

INTRODUCCIÓN

El municipio de Salamina, como todos los Municipios, necesita la sistematización de sus procesos internos, para control y seguimiento de los empleados (clientes internos), así como para la gestión de sus proyectos y dependencias. Con base en esta necesidad, se hace un acercamiento hacia una intranet de carácter interactiva y de consultas para una pequeña localidad.

No es un desarrollo de Extranet (Portal de Internet) dado que en el sitio de Internet (www.salamina.gov.co) proporcionado por las directrices gubernamentales se atienden solicitudes y servicios a la comunidad, el presente desarrollo se creó con los empleados internos de la Alcaldía de Salamina, y con el fin de utilizar una plataforma de software libre de licencias para prestar el servicio de información interno para la alcaldía a modo de Intranet.

El seguimiento a los requerimientos de los empleados del área de sistemas de la alcaldía, y el uso de las metodologías de análisis, diseño, implementación y documentación permiten que este desarrollo sea una realidad funcional para la Alcaldía.

El presente documento presenta un desarrollo realizado para la Alcaldía de Salamina utilizando tecnologías de información de contenido Web utilizando lenguajes de programación y bases de datos con licenciamiento GNU, además trabaja con notación UML.

1. DESCRIPCIÓN DEL ÁREA PROBLEMÁTICA

Las alcaldías de los municipios de Caldas, al igual que su capital Manizales poseen sus secretarías para solucionar los problemas internos de cada localidad. El problema radica en que no se cuenta con un sistema de información que agilice dichos procesos de solución de problemas y que integre la información de estas secretarías, lo que hace que el trabajo sea más demorado e ineficiente (en cuanto a tiempo se refiere) ya que cada funcionario encargado de desarrollar las tareas afines a dicha secretaría debe mandar oficios o memorandos a las demás dependencias.

Como no se cuenta con un sistema de información, la toma de decisiones de carácter prioritario en estas localidades muchas veces se ve retrasada por causa de la desorganización de la información y por ende no garantiza que la información que allí se encuentra sea veraz, coherente y oportuna.

La desunión de las secretarías hace que el trabajo en equipo no sea un gran fuerte en el sector público ya que no se cuenta con un sistema eficaz de consulta y actualización tanto para el personal administrativo como para la gente del municipio.

Por último es posible identificar otro gran problema, la pérdida de resoluciones y proyectos que quedan en el pasado, puesto que el papel se va desintegrando y no se posee copia de los mismos, esto hace que surjan problemas con terrenos y delimitaciones que no existen, o lo que es peor, que la gente resulta pagando dos o tres veces algo que ya había sido cancelado, todo esto debido a la desorganización que se presenta en las secretarías del sector público (principalmente en los municipios no capitales).

2. JUSTIFICACIÓN

Las Tecnologías de la Información son herramientas que permiten el desarrollo de una nueva economía, la construcción de un Estado más moderno y eficiente, la universalización del acceso a la información, y la adquisición y eficaz utilización del conocimiento, todos estos elementos son fundamentales para el desarrollo de la sociedad moderna.

Es por esto que se desarrollo un Sistema de Información para que la Alcaldía de Salamina, pueda masificar el uso de las Tecnologías de la Información y con ello aumentar la competitividad del sector productivo y socializar el acceso a la información en las secretarías de Planeación, Hacienda y Obras Públicas de dicha Alcaldía.

La idea de crear este sistema de información surge de la necesidad de compartir información entre las diferentes secretarias y de esta forma poder agilizar muchos de los procesos que a diario deben realizar los diferentes funcionarios garantizando así que los datos e información quedaran almacenados correctamente; reduciendo considerablemente la posibilidad que un momento dado la información se pueda perder.

Además, es necesario que las alcaldías sean un solo conjunto, donde todos puedan saber lo que pasa en las secretarías. Se tomaron en cuenta las secretarías anteriormente mencionadas puesto que puede tener una relación más estrecha en cuanto a orden territorial se refiere y por ende poseen información común entre ellas. El sistema pretende dar agilidad de consulta y respaldo de información para la realización de proyectos y manejo de actividades y planeación de las secretarias antes mencionadas, prestando un servicio continuo de consulta de información, disponible para cualquier dependencia que lo necesitara.

La infraestructura de red existente en la Alcaldía de Salamina, así como la existencia de servidores libres como Apache, y lenguajes y bases de datos GNU permiten que el desarrollo de un sistema de información no sea una tarea costosa, como lo podría ser en caso de no existir plataforma abiertas, lo cual justifica plenamente la oportunidad y la necesidad de apuntarle al desarrollo del presente Sistema de información y planear futuros desarrollos en bien de la información del Municipio.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

Realizar el análisis, diseño e implementación de un sistema de información el cual permita interconectar las secretarías compartiendo la información común de las Secretarías de Hacienda, Planeación y de Obras Públicas de las alcaldías municipales del departamento de Caldas, para agilizar los trámites que estas realizan y permitir que la información que éstas tengan sea de fácil acceso a las demás secretarías; teniendo como caso piloto a la Alcaldía de Salamina (Caldas).

3.2 OBJETIVOS ESPECÍFICOS

- ✓ Realizar el análisis, diseño e implementación del módulo de control de usuarios y restricciones (Módulo de seguridad del sistema de información) de las secretarías de Hacienda, Planeación y de Obras Públicas, evaluando la posibilidad de implementar un esquema de control y auditoría
- ✓ Diseñar e implementar un módulo de intercambio de información entre las secretarías de Hacienda, Planeación y de Obras Públicas.
- ✓ Implementar los módulos para consultas y reportes de información de las secretarías de Hacienda, Planeación y de Obras Públicas.
- ✓ Describir las actividades y funciones propias de cada secretaría de forma detallada y descriptiva para el desarrollo del sistema de información.
- ✓ Investigar e Identificar la información necesaria de cada secretaría para el desarrollo de una metodología apropiada.

4. MARCO TEÓRICO

4.1 HERRAMIENTAS UTILIZADAS

4.1.1 UML. El lenguaje unificado modelado (UML) provee un sistema de arquitecturas trabajando con objetos, análisis y diseño, con una buena consistencia del lenguaje para especificar, visualizar, construir y documentar los artefactos de un sistema de software. Esta especificación representa la convergencia de las mejores prácticas en la industria de tecnología de objetos.

Los objetivos que se fijaron al desarrollar el UML fueron los siguientes:

- ✓ Proporcionar a los usuarios un lenguaje de modelado visual de tal forma que sea posible intercambiar información de los modelos.
- ✓ Proporcionar mecanismos ampliados y especializados para ampliar los conceptos básicos.
- ✓ Ser independiente de un lenguaje particular y del proceso de desarrollo.
- ✓ Proporcionar bases formales para la comprensión del lenguaje de Modelado.
- ✓ Integración en una mejor práctica.

El UML es un lenguaje de modelado que incorpora a la comunidad orientada a objetos el consenso de los conceptos de modelado básico y permite desviaciones, las cuales se expresan en términos de mecanismo de extensión. Es un conjunto preciso que consiste en la definición de la semántica y notación del UML, definiendo también como se maneja el Lenguaje de Especificación de Objetos.

Partiendo del hecho de que el ser humano requiere de modelos para manejar sistemas complejos, y en cuanto más complejos se vuelven los sistemas, es necesario tener mejores técnicas del modelado. El contar con una notación Universal para el desarrollo de sistemas de software es de gran beneficio en la construcción de todo tipo de sistemas. Disponer de buenos modelos facilita la comunicación entre equipos de trabajo en un gran proyecto de software utilizando la tecnología orientada a objetos¹

4.1.2 PHP. Con estas siglas se hace referencia a un lenguaje de programación que está muy orientado al desarrollo de aplicaciones Web. Cuando pedimos a nuestro servidor de Web una página PHP, que no es más que un programa que genera HTML, antes de enviar dicha página al cliente se la pasa al intérprete de

¹ BOOCH, Grady con RUMBAUGH, James y JACOBSON, IVAR, El Lenguaje Unificado de Modelado. Editorial Addison Wesley Iberoamericana. Madrid 1999.

PHP; este la interpreta y es el resultado de esta interpretación del programa, contenido en la página PHP, lo que termina llegando al cliente.

PHP se ejecuta en el servidor, por eso permite acceder a los recursos que tenga el servidor como por ejemplo una base de datos (Figura 1). El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. El resultado es normalmente una página HTML pero igualmente podría ser una página WML.

Figura 1. Petición de servicios de PHP



Fuente: ALVAREZ, Miguel Ángel Breve Historia de PHP [on-line]. Textinfo2 el. [Ciudad de México, México]: Desarrollo web.com nov. 2007 [citado 20 ene. 2007]. Disponible en Internet:< <http://www.desarrolloweb.com/articulos/436.php> >

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero para que sus páginas funcionen, el servidor donde están alojadas debe soportar PHP²

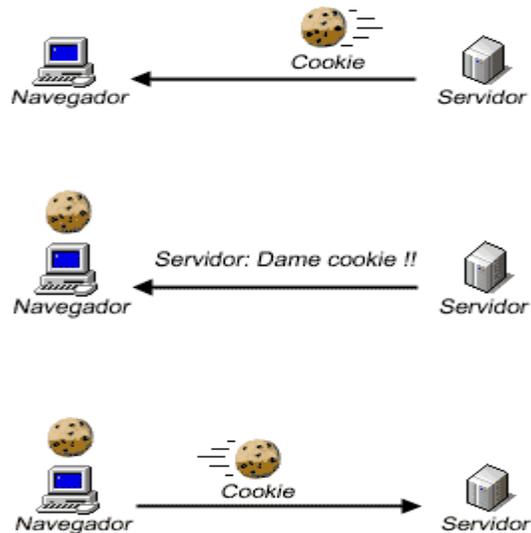
- Cookies. La principal utilidad de las cookies (galletas) es la de solventar el problema de la falta de estado en la navegación a través de las páginas Web.

Con las cookies, pequeñas porciones de información se quedan registradas en el navegador permitiendo identificar a este a través de diferentes páginas de un mismo sitio e incluso durante visitas entre distintos días. Realmente las cookies no son más que cadenas de texto que son enviadas desde el servidor al cliente (navegador) y almacenadas en este, luego el navegador envía estas cookies al servidor permitiendo así la identificación del cliente en el servidor.

² ÁLVAREZ, Miguel Ángel. Breve Historia de PHP [on-line]. Textinfo2 el. [Ciudad de México, México]: Desarrollo web.com nov. 2007 [citado 20 ene. 2007]. Disponible en Internet:< <http://www.desarrolloweb.com/articulos/436.php> >

Su funcionamiento, como puede verse en la Figura 2., empieza cuando la cookie es enviada al navegador desde el servidor y si este la acepta permanece en él. Luego el servidor pide la cookie al navegador; el navegador las envía, permitiendo la identificación del usuario por parte del servidor³.

Figura 2: El navegador solicita la cookie al servidor



Fuente: Carlos Gallús Lahoz y Joaquín Gracia. Manual PHP. [on-line]. Que son las cookies. [Ciudad de México, México]: Desarrollo web.com ene. 2007 [citado 20 ene. 2007]. Disponible en Internet:< <http://www.webestilo.com/php/php13a.phtm> >

4.1.3 SGBD. Los Sistemas de gestión de base de datos son un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos, el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. En los textos que tratan este tema, o temas relacionados, se mencionan los términos SGBD y DBMS, siendo ambos equivalentes, y acrónimos, respectivamente, de Sistema Gestor de Bases de Datos.

³ LAHOZ Gallús Carlos y Gracia Joaquín. Manual PHP. [on-line]. Que son las cookies. [Ciudad de México, México]: Desarrollo web.com ene. 2007 [citado 20 ene. 2007]. Disponible en Internet:< <http://www.webestilo.com/php/php13a.phtm> >

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de información⁴.

Existen distintos objetivos que deben cumplir los SGBD:

- ✓ **Abstracción de la información.** Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario.
- ✓ **Independencia.** La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- ✓ **Redundancia mínima.** Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- ✓ **Consistencia.** En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- ✓ **Seguridad.** La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- ✓ **Integridad.** Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- ✓ **Respaldo y recuperación.** Los SGBD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- ✓ **Control de la concurrencia.** En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de

⁴ CABALLERO Erick. Gestión de bases de datos. [on-line]. Sistema de gestión de bases de datos. [Ciudad de México, México] ene. 2007 [citado 20 ene. 2007]. Disponible en Internet: <<http://www.w3c.org/TR/1999/REC-html401-19991224/loose.dtd>>>

forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

- ✓ **Tiempo de respuesta.** Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Las ventajas de las SGBD son:

- ✓ Facilidad de manejo de grandes volúmenes de información.
- ✓ Gran velocidad y bajo tiempo de respuesta.
- ✓ Independencia del tratamiento de información.
- ✓ Seguridad de la información (acceso a usuarios autorizados), protección de información, de modificaciones, inclusiones y consulta.
- ✓ No hay duplicidad de información, comprobación de información al momento de introducir la misma.
- ✓ Integridad referencial al terminar los registros.

