010	560	NO	55	178.4223	267.7062	446.13	
2010-10-31	5	2010	505	NO	75	250.464	385.0162	635.48	
2010-08-31	4	2010	430	NO	115	787.9279	1239.9997	2027.93	
2010-06-30	3	2010	315	NO	65	291.0445	468.9398	759.98	

Consultar historial del cliente (Menú Consultas)

Permite consultar el historial de movimientos del contribuyente.

Nuevo contrato (Menú Contratos)

Por medio de esta opción se deben capturar las altas de nuevos contratos dentro del sistema.

Nuevo Contrato

Datos del Predio

Clave Catastral:

Dirección:

Datos de la Toma

Diametro de la toma: Tipo de Vivienda: Clasificación Toma:

Num. Viviendas: Num. Derivaciones:

Datos del Medidor

Num. de Medidor: Fecha de Instalación:

Datos del Titular

Titular:

Dirección:

La información a capturar esta dividida en cuatro secciones:

- b. Datos del predio
- c. Datos de la toma
- d. Datos del medidor

e. Datos del titular

Al hacer click en el botón de enviar se generara una confirmación del alta del contrato.

Suspender o cancelar un contrato (Menú Contratos)

Cuando sea necesario suspender una cuenta o cancelarla.

Actualizar Contrato (Menú Contratos)

La pantalla se nos muestra de forma similar a la de nuevo contrato, esta operación permite actualizar datos del contrato.

Consultar tarifas vigentes (Menú Tarifas)

Esta opción permite la consulta de las tarifas vigentes de acuerdo a la ley.

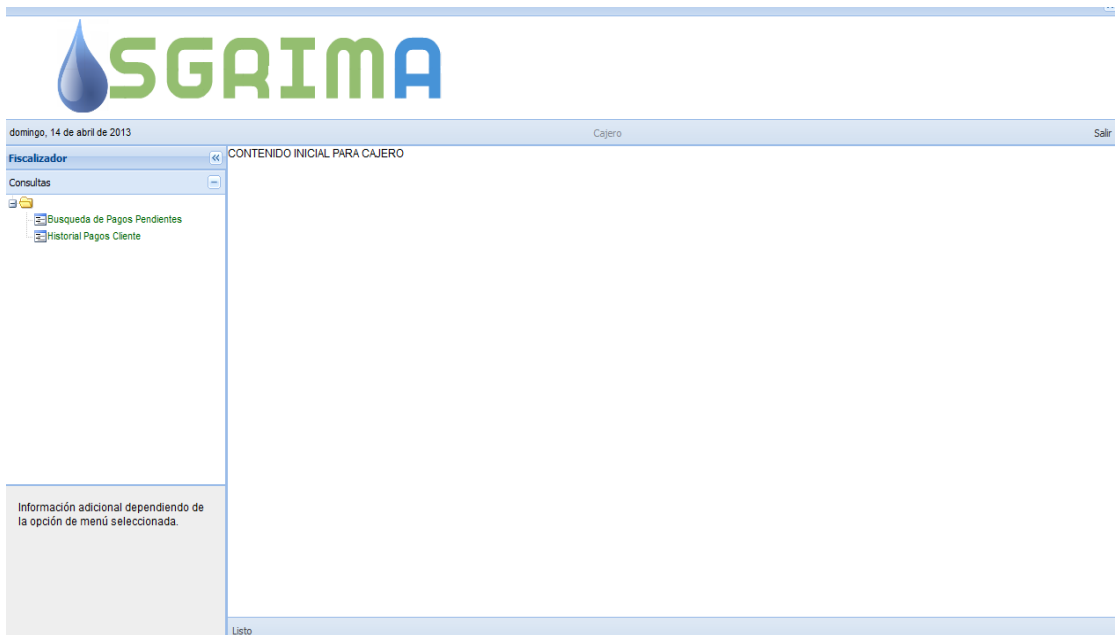
Tarifas										
Domestico con medidor										
fecVigencia	Rango menor	Rango mayor	Cuota mínima Gru	M3 adicional	Cuota mínima Gru	M3 adicional	Cuota mínima Gru	M3 adicional	Cuota mínima Gru	M3 adicional
0	15		1.4727	0	1.2149	0	1.031	0	0.859	0
15.01	30		1.4727	0.0987	1.2149	0.0832	1.031	0.0665	0.859	0.0554
30.01	45		2.9528	0.0988	2.4622	0.0878	2.0288	0.0769	1.6909	0.0659
45.01	60		4.4351	0.1175	3.7798	0.1098	3.1817	0.0094	2.6788	0.0708
60.01	75		6.1975	0.1981	5.4269	0.1752	4.5982	0.1493	3.7411	0.1134
75.01	100		9.1692	0.2324	8.0545	0.1999	6.8376	0.1705	5.4423	0.1266
100.01	125		14.9785	0.3008	13.051	0.2629	11.1001	0.2245	8.6085	0.1617
125.01	150		22.4993	0.3635	19.6243	0.3287	16.7125	0.2809	12.6496	0.1967
150.01	300		31.5868	0.3974	27.8423	0.357	23.7353	0.3056	17.5675	0.2057
300.01	500		91.1989	0.423	81.3979	0.3635	69.5709	0.3116	48.4152	0.2016
500.01	700		175.8068	0.444	154.0974	0.39	131.8909	0.3319	88.7273	0.2128
700.01	1200		264.6139	0.4508	232.0943	0.3963	198.2809	0.3412	131.2789	0.2124
1200.01	999999999		490.0232	0.4508	430.2628	0.3983	368.8856	0.3429	237.4964	0.2135
Domestico sin medidor										
No Domestico con medidor										
No Domestico sin medidor										

Consultar tarifas por año (Menú Tarifas)

Similar a la consulta de tarifas vigenes, pero le permite consultar cuales fueron las tarifas de años anteriores.

Perfil Cajero

Una vez que se haya validado al usuario, se mostrara su area de trabajo.



Buscar de pagos pendientes (Menú Consultas)

Se muestra un caja de texto donde debe ingresar los datos de la cuenta y dar click en el boton enviar.

Se muestra los datos de la cuenta que se esta consultando así como el listado de pagos pendientes.

Desde esta pantalla el cajero puede hacer el cobro del formato de pago y cambiar su estado a pagado.

Historial pagos cliente

Permite revisar el historial de pagos del cliente.

Perfil Contribuyente

Una vez que se haya validado al usuario, se mostrara su area de trabajo.



En la opción Mis contratos (Menú Contratos) se puede consultar cuales son los contactos que estan a su nombre. Se puede consultar mas detalles sobre cada contrato dando click en el ícono de información:

The screenshot shows a table titled 'Contratos' with the following data:

Número de Contrato	Calle	Num. Exterior	Num. Interior	Colonia	Codigo Postal
0870100203010000	Esperanza	100		centro	57800

A 'Detalles' button is visible at the bottom right of the table.

En la opción de Datos de contacto (Menú Mi Información) se puede consultar la información de contacto.