Las desventajas de las SGBD son:

- ✓ El costo de actualización del hardware y software son muy elevados.
 - ✓ Costo (salario) del administrador de la base de datos es costoso.
 - ✓ El mal diseño de esta puede originar problemas a futuro.
 - ✓ Un mal adiestramiento a los usuarios puede originar problemas a futuro.
 - ✓ Si no se encuentra un manual del sistema no se podrán hacer relaciones con facilidad.
 - ✓ Generan campos vacíos en exceso.
- **KEXI:** Es un sistema gestor de bases de datos puede considerarse como un competidor gratuito de programas como Microsoft Access, FileMaker u Oracle en algunos campos. En concreto, permite administrar todo tipo de datos de forma rápida y sencilla y puede utilizarse así mismo para crear esquemas, insertar y procesar datos, y muchas cosas más.
 - **MYSQL.** Su principal objetivo de diseño fue la velocidad, por ello se suprimieron algunas características de los demás SGBDs (Sistema de Gestión de Base de Datos), como las transacciones y las subselects. Consume pocos recursos y se distribuye bajo licencia GPL.

Ventajas:

- ✓ Mayor rendimiento
- ✓ Mejores utilidades de administración
- ✓ Integración perfecta con PHP
- ✓ Sin límites en los tamaños de los registros
- ✓ Mejor control de acceso de usuarios

El lenguaje SQL fue lanzado al mercado por IBM en 1981, después de varios años de desarrollo y en 1983 empezó a vender DB2, un sistema de gestión de bases de datos que se hizo muy popular. Posteriormente, el Instituto de Estándares Nacional Americano (ANSI) adoptó el SQL como estándar para el acceso a bases de datos relacionales, y al año siguiente, se convirtió en estándar ISO. Actualmente, se encuentra en fase de estandarización la versión 3, que será un lenguaje por sí mismo, y no necesitará de otros para actuar.

Alrededor de la década del 90, Michael Windenis comenzó a usar mSQL para conectar tablas usando sus propias rutinas de bajo nivel (ISAM). Sin embargo, después de algunas pruebas se llegó a la conclusión que mSQL no era lo suficientemente rápido ni flexible para sus necesidades. De todo esto surgió en una nueva interfaz SQL (claro que con código mas portable) con algunas apariencias similares en la API de C y en los nombres y funciones de muchos de sus programas. Esto había sido hecho para lograr con relativa facilidad portar aplicaciones y utilidades de MiniSQL a MySQL⁵.

- **POSTGRESQL:** El Sistema Gestor de Bases de Datos Relacionales Orientadas a Objetos conocido como PostgreSQL (y brevemente llamado Postgres95) está derivado del paquete Postgres escrito en Berkeley. Con cerca de una década de desarrollo tras él, PostgreSQL es el gestor de bases de datos de código abierto más avanzado hoy en día, ofreciendo control de concurrencia multi-versión, soportando casi toda la sintaxis SQL (incluyendo subconsultas, transacciones, y tipos y funciones definidas por el usuario), contando también con un amplio conjunto de enlaces con lenguajes de programación (incluyendo C, C++, Java, Perl, etc.).

PostgreSQL soporta grandes cantidades de registros sin bajar su rendimiento y soporta el modelo de cliente/servidor. Y como adición, cuenta con diversas herramientas visuales para su programación.

4.1.4 PHPMYADMIN.

Es una herramienta escrita en PHP con la intención de manejar la administración de MYSQL a través de páginas Web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos.

Es una aplicación tan útil que casi todos los hosting con MySQL disponen de ella. Además, es usado para crear los usuarios MySQL

⁵ ILLERA Marín Alvaro. MYSQL. [on-line]. Manejo de MYSQL. [Ciudad de México, México]: Desarrollo Web. ene. 2007 [citado 20 ene. 2007]. Disponible en Internet:< <http://www.e-ghost.deusto.es/docs/TutorialMySQL.html> >

para así poder utilizar las bases de datos de forma segura. Al ser una aplicación escrita en PHP, necesita de Apache y MySQL para poder funcionar⁶.

4.1.5 APACHE. Según Tora⁷, Apache es un servidor Web gratuito, potente y que ofrece un servicio estable y sencillo de mantener y configurar, siendo, indiscutiblemente, uno de los mayores logros del Software Libre.

Las principales características, se pueden resumir en:

- ✓ Es multiplataforma, aunque idealmente está preparado para funcionar bajo Linux.
- ✓ Muy sencillo de configurar.
- ✓ Es Open-source.
- ✓ Muy útil para proveedores de Servicios de Internet que requieran miles de sitios pequeños con páginas estáticas.
- ✓ Amplías librerías de PHP y Perl a disposición de los programadores.
- ✓ Posee diversos módulos que permiten incorporarle nuevas funcionalidades, estos son muy simples de cargar.
- ✓ Es capaz de utilizar lenguajes como PHP, TCL, Python, etc.

4.1.6 VERTRIGO SERVER. Vertrigo Server ha sido desarrollado para proveer en un paquete altamente profesional y de fácil instalación el Apache (servidor HTTP), PHP (lenguaje de scripting), MySQL (Sistema de administración de bases de datos SQL multithread y multiusuario), SQLite (sistema de administración de bases de datos relacionales ACID), SQLiteManager (aplicación web multilingüe para manejo de bases de datos SQLite), PhpMyAdmin (aplicación escrita en PHP para ocuparse de la administración de MySQL) y Zend Optimizer (que incrementa el rendimiento en tiempo de ejecución en hasta un 40%) para la plataforma Windows. Con un práctico instalador todo en uno, todos los componentes son instalados en un solo directorio y pueden ser utilizados inmediatamente después de completado el proceso de instalación. Un desinstalador le permite remover Vertrigo Server del disco duro. Está diseñado para ser tan pequeño y flexible como sea posible y es en consecuencia muy apto para la distribución por Internet⁸.

⁶ ILLERA MARÍN, Alvaro. PHPMYADMIN. [on-line]. Manejo de PHPMYADMIN. [Ciudad de México, México]: Wikipedia. ene. 2007 [citado 20 ene. 2007]. Disponible en Internet: <<http://mx.php.net/history>>

⁷ TORA, Felipe. Apache. [online]. Que es apache. [Ciudad de México, México]: Wikipedia. ene. 2007 [citado 22 ene. 2007]. Disponible en Internet:<<http://www.mailxmail.com/curso/informatica/configuracionapache/capitulo2.htm>>

⁸ FERNANDEZ Pablo. VERTRIGOSERV. [on-line]. Información sobre vertrioserv. [Ciudad de México, México]: SOURCEFORGE. ene. 2007 [citado 18 ene. 2007]. Disponible en Internet:<<http://vertrigo.sourceforge.net/sheets>>

4.2 SECRETARÍAS Y SUS FUNCIONES

4.2.1 Secretaría de Hacienda y Finanzas. Sus funciones principales consisten en:

- ✓ Coordinar y supervisar el desarrollo de las actividades del sistema de control de gestión de los planes y programas que adelanta la dependencia.
- ✓ Reglamentar la forma como la administración central levantará el inventario de compromisos pendientes de pago a diciembre 31 y la forma como se elaborará la relación de las cuentas por pagar.
- ✓ Preparar el proyecto anual del presupuesto de ingresos y gastos, el plan operativo anual de inversiones en coordinación con la secretaría de planeación y con la colaboración de las demás secretarías.
- ✓ Emitir concepto previo favorable, si es el caso, para modificar, trasladar o incluir nuevas apropiaciones en el proyecto de presupuesto de gastos.
- ✓ Incluir forzosamente en el proyecto de presupuesto la partida necesaria para saldar el déficit fiscal del balance del tesoro de la vigencia anterior, si este resultare.

• **Secretaría de Planeación** Sus funciones principales consisten en:

- ✓ Preparar estudios técnicos necesarios para la elaboración y actualización del plan de desarrollo y el programa de inversiones.
- ✓ Presentar al alcalde y al concejo municipal de planeación el plan y programa de desarrollo económico y social, o de obras públicas.
- ✓ Participar en la elaboración, actualización y modificación del presupuesto municipal en lo referente a inversiones y hacer las evaluaciones periódicas del caso.
- ✓ Elaborar en coordinación con la secretaría de hacienda el presupuesto anual del municipio.
- ✓ Coordinar la elaboración de los planes de acción en cada una de las dependencias del municipio, con base en el plan de desarrollo municipal aprobado, teniendo en cuenta lo establecido en el artículo 3 de la ley 152 de 1994 y las disposiciones constitucionales y legales pertinentes.
- ✓ Asesorar permanentemente a la administración municipal y sus dependencias en el establecimiento de sistemas de programación, evaluación y control de actividades que garanticen el racional aprovechamiento de los recursos disponibles en el artículo 3 de la ley 152 de 1994 y las disposiciones constitucionales y legales permanentes.
- ✓ Organizar, poner en funcionamiento y administrar el banco de programas y proyectos de inversión municipal, acorde con el sistema de seguimiento y evaluación de proyectos de inversión.
- ✓ Realizar seguimiento al plan de ordenamiento territorial de acuerdo con las normas legales y vigentes.
- ✓ Realizar estudios de prefactibilidad, factibilidad económica social en cada uno de los proyectos de inversión que se elaboren en el municipio y dar el respectivo concepto de viabilidad.

- ✓ Hacer cumplir las normas de zonificación, urbanismo y demás instrumentos de planeación contenidos en el plan de ordenamiento.
- ✓ Mantener actualizada toda la información cartográfica en coordinación con el instituto geográfico Agustín Codazzi.

• **Secretaría de Obras Públicas y Valorización.** Sus principales funciones son:

- ✓ Controlar los programas de construcción y mantenimiento de la infraestructura física del municipio, como vías, obras civiles, edificios públicos, parques, zonas verdes, acueductos, alcantarillados.
- ✓ Supervisar los estudios técnicos y diseños para nuevos programas y proyectos del municipio.
- ✓ Mantener una interrelación con el equipo técnico del municipio, como Planeación y Empresa de Servicios Públicos.
- ✓ Evaluar las obras en ejecución y por proyectarse por concepto de valorización municipal, con base en el Plan de Gobierno Municipal.

4.3 LINEAMIENTOS GUBERNAMENTALES

4.3.1 Gobierno en línea. La estrategia de Gobierno en Línea de la Agenda de Conectividad, que establece como uno de sus objetivos el proveer al Estado la conectividad que facilite la gestión en línea de los organismos gubernamentales y apoye su función de servicio al ciudadano, como un complemento al esquema actual, en el que se realizan estos procesos en forma presencial en las oficinas del gobierno y se sustentan con documentos escritos en papel. En desarrollo de esta estrategia de Gobierno, a partir del 24 de agosto de 2000, en periodo de prueba, y desde el 13 de octubre de 2000, en forma definitiva, entrara en funcionamiento el portal de Gobiernoenlinea.gov.co, cuyos principales objetivos son facilitar a los ciudadanos, empresas, funcionarios y otras entidades estatales el acceso a la información de las entidades públicas e iniciar la integración y coordinación de los esfuerzos de las entidades en este propósito⁹.

4.3.2 Antecedentes y directivas: Las estrategias de la Agenda de Conectividad están dirigidas a los siguientes sectores: para el Sector Público, los lineamientos estratégicos están orientados a modernizar la Administración Pública y apoyar la política de seguridad democrática; con el Sector Privado y Comunidad, incrementar la productividad y la competitividad y facilitar a los ciudadanos el acceso a las tecnologías de información y comunicaciones; y por último, con la Academia, apoyar el desarrollo del sector de la educación en el país.

⁹GOBIERNO EN LINEA (Colombia). Proyectos especiales de sistemas [on-line] [Bogotá, Colombia]: Gobierno en Línea. 2007. [citado 10 Ene. 2007]. Disponible en Internet: <<http://www.gobiernoenlinea.gov.co>>

- **La Directiva Presidencial # 02:** La cual “establece como uno de sus objetivos el proveer al Estado la conectividad que facilite la gestión en línea de los organismos gubernamentales y apoye su función de servicio al ciudadano, como un complemento al esquema actual, en el que se realizan estos procesos en forma presencial en las oficinas del gobierno y se sustentan con documentos escritos en papel”. La Directiva plantea la necesidad de dotar a las entidades con estándares que les permitan construir un sitio Web que mantenga la idea de homogeneidad sobre la presencia del Estado en Internet. Esta directiva implica el generar el sustento legal necesario, que oriente a las entidades sobre el contenido que las entidades del Estado pueden publicar en Internet.

El Portal Gobierno en Línea provee a los organismos públicos, ciudadanos y empresas, un medio fácil, seguro y eficiente para ofrecer y acceder a información y servicios del Estado en Internet. Adicionalmente, el Portal proporciona a organismos públicos, ciudadanos y empresas canales directos de comunicación interactiva¹⁰.

- **Agenda de Conectividad:** Su rol puede resumirse en las palabras *estrategia y coordinación*. Para el desarrollo de Gobierno en Línea, su papel fundamental es el de promover la utilización de las Tecnologías de la Información y la Comunicación en la administración pública para mejorar el acceso, costos, calidad y oportunidad de los servicios que presta el Estado al ciudadano¹¹. Sus principales acciones son:

- ✓ Definir los lineamientos estratégicos que permitan a los ciudadanos contar con una oferta de información, trámites, servicios y contratación del Estado colombiano en Internet.
- ✓ Coordinar, asesorar y dar soporte a las entidades que construyan gradualmente un Gobierno en Línea.
- ✓ Realizar seguimiento a la implementación y el grado de avance de las entidades públicas con respecto a iniciativas relacionadas con Gobierno en Línea.
- ✓ Administrar el Conocimiento, para compartirlo entre las entidades que implementan Gobierno en Línea.

¹⁰GOBIERNO EN LINEA (Colombia). Directiva Presidencial 02 [on-line] [Bogota, Colombia]: Gobierno en Línea. 2007. [citado 10 Ene. 2007]. Disponible en Internet: <http://www.gobiernoenlinea.gov.co/directiva_presidencial_02>

¹¹ GOBIERNO EN LINEA (Colombia). Agenda de Conectividad [on-line] [Bogotá, Colombia]: Gobierno en Línea. 2007. [citado 10 Ene. 2007]. Disponible en Internet: <http://www.gobiernoenlinea.gov.co/agenda_de_conectividad>

Objeto de la Agenda de Conectividad

Contratar el servicio de alojamiento, hosting u hospedaje y administración de las páginas Web para las alcaldías mediante el uso de la herramienta “Internet para la rendición de Cuentas del Alcalde Municipal”, la cual fue donada por Transparencia por Colombia al gobierno nacional mediante convenio suscrito entre Vicepresidencia de la República, Transparencia por Colombia y Agenda de Conectividad el día 4 de marzo de 2004.

Alcances del objeto

- ✓ Adaptar el sistema “Internet para la rendición de Cuentas del Alcalde Municipal”, a una Modalidad ASP (*Application Service Provider*), donde el sistema se pueda implementar en una plataforma centralizada y multi-alcaldía.
- ✓ Desarrollar un programa de capacitación y soporte a las alcaldías, para registrar y actualizar la información que se presenta en la página Web.
- ✓ Brindar un servicio de soporte a las alcaldías para en el proceso de actualización de información publicada.
- ✓ Proveer 3 buzones de correo electrónico a cada una de las alcaldías

4.4. SALAMINA, "LA CIUDAD LUZ". Salamina está situada en la zona norte del Departamento de Caldas a una altura de 1.700 metros sobre el nivel del mar. Se ha destacado en la historia nacional como una tierra que ha aportado numerosos valores humanos en todas las manifestaciones de la inteligencia y el saber.

Es Salamina, la “Ciudad Luz de Caldas”, y guarda celosamente los pergaminos de su tradición, sin que por ello deje de manifestarse su incontenible progreso y adelanto. Tierra privilegiada y bella, envidiada por la esperanza. Los Salamineños no solo se sienten orgullosos de mostrar a los demás una ciudad bonita. La tradición intelectual y cultural continúa perenne entre sus gentes. Es notorio el interés por toda manifestación que motive el intelecto.

El sistema económico gira alrededor de la minería, agricultura, ganadería, piscicultura.

AGRICULTURA: La principal actividad agrícola y generadora de mano de obra es el café, asociado al plátano que se produce para el auto consumo, principalmente. La caña panelera le sigue en importancia al café y al plátano; también se cultiva papa, frijol, maíz, tomate y mora.

Las dos principales vías de comunicación son por carretera las cuales comunican el Sector Urbano de Salamina con la Capital de Caldas, una de ellas y la más importante es por el norte del departamento de caldas, comenzando por el

municipio de Neira y pasando por el municipio de Aranzazu. La segunda vía es por el occidente del departamento en la vía que conduce a Medellín, desviándose por el municipio de la pintada (Antioquia), pasando por aguadas y Pácora, esta vía alterna tiene un recorrido más largo.

Límites del municipio:

Norte: Con los municipios de Pácora, Aguadas y Sonsón (ANT.)

Sur: Con los municipios de Marulanda y Pensilvania.

Occidente: Con el municipio de la Merced.

Oriente: Con los municipios de Aranzazu y Neira

Extensión total: 40.354,94 has Km²

Extensión área urbana: 79,69 has Km²

Extensión área rural: 40.245,96 Km²

Altitud de la cabecera municipal (metros sobre el nivel del mar): 1822 m.

Temperatura media: 22° C

Distancia de referencia: a 75 Km. de Manizales

4.5. ANTECEDENTES En el ámbito colombiano, se cuentan con sistemas de información en las alcaldías de las ciudades capitales y unos pocos municipios de avanzada, ya que la mayoría depende en su totalidad del manejo manual de la información o del sistema centralizado en la capital.

En este punto se revisaron algunas experiencias en el estado de sistematización de municipios y entidades públicas en general, con el fin de que nos sirva como punto de referencia para medir y apalancar el norte de los futuros desarrollos.

- ✓ La Alcaldía de Pasto pone a disposición información de interés público y trámites en línea con el fin de optimizar los procesos y darle un mayor sentido de transparencia. Es un Sistema de Información Integral, para consultar información, realizar trámites y registrar todos los proyectos. Licencias de construcción y planes de acción de la comunidad.¹

¹ ALCALDIA DE PASTO (Colombia). Sistema Integral de Información. [on-line]. [Pasto, Colombia]: Alcaldía de Pasto. 2007. [citado 23 ene. 2007]. Disponible en Internet: <<http://www.pasto.gov.co/sii/alcaldia.html>>

- ✓ En el Municipio de Barbosa se ha instalado un Sistema de información en la Alcaldía Municipal para el manejo de sus proyectos y procesos internos, el cual está suministrando un alojamiento (hosting) gratuito del sistema durante dos años, además de habilitar tres cuentas de correo electrónico para uso institucional.²
- ✓ En el Departamento de Boyacá (Colombia) un grupo de estudiantes desarrollaron un sistema integral de información el cual permite a las dependencias y entes descentralizados de la alcaldía mantener la información de forma centralizada. Asesoría en la reestructuración de la red de comunicaciones de la administración municipal.³
- ✓ Por medio de Internet se encuentra disponible la Agenda de Conectividad, la cual ofrece apoyo a las alcaldías para Gobierno en Línea dándoles la posibilidad a cada alcaldía de tener su propia Página Web, correos electrónicos y sistemas administrativos de gestión. En la actualidad hay 624 municipios beneficiados en la primera etapa.
- ✓ La Agenda de Conectividad reiteró a los municipios colombianos la importancia de vincularse al proyecto de Gobierno en Línea, que en su primera etapa llevará a 656 entes territoriales (32 Gobernaciones y 624 municipios), distintas soluciones tecnológicas que incluyen página Web para las alcaldías, correos electrónicos institucionales, sistemas administrativos de gestión y trámites electrónicos, lo que facilitará el acceso de los ciudadanos a los servicios del Estado.⁴
- ✓ En los meses de noviembre y diciembre de 2006 en la ciudad de Bogotá se inicio el proceso de implementación del Sistema Distrital de Información Disciplinaria en las entidades del sector central que hacían falta. De esta manera, las Secretarías de Educación, Hacienda, Tránsito y Obras, y los Departamentos de Catastro, Servicio Civil y Medio Ambiente quedaran así con toda su información integrada en este sistema⁵.

² ALCALDIA DE BARBOSA (Colombia).Modernización de los sistemas de información en la Alcaldía de Barbosa. [on-line]. [Santander, Colombia]: Alcaldía de Barbosa. 2007. [citado 20 ene. 2007]. Disponible en Internet: <<http://www.barbosa-santander.gov.co/>>

³ ALCALDIA DE BOYACA (Colombia). Proyectos especiales de sistemas [on-line] [Boyacá, Colombia]: Alcaldía de Boyacá. 2007. [citado 10 Ene. 2007]. Disponible en Internet: <<http://www.boyaca.gov.co>>

⁴ FEDERACION COLOMBIANA DE MUNICIPIOS (Colombia).Agenda de Conectividad ofrece apoyo a las alcaldías para Gobierno en Línea. [on-line]. [Bogotá, Colombia]: Federación Colombiana de Municipios. 2006. [citado 8 oct. 2006]. Disponible en Internet: <<http://www.fcm.org.co/es/noticia.php?uid=0&todo=0&det=3459&leng=es>>

⁵ ALCALDIA DE BOGOTA (Colombia). Sistema de información disciplinario del distrito capital. [on-line]. [Bogotá, Colombia]: Alcaldía de Bogotá .2006. [citado 15 oct. 2006]. Disponible en Internet: <<http://www.alcaldiabogota.gov.co/SiiD/>>

- ✓ En la Gobernación de Cundinamarca se encuentra en este momento un sistema de información implementado que permite a los ciudadanos conocer todos los proyectos que se están realizando además de poder dar a conocer sus reclamos e inquietudes de algún proceso.⁶
- ✓ En la actualidad la ciudad de Bogotá cuenta con sistemas transversales como son el Sistema Contratación a la Vista mediante el cual todas las entidades distritales publican sus procesos de contratación hacia la ciudadanía; la cual está publicada en una página Web para el fácil acceso de todos los ciudadanos.⁷ Además se encuentra a través de una página Web el Sistema Distrital de Quejas y Soluciones donde se centralizan este tipo de trámites en diferentes Entidades Distritales.⁸ De igual forma en Bogotá se encuentra en desarrollo el modelo de intercambio en el que se encontrará la información espacial georreferenciada para consulta de las diferentes entidades.
- ✓ Santiago de Cali contará con el Sistema de información local: SIL. Se trata de un viaje sencillo por cada rincón de la ciudad en busca de toda la información posible. En el SIL se recopila, sistematiza y divulga toda la información sobre actividades propias de la dinámica administrativa, social, económica, política y cultural del municipio y sus comunas y corregimientos⁹.
- ✓ Para el desarrollo de este proyecto se tuvo en cuenta el manejo que se tiene de la información en la Alcaldía de Manizales¹⁰, donde cada secretaría

⁶ GOBERNACION DE CUNDINAMARCA (Colombia). Cundinamarca es tiempo de crecer. [on-line]. [Cundinamarca, Colombia]: Gobernación de Cundinamarca. 2006. [citado 8 oct. 2006]. Disponible en

Internet:<http://www.cundinamarca.gov.co/cundinamarca/plantillaPpal23.asp?seccion_a=SEC_DE_SCSEC&seccion_b=SEC_BOLEPAGINAPRIN&seccion_c3=SECBLANCO&seccion_c2=SEC_ENLACESMUNI&seccion_c1=SEC_LINKS&seccion_d=SECBLANCO&codigosec=31>

⁷ ALCALDIA DE BOGOTA (Colombia). Contratación a la vista. [on-line]. [Bogotá, Colombia]: Página de Contratación Distrital. 2006. [citado 8 oct. 2006]. Disponible en Internet:<<http://www.Bogotá.gov.co/contratación>>

⁸ ALCALDIA DE BOGOTA (Colombia). Sistema distrital de quejas y soluciones. [on-line]. [Bogotá, Colombia]: Alcaldía mayor de Bogotá Secretaría General. 2006. [citado 8 oct. 2006]. Disponible en Internet: <<http://alcaldia03.bogota.gov.co/sdqs/>>

⁹ ALCALDIA DE SANTIAGO DE CALI (Colombia). Sistema de Información Local - (sil). [on-line]. [Santiago de Cali, Colombia]: Todo de Santiago de Cali para vos. 2006. [citado 6 oct. 2006]. Disponible en Internet:< <http://www.cali.gov.co/index.php?servicio=Noticias&funcion=ver&id=2467>

>
¹⁰ ALCALDÍA DE MANIZALES (Colombia). Manual de funciones de las secretarías. [on-line]. [Manizales, Colombia]: Secretaria de Organización y Sistemas. 2006. [citado 11 nov. 2006]. Disponible en Internet:<http://www.alcaldiamanizales.gov.co/Manizales_Alcaldia/Acerca+de+la+Alcaldia/Organigrama+y+Funciones/>

tiene su información (proyectos, funciones, información general, etc.) condensada en una base de datos general a la cual tienen acceso las diferentes dependencias. De igual manera se encuentra disponible en Internet para que los ciudadanos se enteren con qué funcionarios cuenta la alcaldía y cuáles son sus labores, y con lo que respecta a las secretarías tomadas en cuenta en este proyecto, se tiene un plan informativo de lo que estas secretarías han venido haciendo y como se manejan los dineros públicos.

5. METODOLOGÍA

5.1 TIPO DE ESTUDIO

Es un desarrollo tecnológico, el cual se llevó a cabo utilizando una metodología orientada a objetos (OMT), con una notación UML.

5.2 PROCEDIMIENTO

5.2.1 Fase 1. Diagnóstico. Para la realización de este sistema de información el primer paso que se dio fue ir a la Alcaldía de Salamina e indagar y recopilar información de cómo está estructurada la alcaldía, qué funciones cumple cada secretaría y cómo es procesada y guardada la información. De esta manera y con la ayuda de diferentes funcionarios de la alcaldía nos dimos cuenta de la falta que hacía un Sistema de Información y que útil podría ser no sólo para los funcionarios si no también para la Alcaldía como tal, ya que por medio de este sistema de información se pueden agilizar muchos de los procesos que ellos realizan diariamente como lo son registro de empleados, registro de proyectos, comunicación entre las secretarías entre otros. Dando la posibilidad de que la Alcaldía de Salamina sea pionera de poder contar con un Sistema de Información que no solo integre información sino varias secretarías como lo son la de Hacienda, Planeación y Obras Públicas dando la oportunidad de compartir recursos entre ellas de una forma fácil y sencilla. Y se dice que pionera ya que en ninguna de las alcaldías de los municipios del norte de Caldas cuenta con un Sistema de Información manejado de forma interna por la alcaldía.

5.2.2 Fase 2. Determinación de instrumentos. Para la determinación de instrumentos lo primero que se hizo fue recopilar la información, estructura y funciones de cada secretaría, indagando a cada uno en qué forma manejaban la información correspondiente para cada secretaría en cuanto al manejo de información de cada empleado, proyectos, funciones (Anexo A). De igual manera se le preguntó a cada uno de ellos como contemplarían la posibilidad de diseñar, desarrollar e implementar un Sistema de información en el cual los procesos que ellos realizan diariamente les gustaría fueran implementados y de que utilidad sería para cada uno y como esto contribuiría para que su trabajo diario se realizara de una manera mas óptima y segura.

La recopilación de esta información sirvió para tener un conocimiento claro del funcionamiento de cada secretaría y de que forma era procesada la información y que falencias había en cada secretaría en el momento de procesar y guardar la

información ya que ellos en su mayoría la hacen en forma manual (memorandos, la información es guarda en carpetas y luego almacenada en estantes) y esta con el tiempo se puede perder o dañar y es algo que se puede solucionar con el desarrollo del Sistema de Información

5.2.3 Fase 3. Análisis y Diseño. Para efectos de la propuesta se optó por la metodología OMT (Object Modeling Technique), en sus fases:

- **Análisis.** Para la realización del Análisis de Sistema se tuvo en cuenta los siguientes objetivos:
 - ✓ Se identificaron las necesidades del cliente teniendo en cuenta las necesidades básicas de las diferentes secretarías y determinando de que manera el desarrollo de este software contribuiría a que el trabajo diario de cada funcionario, ya que por medio de este sistema de información se pueden agilizar muchos de los procesos que ellos realizan diariamente como lo son registro de empleados, registro de proyectos, comunicación entre las secretarías entre otros. Dando la posibilidad de que la Alcaldía de Salamina pueda no solo integrar información sino varias secretarías como lo son la de Hacienda, Planeación y Obras Públicas dando la oportunidad de compartir recursos entre ellas de una forma fácil y sencilla. Al realizar el análisis de este sistema y escuchar lo que los funcionarios querían obtener con esta aplicación se tuvo una definición precisa de lo que debe ser el sistema a desarrollar, en términos técnicos y claros que permita una buena interacción con los usuarios y sus funciones dentro del mismo.
 - ✓ Se evaluaron los conceptos y expectativas que tienen los funcionarios de la alcaldía con respecto al sistema y de esta forma establecer su viabilidad técnica, económica como legal ayudando a determinar así que el desarrollo del sistema sea fiable y brinde confiabilidad a cada secretaria.
 - ✓ Se evaluó la interfaz del sistema preguntándole a diferentes funcionarios de la alcaldía para determinar que el entorno e interfaz que se diseñara fuera totalmente amigable y de fácil manejo para cualquiera de ellos.
 - ✓ Se desarrolló el sistema de información de forma que pudiera suplir la necesidad de poder compartir información entre las diferentes secretarías de la alcaldía. Posteriormente se implemento el sistema para poder determinar las posibles falencias que el sistema de información pudiera presentar en el momento de almacenar, consultar o transferir información y así poder llevar a cabo las correcciones para poder garantizar su buen funcionamiento.

- ✓ En el desarrollo de las pruebas se tuvo en cuenta el rendimiento, las restricciones, las interfaces y la fiabilidad. Las restricciones de rendimiento abarcan los requisitos de tiempo de respuesta y procesamiento, identificando los límites del software originados por el hardware externo, por la memoria disponible y por otros sistemas existentes.
- ✓ El backup del sistema se realiza cada vez que se ingresa o modifica los datos en la misma.
- ✓ El sistema de información se diseñó con un sistema de seguridad que se aplica en el momento de ingresar a la página principal el sistema lo primero que hace es pedir un password y login.
- ✓ Al software se realizaron las pruebas de caja de vidrio por medio de la cual se pudo garantizar que cada línea de código se ejecutara de forma correcta, que hubiera buen acoplamiento y buena cohesión, verificar que se ejerciten todas las decisiones lógicas, verificar que se ejerciten todos los ciclos y así poder validar a la información que será procesada por cada línea de código del sistema de información.
- ✓ De igual manera se realizaron las pruebas de caja negra las cuales permiten detectar los errores de interfaz, errores en estructuras de datos o acceso a bases de datos externas, errores de rendimiento, errores al ingresar o salir del sistema.
- ✓ Para verificar la seguridad del sistema de información se tuvo en cuenta los siguientes puntos:
 - Verificar la identificación y autenticación del usuario.
 - Controlar el acceso de acuerdo al perfil.
 - Verificar que tan seguro es el ingreso al sistema
 - Verificar que la sesión termine después de un tiempo previsto de inactividad del sistema
 - Verificar que el acceso a la base de datos dependa de los servicios activos del sistema operacional
 - Verificar que las tablas, consultas, informes de la base de datos se encuentran protegidos con permisos de lectura y escritura de acuerdo con el perfil del usuario que los acceda.
- **Diseño.** Basada en la fase de análisis se determinó la arquitectura global del sistema aplicando la notación UML, elaborando diagramas de clases y posteriormente el modelo relacional de la base de datos (Anexo B). También se incluyó en esta etapa la elaboración de interfaces o prototipos de formularios de interfaz con el usuario, así mismo se desarrolló el manual de usuario final (Anexo C) y el manual de instalación (Anexo D).

- **Implementación.** En esta fase se generó el código fuente necesario en el lenguaje de programación. Para este caso se propuso el lenguaje PHP. Esta fase se cumplió en forma posterior al desarrollo de las fases de análisis y diseño.
- **Pruebas.** Esta fase fue fundamental para evaluar la funcionalidad del sistema de Información propuesto.

El nuevo sistema de información fue verificado de acuerdo al siguiente procedimiento:

- ✓ Instalar el nuevo sistema de información y realizar pruebas con cada uno de los servicios ofrecidos para los cuales fue diseñado el nuevo sistema de información.
- ✓ Capacitar a los funcionarios sobre cómo se instala el sistema de información, sobre su manejo y funciones tales como
 - Ingreso de los datos de los empleados
 - Creación y manejo de perfiles.
 - Gestión de la información del personal.
 - Ingreso de la información a cada dependencia.
 - Elaboración y manejo de mensajes.
 - Elaboración e ingreso al sistema de los diferentes proyectos
- ✓ Los funcionarios ingresaran al sistema
- ✓ Evaluar a los funcionarios que han sido capacitados para el manejo de dicho sistema, con el fin de verificar qué aspectos se requieren reforzar en su capacitación.
- ✓ Preguntar a los funcionarios que beneficios les representa el sistema, cual de los procesos que ellos realizan diariamente se encuentran implementados en el sistema y de que manera agilizan los procedimientos a realizar dirimente en su sitio de trabajo y de que manera mejoran su entorno laboral.

5.2.4 Fase 4. Herramienta de trabajo. Implementación del Sistema. Se utilizó la notación UML para el análisis y diseño del sistema de información con independencia del lenguaje y herramientas, para ello se generaron algunos gráficos como los de los casos de uso y otros fueron actualizados en las diferentes etapas de desarrollo.

El análisis anterior dio pie a la elección de herramientas para el desarrollo, y dado que se tenían dos opciones en el lenguaje de programación PHP y JAVA, se decidió por PHP por su facilidad y compatibilidad con la base de datos MySQL la cual no genera costos para la alcaldía, además el servidor Apache es un servidor robusto y probado para este tipo de aplicaciones.

Buscando sobre plataformas de desarrollo LAMP (LINUX - Apache - MySQL - PHP, Perl o Python) ó WAMP (Windows – Apache – MySQL - PHP, Perl o Python) se encontró una muy poderosa y recomendada que es la utilizada en este proyecto: Vertrigo, al cual puede ser descargada desde <http://vertrigo.sf.net> siendo un desarrollo GNU de la SourceForge (Ver www.sourceforge.net)

Con estas herramientas se comenzó la implementación de acuerdo al cronograma propuesto, con tres semanas de retardo, debido a problemas de comunicación y agenda con los funcionarios de la alcaldía.

5.2.5 Fase 5. Resultados y respuestas. Se instaló el servidor vertrigo, y se implementó el sistema en un computador ubicado en las instalaciones de la alcaldía, donde se tomaron todas las apreciaciones del funcionario encargado.

Se eligieron en colaboración con el personal de la alcaldía, la plantilla de presentación del Sistema de Información, la cual se personalizó, y se ubicó en una hoja de estilos externa llamada "Alcaldia.css".

5.2.6 Fase 6. Finalización. Se desarrolló el presente documento así como los manuales anexos y se dio capacitación a los funcionarios de la Alcaldía.

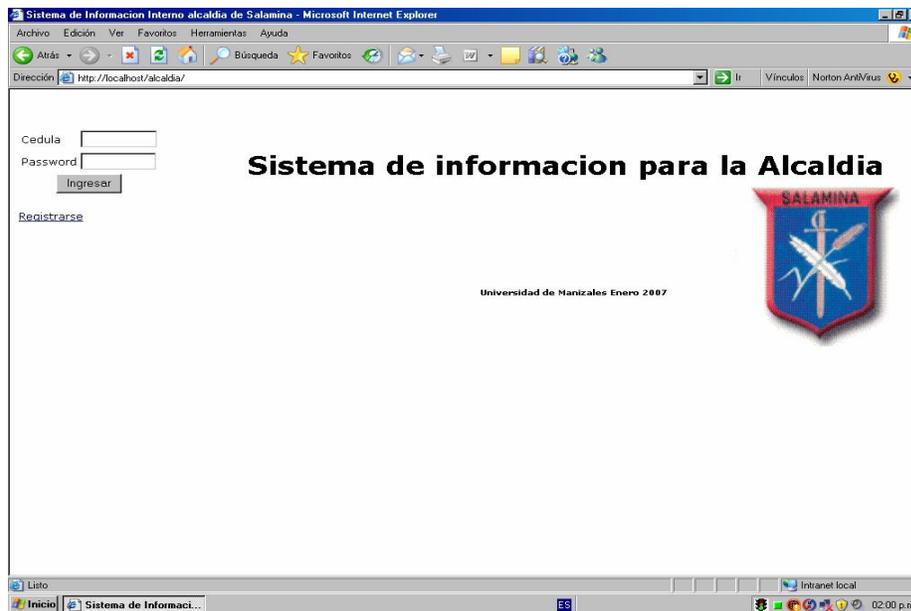
5.2.7 Fase 7 Pruebas. La primera prueba que se le realizó al sistema fue la de usabilidad ya que por medio de esta se pudo concluir qué tan amigable era el sistema tanto su entorno como su manejo, al ser utilizado por un usuario promedio (secretarías y funcionarios de la alcaldía como tal). Posterior a esta se realizaron las pruebas de validación de formularios, la validación de datos, el tiempo de respuesta del sistema en caso de que varios usuarios ingresaran al sistema al mismo tiempo, la seguridad e integridad de los datos en el momento de ser ingresados al sistema.

6. RESULTADOS

El resultado de este trabajo, consiste en un programa que permite manejar integralmente la información relacionada con cada secretaría, sus empleados, sus proyectos, a través de los módulos se crearon diferentes perfiles para garantizar que el sistema fuese mas seguro asegurando que la información no va a ser manipulada sino por las personas adecuadas, de igual manera el programa permite llevar un registro histórico de cada empleado permitiendo saber que acciones realiza.

La aplicación presenta como entrada (Figura 3) la solicitud de la cédula y el password para garantizar que en el momento de ingresar los datos hace que el sistema sea eficaz y seguro.

Figura 3: Ingreso al Sistema de Información de la Alcaldía de Salamina



Para el adecuado ingreso de los datos al sistema se presenta formulario (Figura 4) en el cual cada campo se encuentra validado para garantizar que cada dato que sea ingresado sea correcto.

Figura 4: Formulario para el ingreso de datos.

En el funcionamiento de sistema la gestión de proyectos y archivos anexos ha sido una manera novedosa de emplear las herramientas digitales con que actualmente cuenta la Alcaldía de Salamina. Así mismo el desarrollo de perfiles (Figura 5), para los cuales se agrupan los usuarios, ha permitido control, seguridad y registro de los diferentes actores y las diferentes acciones en el Sistema de Información.

Figura 5: Formulario para editar los perfiles.

Perfil	Comentarios
superusuario	Es el unico usuario que tiene permiso para todas las acciones
Miron	Solo tiene acceso a consultas para mostrar cosas
asistente oficial	Asistente oficial del secretario
asistente oficial	Asistente oficial del secretario
gerente	gerente de una dependencia
aseadora	una aseadora por cada dependencia
gerente	gerente de una dependencia
eliminador	Solo podra eliminar de las diferentes tablas
conshistorico	Una persona que puede consultar historicos sobre la base de datos
editor	Es el que puede editar cualquier tabla
Analista	El analista es una persona del area de sistemas encargada de lo relativo a perfiles
xxxx	cccccccccccccc
Consultor	Es una persona encargada de visualizar ciertas acciones de un tercero
historiador	Solo puede ver historicos
zzzzz	zzzzzzzzzzzz zzzzzzzzzzzzz zzzzzzzzzzzzzzz
zzzzz	zzzzzzzzzzzz zzzzzzzzzzzzz zzzzzzzzzzzzzzz
zzzzz	zzzzzzzzzzzz zzzzzzzzzzzzz zzzzzzzzzzzzzzz
yyy	este es el perfil y
yyy	este es el perfil y
rrrrr	rrrrrrrr
Consultor	Es un secretario encargado de una de las oficinas de la empresa
recepcionista	empleado de cadaa oficina encargado de atencion al cliente
zzzzzzzz	zzzzzzzzzzzz

El seguimiento a las diferentes acciones de un usuario (Figura 6) permite un mayor control de labores realizadas y de igual manera permite avanzar en la construcción de indicadores para la valoración de desempeño.

Figura 6: Formulario para ver las acciones de cada empleado.

Cedula	nombre	Apellido	Email	Tel fijo	Tel Cel	Direccion	Perfil	Dependenci
75070687	Angela Maria	Garcia Osonio	cgarcia@ayv.com	8821279	3124222276	Calle 36 #29a36	1	2
75070688	Usuario limitado	limitadisimo	charlyg@gmail.com	8822323	32432434	calle tal y ascual	2	0
25112431	Angela Maria	Alvarez	angiealoe@yahoo.com	8888888	3113252129	Calle tal y pasucal Salamina	1	2
11111111	Jhonatan	Berrio Ruiz	jo@berrio.com	5555		cuartel policia	1	1
22222222	Jhonatan	berrio	jop@hanatn.com	8822323	3122232323	calle tal y pasucal	1	1
75096583	Jorge humberto	Patiño	jhpchr@yahoo.com	8803915	3137971448	Conjunto Cerrado la Estación	24	2
4454545	pedro	gomez	gg@gmail.com	3343434	322322323	calle tal y pasucal	25	1

El Sistema de Información diseñado responde a las necesidades planteadas por la Alcaldía de Salamina, logrando tener una herramienta que apoye la gestión de un mejor servicio para los usuarios del Sistema de Información.

7. CONCLUSIONES

- El sistema desarrollado en este proyecto en ningún momento desplaza, suprime o modifica el número de puestos de trabajo existentes en las secretarías. Se trata de una herramienta que permite agilizar los procesos de intercambio y almacenamiento de información.
- Se lograron cumplir en su totalidad todos los objetivos establecidos al comienzo del mismo, el Sistema de Información que se entrega es un sistema robusto y confiable gracias al análisis, diseño y estudio detallado del funcionamiento de cada una de las secretarías.
- El desarrollo de este proyecto permitió dar respuesta a una necesidad de desarrollo tecnológico para las diferentes secretarías de las alcaldías. En este caso para la Alcaldía de Salamina.
- Las herramientas de plataformas abiertas son una completa y poderosa opción a la hora de llevar a cabo desarrollos en entidades públicas, específicamente en alcaldías.

8. RECOMENDACIONES

- El hardware necesario para que este sistema pueda ejecutarse sin ningún problema es memoria de 512 MB y disco duro de 40 GB sistema operativo Windows 2000, un servidor de Linux.
- Para futuros desarrollos es importante, crear una estructura de colaboración, al interior de la Alcaldía de Salamina.
- Se deben comenzar a desarrollar los manuales de funciones y la documentación de procesos, de las diferentes funciones en una alcaldía, pues esto facilitará en gran manera el desarrollo de nuevos casos de uso y requerimientos para versiones posteriores del sistema.
- Se debe incluir en el presupuesto Municipal un rubro para la adquisición de la plataforma teleinformática que soporte este y posteriores desarrollos, dado que lo ideal es que cualquier dependencia tenga sus desarrollos y datos, en sus instalaciones en un servidor dedicado.
- El sistema de información se encuentra totalmente analizado, diseñado e implementado para las secretarías de Hacienda, Obras Públicas y Planeación en la Alcaldía de Salamina de igual manera se deja abierto para posteriores cambios ya sea de la alcaldía o de código como tal.

BIBLIOGRAFÍA

ALCALDÍA DE BOGOTÁ (Colombia). Contratación a la vista. [on-line]. [Bogotá, Colombia]: Página de Contratación Distrital. 2006. [citado 8 oct. 2006]. Disponible en: <<http://www.bogota.gov.co/contratacion> >

_____. Sistema distrital de quejas y soluciones. [on-line]. [Bogotá, Colombia]: Alcaldía mayor de Bogotá Secretaria General. 2006. [citado 8 oct. 2006]. Disponible en Internet: <<http://alcaldia03.bogota.gov.co/sdqs/>>

_____. Sistema de información disciplinario del distrito capital. [on-line]. [Bogotá, Colombia]: Alcaldía de Bogotá .2006. [Citado 15 oct. 2006]. Disponible en: <<http://www.alcaldiabogota.gov.co/SiiD/> >

ALCALDÍA DE BARBOSA (Colombia).Modernización de los sistemas de información en la Alcaldía de Barbosa. [on-line]. [Santander, Colombia]: Alcaldía de Barbosa. 2007. [citado 20 ene. 2007]. Disponible en: <<http://www.barbosa-santander.gov.co/>>

ALCALDÍA DE BOYACÁ (Colombia). Proyectos especiales de sistemas [on-line] [Boyacá, Colombia]: Alcaldía de Boyacá. 2007. [citado 10 Ene. 2007]. Disponible en: <<http://www.uniboyaca.edu.co>>

ALCALDÍA DE MANIZALES (Colombia). Manual de funciones de las secretarías. [on-line]. [Manizales, Colombia]: Secretaria de Organización y Sistemas. 2006. [citado 11 nov. 2006]. Disponible en: <http://www.alcaldiamanizales.gov.co/Manizales_Alcaldia/Acerca+de+la+Alcaldia/Organigrama+y+Funciones/>

ALCALDÍA DE PASTO (Colombia).Sistema Integral de Información. [on-line]. [Pasto, Colombia]: Alcaldía de Pasto. 2007. [citado 23 ene. 2007]. Disponible en: <<http://www.pasto.gov.co/sii/alcaldia.html>>

ALCALDÍA DE SANTIAGO DE CALI (Colombia). Sistema de Información Local - (sil). [on-line]. [Santiago de Cali, Colombia]: Todo de Santiago de Cali para vos. 2006. [citado 6 oct. 2006]. Disponible en:<<http://www.cali.gov.co/index.php?servicio=Noticias&funcion=ver&id=2467> >

ÁLVAREZ, Miguel Ángel Manual de programación en web [online]. Textinfo2 el. [Ciudad de México, México]: Desarrollo web.com nov. 1994 [citado 11 nov. 2004]. Disponible en: <http://www.desarrolloweb.com/articulos/711.php?manual=27>>

BOOCH, Grady con RUMBAUGH, James y JACOBSON, IVAR, El Lenguaje Unificado de Modelado. Editorial Adisson Wesley Iberoamericana. Madrid 1999. 432 p.

CABALLERO Erick. Gestión de bases de datos. [On-line]. Sistema de gestión de bases de datos. [Ciudad de México, México] ene. 2007 [citado 20 ene. 2007]. Disponible en:< <http://www.w3c.org/TR/1999/REC-html401-19991224/loose.dtd>">

DATE, C.J. Introducción a los sistemas de bases de datos. México: Addison Wesley Iberoamericana S.A., 1998. Vol 1. 860p

FEDERACIÓN COLOMBIANA DE MUNICIPIOS. Agenda de Conectividad ofrece apoyo a las alcaldías para Gobierno en Línea. [on-line]. [Bogotá, Colombia]: Federación Colombiana de Municipios. 2006. [citado 8 oct. 2006]. Disponible en Internet: < <http://www.fcm.org.co/es/noticia.php?uid=0&todo=0&det=3459&leng=es> >

FERNANDEZ Pablo. VERTRIGOSERV. [on-line]. Información sobre vertrioserv. [Ciudad de México, México]: SOURCEFORGE. ene. 2007 [citado 18 ene. 2007]. Disponible en:< <http://vertrigo.sourceforge.net/sheets>>

GOBERNACIÓN DE CUNDINAMARCA (Colombia). Cundinamarca es tiempo de crecer. [on-line]. [Cundinamarca, Colombia]: Gobernación de Cundinamarca. 2006. [citado 8 oct. 2006]. Disponible en: <http://www.cundinamarca.gov.co/cundinamarca/plantillaPpal23.asp?seccion_a=SEC_DESCSEC&seccion_b=SEC_BOLEPÁGINAPRIN&seccion_c3=SECBLANCO&seccion_c2=SEC_ENLACESMUNI&seccion_c1=SEC_LINKS&seccion_d=SECBLANCO&codigosec=31 >

GOBIERNO EN LÍNEA (Colombia). Directiva Presidencial 02 [on-line] [Bogotá, Colombia]: Gobierno en Línea. 2007. [citado 10 Ene. 2007]. Disponible en: <<http://www.gobiernoenlinea.gov.co>>

ILLERA MARÍN, Álvaro. MYSQL. [on-line]. Manejo de MYSQL. [Ciudad de México, México]: Desarrollo web. ene. 2007 [citado 20 ene. 2007]. Disponible en:< <http://www.e-ghost.deusto.es/docs/TutorialMySQL.html> >

LAHOZ GALLÚS, Carlos y GRACIA, Joaquín. Manual PHP. [on-line]. Que son las cookies. [Ciudad de México, México]: Desarrollo web.com ene. 2007 [citado 20 ene. 2007]. Disponible en:<<http://www.webestilo.com/php/php13a.phtm> >

LARMAN, Craig. UML y patrones: Introducción al análisis y diseño orientado a objetos. México: Prentice Hall, 1999. 536 p

LAUDON, Kenneth C. y LAUDON, Jane P. Administración de sistemas de información. Madrid: McGrawHill. 1995. 885 p.

MEYER, Bertrand. Construcción de software orientado a objetos. Segunda edición. Madrid: Prentice Hall, 1999. 1198 p.

TORO, Felipe. Apache. [on-line]. Que es apache. [Ciudad de México, México]: Wikipedia. ene. 2007 [citado 22 ene. 2007]. Disponible en:<<http://www.mailxmail.com/cursos/informatica/configuracionapache/capitulo2.htm>>

ANEXOS

ANEXO A. RECOPIACIÓN DE INFORMACIÓN

1. INSTRUMENTO DE ENCUESTA

Aplicada a los diferentes funcionarios de las secretarías de Hacienda, Planeación y Obras Públicas de la Alcaldía de Salamina.

1. Considera usted que el Sistema de información es amigable y de fácil manejo para los diferentes usuarios

Si _____ No _____

Porque: _____

2. Considera usted que el ingreso de los datos al Sistema de información se realiza de una sencilla

Si _____ No _____

Porque: _____

3. Considera usted que la selección de cada vínculo dentro del Sistema de información es de fácil manejo para los diferentes usuarios

Si _____ No _____

Porque: _____

4. Considera que el Sistema de Información ayuda agilizar los procesos que se realizan diariamente en cada secretaría

Si _____ No _____

Porque: _____

5. Considera que el Sistema de Información ayuda a mejorar los procesos para almacenar la información de una forma más adecuada y segura

Si _____ No _____

Porque: _____

6. Considera que por medio del Sistema de Información los proyectos tanto en su proceso de licitación, como en el proceso de selección y desarrollo son de fácil entendimiento y manejo para los diferentes usuarios

Si_____ No_____

Porque: _____

7. Considera que el Sistema de Información es un sistema seguro

Si_____ No_____

Porque: _____

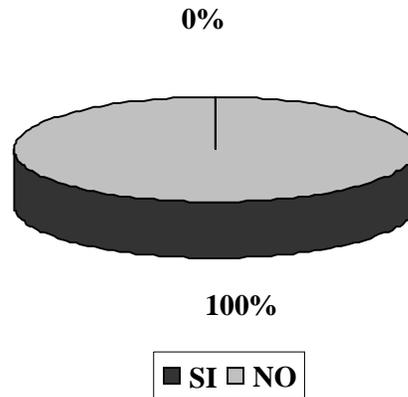
8. Considera que tanto las asesorías brindadas como los manuales proporcionados son claros para el manejo del Sistema de Información.

Si_____ No_____

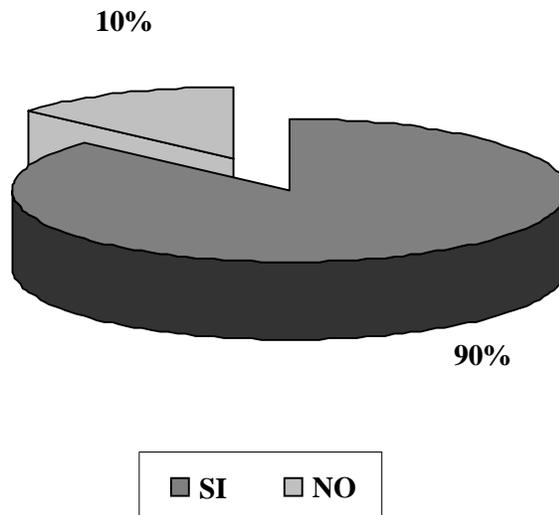
Porque_____

2. RESULTADOS OBTENIDOS

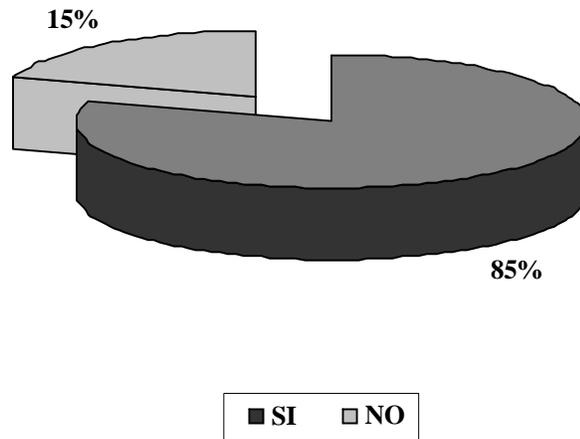
1. Considera usted que el Sistema de información es amigable y de fácil manejo para los diferentes usuarios



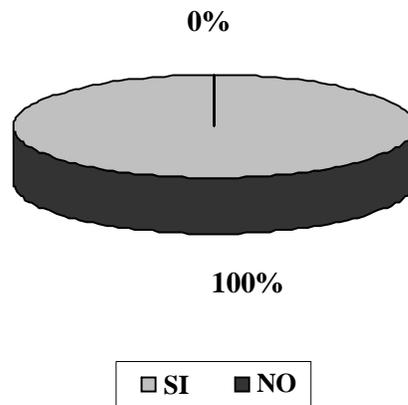
2. Considera usted que el ingreso de los datos al Sistema de información se realiza de una sencilla



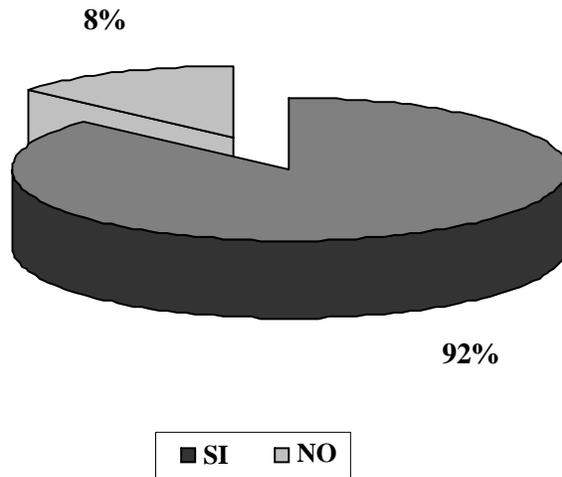
3. Considera usted que la selección de cada vinculo dentro del Sistema de información es de fácil manejo para los diferentes usuarios



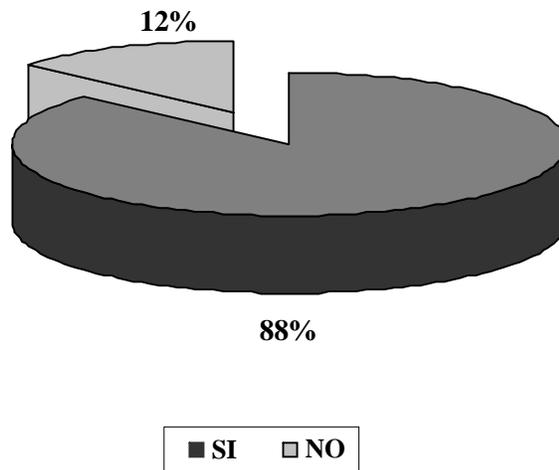
4. Considera que el Sistema de Información ayuda agilizar los procesos que se realizan diariamente en cada secretaria



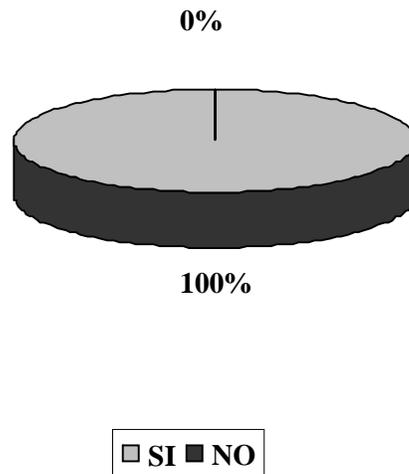
5. Considera que el Sistema de Información ayuda a mejorar los procesos para almacenar la información de una forma más adecuada y segura



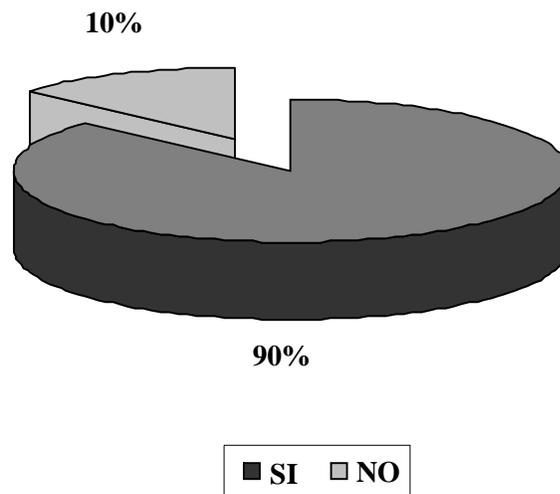
6. Considera que por medio del Sistema de Información los proyectos tanto en su proceso de licitación, como en el proceso de selección y desarrollo son de fácil entendimiento y manejo para los diferentes usuarios



7. Considera que el Sistema de Información es un sistema seguro

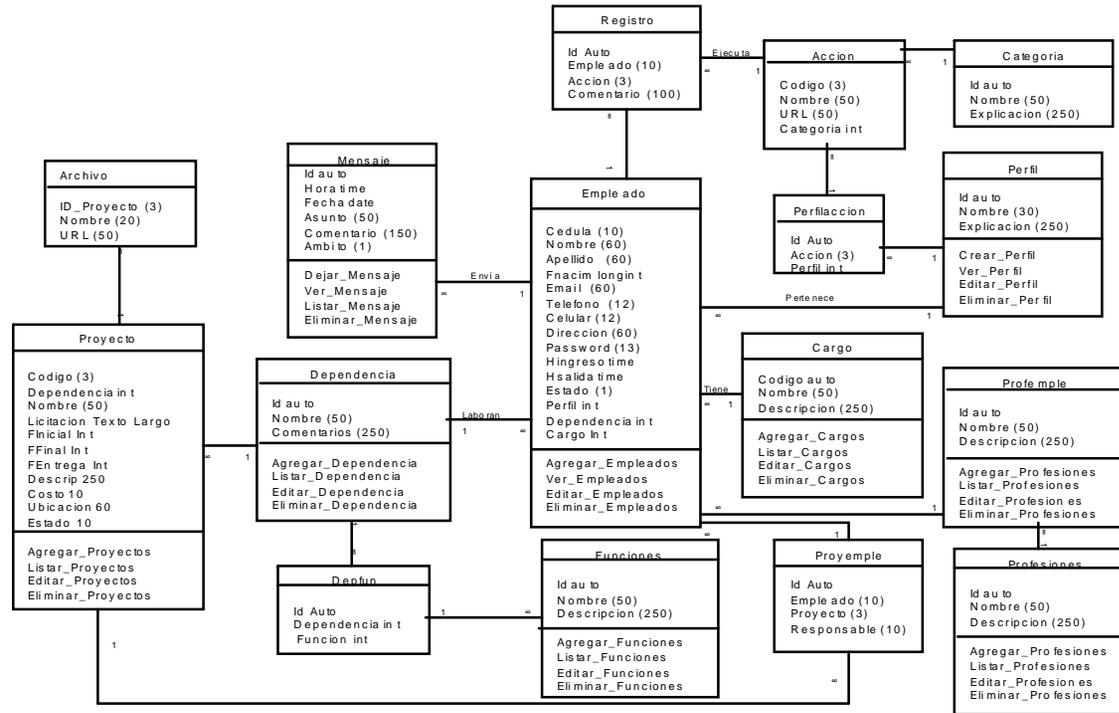


8. Considera que tanto las asesorías brindadas como los manuales proporcionados son claros para el manejo del Sistema de Información.



ANEXO B ANÁLISIS

1. DIAGRAMA DE CLASES



1.1 DIAGRAMAS DE CASOS DE USO Y REQUERIMIENTOS

R1: El sistema debe tener la opción de crear diferentes perfiles de acuerdo a los cuales los diferentes usuarios pueden ejecutar las diferentes acciones.

R2: Se debe poder guardar la información de: empleados, dependencias, funciones, proyectos, mensajes enviados por un usuario, a los compañeros de su dependencia o a todos en general, se debe poder crear, eliminar y cambiar un perfil, y las diferentes acciones permitidas a un usuario de un perfil, y gestionar toda la información anterior, para consulta de los usuarios con los perfiles apropiados.

R3: El sistema debe guardar información de archivos pertinentes a cualquier proyecto para ser descargados por los usuarios que tengan acceso a poder listar proyectos, por lo tanto debe permitir que a la hora de crear un proyecto de alguna dependencia de la alcaldía se puedan colocar dichos archivos al servidor.

R4: Se deben manejar tres tipos fundamentales de usuarios, (súper usuario, responsable y consultas) sin que ello elimine la flexibilidad del sistema, pues éste debe estar en capacidad de permitir la creación de nuevos usuarios con una serie de acciones permitidas dentro del sistema: por ejemplo, 'julio puede hacer las acciones 1 4 y 5' para ello se debe tener una lista de acciones.

R5: Los diferentes usuarios pueden enviar mensajes a sus compañeros de dependencia o a los empleados de manera general de toda la alcaldía, pueden leer los mensajes enviados a un tablón general y solo pueden eliminar los mensajes enviados por ellos mismos.

R6: Cada usuario puede actualizar su información propia, menos la que es relativa al encargado de la base de datos, por ejemplo dependencia a la cual está asignado, perfil etc.

R7: El usuario debe estar en capacidad de actualizar su password el cual se guardará encriptado en la base de datos con el fin de garantizar que ningún usuario, pueda acceder a suplantar a otro, ni aun teniendo acceso a la información de empleados directamente. El campo donde se almacene el password debe estar encriptado y al cambiar un usuario su password debe seguir así.

R8: Al agregar un proyecto se debe asignar los empleados que ejecutarán dicho proyecto con la restricción que los empleados que participan en dicho proyecto, deben ser empleados asignados únicamente a la dependencia a la cual ha sido asignado el proyecto.

R9: Todas y cada una de las acciones que cualquier usuario del sistema ejecute, debe quedar registrada, con fecha hora y una descripción de dicha acción, además cada acción debe llevar un código mediante la cual se debe reconocer.

Esto es con el fin de poder hacer un rastreo o seguimiento de las actividades de cualquier usuario.

También esto servirá para saber qué usuarios han realizado una acción determinada y sobre qué registros de la base de datos.

R10: En cualquier momento el usuario debe ser fácilmente identificable, por el sistema, para presentar la opción sólo de las acciones para la cual este usuario tenga permiso de ejecutar, y también para que el sistema no tenga que preguntar el login y password de esa persona para cualquier acción, o sea el usuario se debe validar ante el sistema y además el sistema debe desconectar al usuario después de cierto tiempo con el fin de proteger las acciones sobre el sistema por parte de terceros, mediante la suplantación de identidad.

1.2 ACTORES

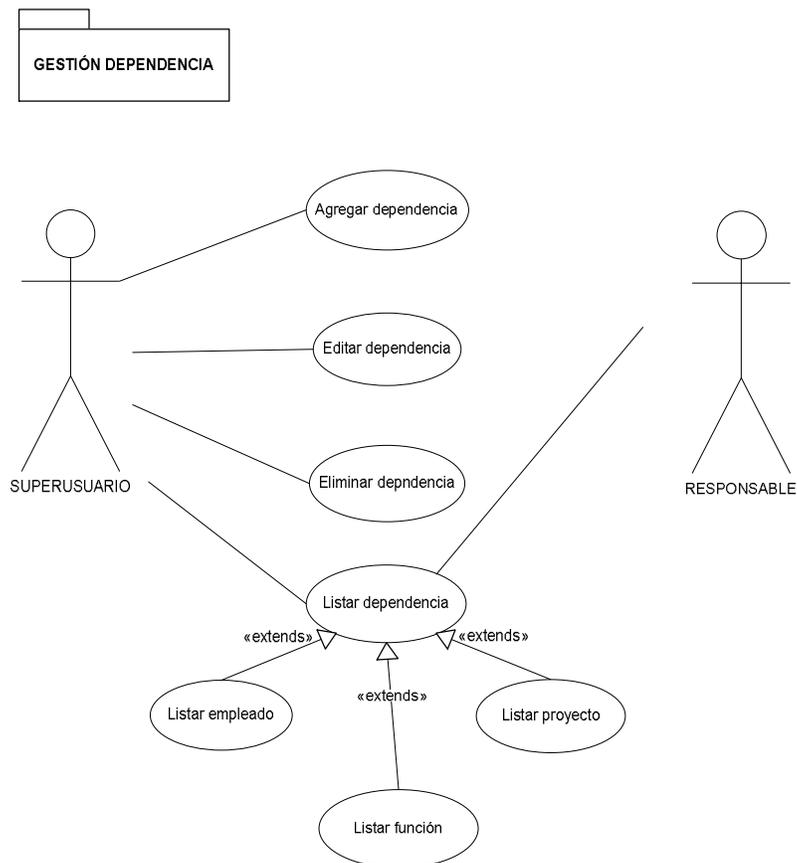
Superusuario: Éste es un actor con los permisos y perfiles suficientes para ejecutar cualquiera de las acciones del sistema, solo debe haber un usuario y es el encargado de crear nuevos perfiles, así como de crear nuevos perfiles de acuerdo a los requerimientos solicitados.

Responsable: Es un actor encargado de la gestión de los proyectos así como de la edición de la información de dependencias y sus funciones, nuevos empleados y gestión de mensajes de la BD.

Consultas: Es un actor encargado de realizar consultas sobre la base de datos, sobre el estado de proyectos, información de empleados, información de dependencias y mensajes, con fines meramente informativos

2. DIAGRAMAS DE CASO DE USO

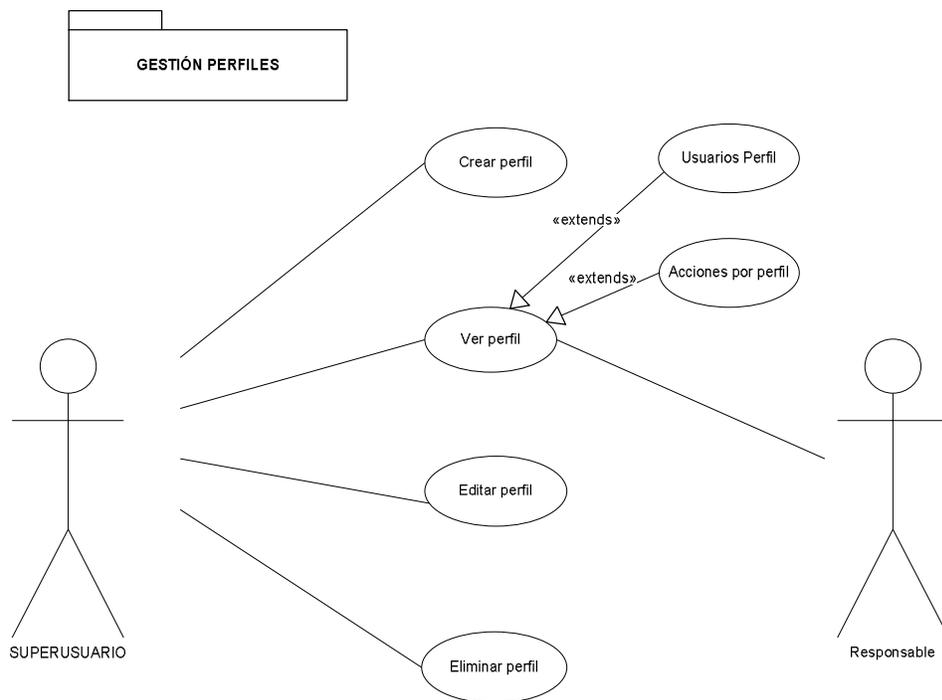
2.1 GESTIÓN DEPENDENCIA El súper usuario tiene el permiso para acceder a toda la información correspondiente a cada dependencia tiene la potestad de ingresar a cada formulario y modificar los datos de cada dependencia cada vez que sea necesario, el sistema lleva un registro histórico de cada movimiento que se haga dentro de la base de datos, así como la hora y fecha en el que fueron realizados. El responsable solo podrá acceder a listar la dependencia de igual manera toda acción que el realice quedara guardada en el registro histórico.



2.2 GESTIÓN PERFILES

El súper usuario tiene el permiso para acceder a los diferentes perfiles creados, tiene la potestad de ingresar a cada formulario y modificar los datos de cada perfil cada vez que sea necesario, el sistema lleva un registro histórico de cada movimiento que se haga dentro de la base de datos, así como la hora y fecha en el que fueron realizados.

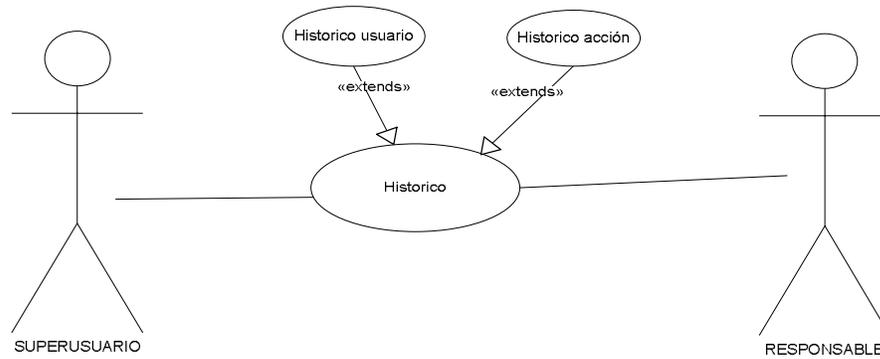
El responsable solo podrá acceder a ver los perfiles creados, a los usuarios por perfil y las acciones por perfil, de igual manera toda acción que el realice quedara guardada en el registro histórico.



2.3 GESTIÓN REGISTRO

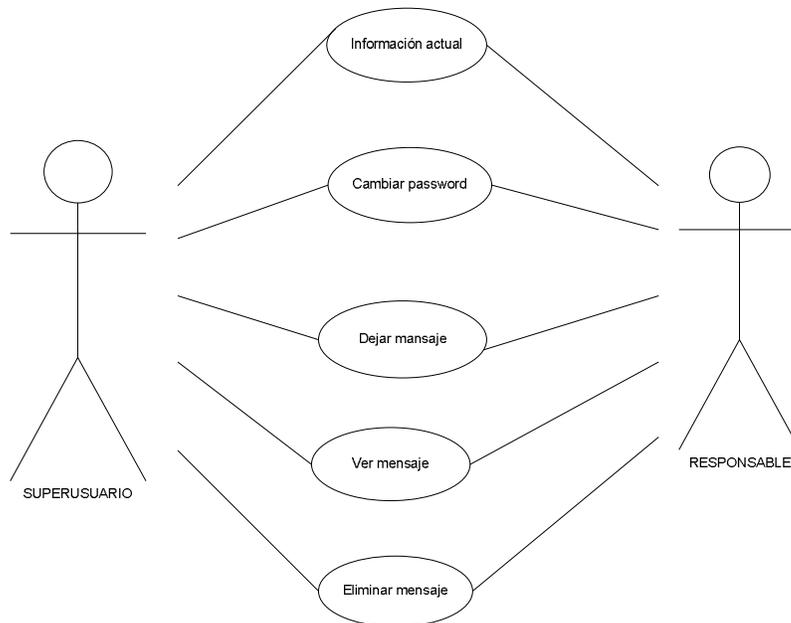
El súper usuario tiene el permiso para acceder al registro histórico y ver que acciones fueron realizadas, quien las realizo y en que fecha y hora fueron realizadas.

El responsable tiene el permiso para acceder al registro histórico y ver que acciones fueron realizadas, quien las realizo y en que fecha y hora fueron realizadas.



2.4 GESTIÓN USUARIO

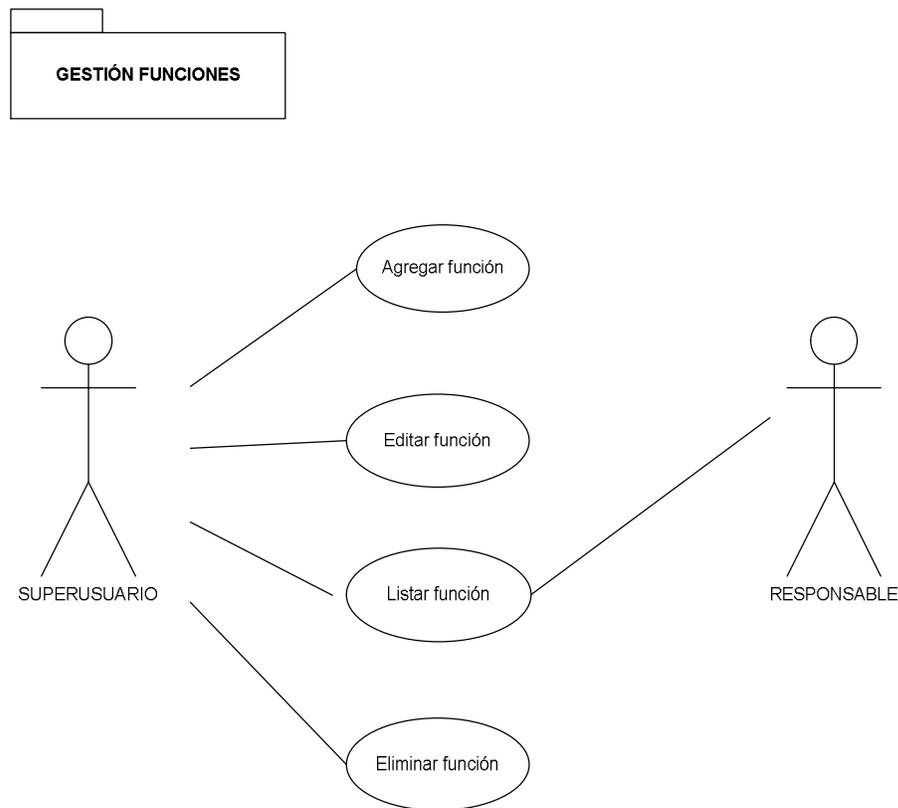
Tanto el súper usuario como el responsable tienen el permiso para acceder a ver sus datos, a cambiar su password, así como, a ingresar al formulario de los mensajes y dejar, ver o eliminar un mensaje. Estas acciones quedan registradas en el registro histórico, quien las realizó y en que hora y fecha fueron realizadas.



2.5 GESTIÓN FUNCIONES

El súper usuario tiene el permiso para acceder a las diferentes funciones creadas para cada empleado, tiene la potestad de ingresar a cada formulario y modificar los datos de cada función cada vez que sea necesario, el sistema lleva un registro histórico de cada movimiento que se haga dentro de la base de datos, así como la hora y fecha en el que fueron realizados.

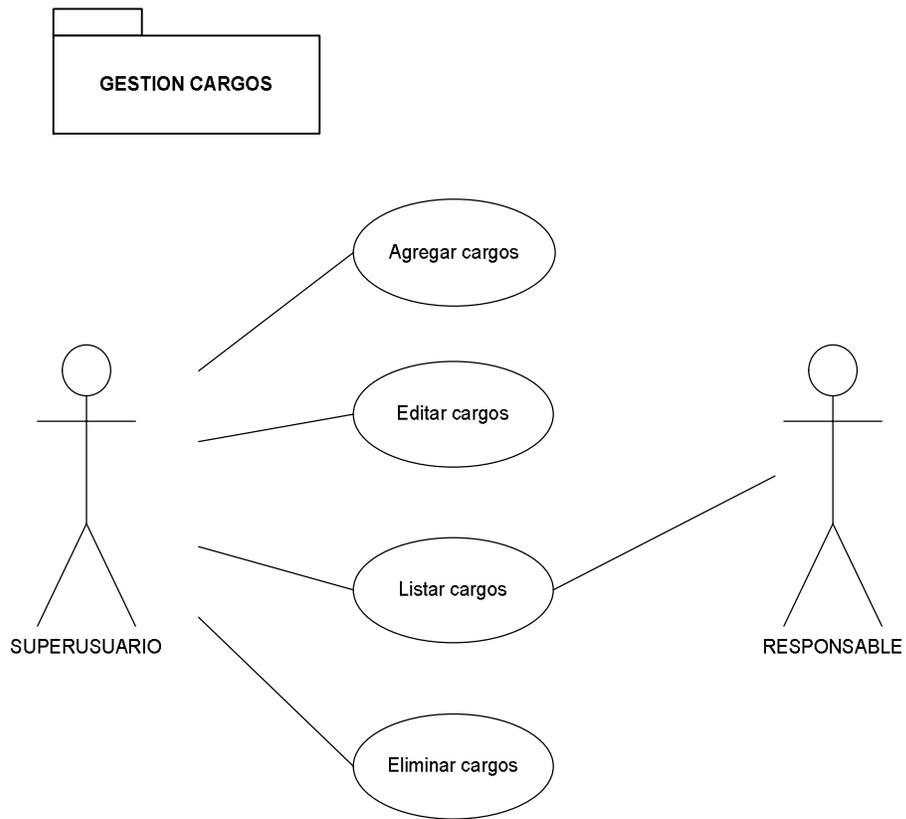
El responsable solo podrá acceder a listar las funciones de cada empleado, de igual manera toda acción que el realice quedara guardada en el registro histórico.



2.6 GESTIÓN CARGOS

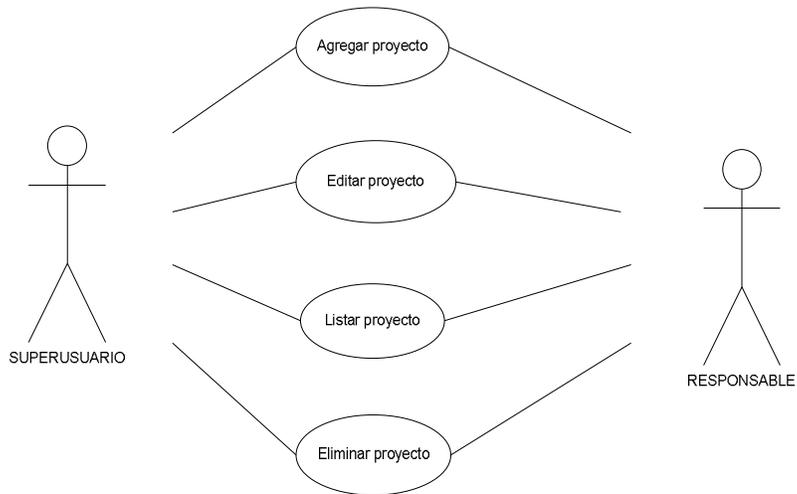
El súper usuario tiene el permiso para acceder a los diferentes cargos que puede tener un empleado, tiene la potestad de ingresar a cada formulario y modificar los datos de cada cargo cada vez que sea necesario, el sistema lleva un registro histórico de cada movimiento que se haga dentro de la base de datos, así como la hora y fecha en el que fueron realizados.

El responsable solo podrá acceder a listar los de cada empleado, de igual manera toda acción que el realice quedara guardada en el registro histórico.



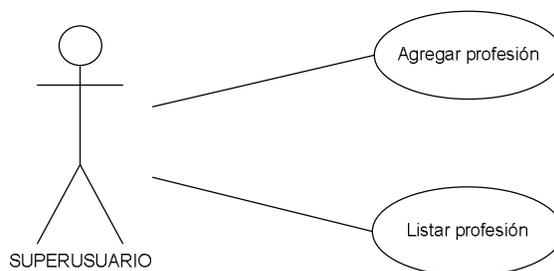
2.7 GESTIÓN PROYECTOS

Tanto el súper usuario como el responsable tienen el permiso para acceder a los diferentes proyectos creados, teniendo la potestad de ingresar a cada formulario y modificar los datos de cada proyecto cada vez que sea necesario, el sistema lleva un registro histórico de cada movimiento que se haga dentro de la base de datos, así como la hora y fecha en el que fueron realizados.



2.8 GESTIÓN PROFESIÓN

El súper usuario tiene el permiso para acceder a los datos de las profesiones de los empleados y agregar ya sea una profesión o listarla según sea necesario, el sistema lleva un registro histórico de cada movimiento que se haga dentro de la base de datos, así como la hora y fecha en el que fueron realizados.



2.8 Explicación Casos de uso

Nombre: **agregareempleados**

Código: **001**

Actores: responsable, súper usuario

Función: Agregar empleados a la base de datos.

Descripción: A los usuarios que les sea permitido agregar a otros empleados se les debe presentar la opción de un formulario de ingreso en el cual se debe guardar un registro histórico de cualquier empleado que use esta opción en la base de datos, así como la hora y fechas de la transacción al guardar un empleado se debe seleccionar a qué dependencia será asignado, este usuario debe ser asignado a uno de los perfiles que existen, debe ser ingresado como Activo, y se le debe escoger uno de los cargos disponibles en la B

Referencias:

De requerimientos: R2, R9, R10

De Casos:

Nombre: **VerEmpleados**

Código: **002**

Actores: responsable, súper usuario, consultas

Función: Ver una lista de empleados de la alcaldía.

Descripción: A los usuarios que les sea permitido ver la lista de empleados se les debe presentar la opción de ver todos los empleados y se muestra la lista, se debe guardar un registro histórico de cualquier empleado que use esta opción en la base de datos

Referencias:

De requerimientos: R2, R9, R10

De Casos:

Nombre: **EditarEmpleados**

Código: **003**

Actores: responsable, súper usuario

Función: Editar la información de un usuario de la BD.

Descripción: A Cualquier usuario debe ser posible cambiar sus datos en cualquier momento por parte de los empleados que tengan permiso para dicha acción, se debe guardar un registro histórico de cualquier empleado que edite a otro usuario.

Referencias:

De requerimientos: R2, R9, R10

De Casos: 010,011

Nombre: **EliminarEmpleados**

Código: **004**

Actores: responsable, súper usuario

Función: Elimina la información de un usuario de la BD.

Descripción: Eliminar un usuario es una opción que permite a quienes tengan el perfil necesario para hacerlo desaparecer el registro de la base de datos, pero se debe tener en cuenta que al eliminar un empleado se eliminan los mensajes que ha enviado y también se le desvincula de los proyectos en los cuales esta inscrito en ese momento también al igual que en todos los casos de uso se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

Referencias:

De requerimientos: R2, R9, R10

De Casos:

Nombre: **CrearPerfil**

Código: **005**

Actores: súper usuario

Función: Crea un nuevo perfil en la BD.

Descripción: Crear un perfil permite crear una agrupación lógica de funciones (acciones) que permite segmentar el tipo de usuario de acuerdo a lo que pueda hacer sobre el sistema: ejm

Jorge es de perfil "consultor" los consultores pueden, hacer X y Y pero no Z, al crear un usuario; a este usuario se le agrega a un perfil determinado y al igual que en todos los casos de uso se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

Referencias:

De requerimientos: R4, R9, R10

De Casos:

Nombre: **VerPerfiles**

Código: **006**

Actores: súper usuario, responsable

Función: Listar los perfiles actualmente creados

Descripción: Listar un perfil permite ver la lista de perfiles creados, pero además es muy importante conocer que usuarios tiene un perfil (subcaso de uso **UsuariosDeUnPerfil**) determinado, también es necesario poder preguntar al sistema que acciones puede un perfil dado ejecutar (Subcaso **AccionesDeUnPerfil**) y al igual que en todos los casos de uso se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

Referencias:

De requerimientos: R1, R4, R9, R10

De Casos:

Nombre: **Editarperfiles**

Código: **007**

Actores: súper usuario

Función: Editar un perfil determinado

Descripción: Al usuario que tenga el permiso para editar perfiles debe mostrar una lista de todos los perfiles existentes y de esa manera, el usuario debe seleccionar un perfil para editar, luego se le debe mostrar un formulario con los campos evitables, el cual tiene la información del perfil que se desea eliminar, se debe tener en cuenta que editar un perfil puede también implicar agregar o eliminar acciones que ese perfil puede realizar, por tal motivo se deben además mostrar las acciones de dicho perfil para que el usuario elija si lo desea dejar eliminar o agregar otra acción a dicho perfil, se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

Referencias:

De requerimientos: R1, R4, R9, R10

De Casos:

Nombre: **EliminarPerfiles**

Código: **008**

Actores: súper usuario

Función: Editar un perfil determinado

Descripción: Al usuario que tenga el permiso para eliminar perfiles debe se le debe permitir una lista de todos los perfiles existentes y de esa manera, el usuario debe seleccionar un perfil para eliminar, además se debe tener en cuenta que se debe eliminar cualquier rastro que hubiese existido en el sistema de dicho perfil, por ejemplo: en su relación con acciones, si hay algún usuario con el perfil que se desea eliminar entonces se debe mostrar el mensaje de que primero se debe cambiar el perfil de los usuarios X y Y dado que ese perfil va a ser eliminado, se debe guardar un registro histórico de cualquier empleado que elimine un perfil.

Referencias:

De requerimientos: R1, R4, R9.R10

De Casos:

Nombre: **Histórico**

Código: **009**

Actores: súper usuario, responsable

Función: Mostrar lo que un usuario determinado ha realizado sobre el sistema, o que usuarios han ejecutado una acción determinada.

Descripción: El histórico es una opción que permite Mostrar lo que un usuario determinado ha realizado sobre el sistema, o que usuarios han ejecutado una acción determinada, por tal motivo lo primero que se debe hacer es mostrar una opción en la cual la persona elija si quiere ver lo que un usuario ha realizado o los usuarios que han realizado una acción determinada.

Si el usuario elije la primera opción se debe desplegar una lista de los usuarios del sistema con el fin de que se pueda elegir uno de ellos y de esa manera poder ver todo el histórico de acciones de ese usuario.

Si el usuario elije la segunda opción (Histórico de acciones) se debe mostrar una lista de las acciones disponibles en el sistema y seleccionar una de ellas para ver quienes de los usuarios que tienen login y password han ejecutado esa acción,

además no sólo se debe decir qué acción ejecutó, si no que también se debe guardar una descripción de lo que hizo así como su fecha y la hora en que se ejecutó. Como Consultar el histórico es una acción, entonces se debe guardar un registro histórico de cualquier empleado que consulte ya sea el histórico de acciones o de un usuario del sistema.

Referencias:

De requerimientos: R9

De Casos:

Nombre: **ActualizarInfo**

Código: **010**

Actores: súper usuario, responsable, consultas

Función: Editar la información personal, como datos de contacto, email etc, de la base de datos.

Descripción: Los usuarios deben tener la opción de editar su información personal, pero sólo alguna parte de la información, pues no podrá por ejemplo: activarse, desactivarse, cambiarse de dependencia, ya que esto lo hace el responsable de este usuario cuando edite información de usuario (Caso de uso 003), él debe actualizar su información personal , tal como teléfono, celular, email, dirección, debe guardar un registro histórico de cuando un usuario haga algún tipo de edición de su información personal.

Referencias:

De requerimientos: R2, R4, R6, R9, R10

De Casos: 003,011

Nombre: **CambiarPassword**

Código: **011**

Actores: súper usuario, responsable, consultas

Función: Cambiar password.

Descripción: Para efectos de seguridad de la información y de la gestión de los diferentes usuarios del sistema, cada usuario debe tener un password, único y que además se debe guardar en la base de datos usando encriptación, ya que este password, no debe ser visible para nadie que tenga acceso a las tablas de la base de datos.

A pesar de esto, cualquier usuario debe cambiar su password como política de la alcaldía y por ello cada usuario debe tener habilitada esta opción, en este caso se debe solicitar la siguiente información:

Password antiguo: con el fin de comprobar la identidad del usuario al momento de hacer un cambio.

Nuevo password: El password que desea, el cual se debe solicitar dos veces para eliminar la probabilidad de cualquier error sintáctico en el momento de crear el password nuevo. Es muy importante validar esta información de manera correcta dado que este es un proceso crítico. Y se debe guardar un registro histórico de cuando un usuario cambia su password.

Referencias:

De requerimientos: R2, R7, R9, R10
De Casos: 010

Nombre: **DejarMensaje**

Código: **012**

Actores: súper usuario, responsable, consultas

Función: Dejar un mensaje, para un usuario o un grupo de usuarios de la alcaldía.

Descripción: Quien tenga los permisos de editar perfiles y que puede enviar mensajes se debe presentar la opción, de enviar un mensaje general que puede tener uno de dos ámbitos:

Para la dependencia, cualquier empleado de una dependencia debe estar en capacidad de enviar mensajes a sus compañeros de dependencia

Para la alcaldía en general: cualquier empleado puede estar en capacidad de enviar un mensaje a todos los empleados de la alcaldía.

Un mensaje se envía por medio de un formulario y el tipo de ámbito o grupo al cual se le ve a enviar dicho mensaje debe ser de escogencia del usuario, se debe guardar un registro histórico de cuando un usuario deja un mensaje de cualquier ámbito, guardando su fecha y hora.

Referencias:

De requerimientos: R2, R5, R9, R10

De Casos: 013

Nombre: **VerMensaje**

Código: **013**

Actores: súper usuario, responsable, consultas

Función: Consultar mensajes enviados.

Descripción: Cualquier usuario: responsable, superusuario, o consultas debe poder leer los mensajes enviados a un tablón de mensajes donde se discrimina, los mensajes enviados a la dependencia del usuario, para lo cual se debe detectar la dependencia de dicho usuario, y luego los mensajes enviados a todos los funcionarios en general de la alcaldía, se debe guardar un registro histórico de cuando un usuario lee los mensajes enviados a cualquier ámbito, guardando su fecha y hora.

Referencias:

De requerimientos: R2, R5, R9

De Casos: 012

Nombre: **EliminarMensaje**

Código: **014**

Actores: súper usuario, responsable

Función: Eliminar mensajes enviados.

Descripción: Los usuarios que pertenezcan al perfil responsable o superusuario, pueden verificar la lista de mensajes enviado por cualquier usuario y de acuerdo a la pertinencia o a algun criterio de significado de dicho mensaje puede eliminarlo,

para lo cual se muestra la lista de mensajes organizados por fecha y se seleccionan los que se van a eliminar, se debe guardar un registro histórico de cuando un usuario elimina los mensajes de cualquier otro usuario, guardando su fecha y hora.

Referencias:

De requerimientos: R2, R5, R9, R10

De Casos: 012,013

Nombre: **AgregarDependencias**

Código: **015**

Actores: súper usuario

Función: Agregar una nueva dependencia al sistema.

Descripción: Los usuarios que pertenezcan al perfil Superusuario, pueden agregar dependencias a la base de datos, se debe presentar un formulario, desde el cual se ingresen los datos de una dependencia y de esta manera los próximos empleados al ser ingresados al sistema encontrarán la nueva dependencia dentro de las opciones a las cuales se asignará, Cuando alguien agregue un usuario se debe guardar un registro histórico guardando su fecha y hora, además quién agregó al sistema y quién ha sido agregado al sistema..

Referencias:

De requerimientos: R2, R9, R10

De Casos: 012,013

Nombre: **EditarDependencias**

Código: **016**

Actores: súper usuario

Función: Editar una dependencia de las actuales en el sistema

Descripción: Al usuario que tenga el permiso para editar dependencias se le debe mostrar una lista de todos las dependencias existentes y de esa manera, el usuario debe seleccionar una para editar, luego se le debe mostrar un formulario con los campos editables de ella, el cual tiene la información del perfil que se desea editar, se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

Referencias:

De requerimientos: R2, R9, R10, R16

De Casos:

Nombre: **ListarDependencias**

Código: **017**

Actores: responsable, súper usuario, consultas

Función: Ver una lista de Dependencias inscritas en el sistema.

Descripción: A los usuarios que les sea permitido ver la lista de dependencias se les debe presentar la opción de ver todas las dependencias en una lista, se debe guardar un registro histórico de cualquier empleado que use esta opción en la base de datos

Referencias:

De requerimientos: R2, R9, R10, R16

De Casos:

Nombre: **EliminarDependencias**

Código: **018**

Actores: súper usuario

Función: Eliminar Dependencias del sistema.

Descripción: Los usuarios que pertenezcan al perfil superusuario, pueden verificar la lista de dependencias, y seleccionar de entre ellas una o varias para eliminar, luego de seleccionadas se eliminan, se debe guardar un registro histórico de cuándo un usuario elimina alguna dependencia, guardando su fecha y hora.

Referencias:

De requerimientos: R2, R5, R9, R10, R15, R16,

De Casos: 012,013

Nombre: **AgregarProyecto**

Código: **019**

Actores: súper usuario

Función: Agregar un proyecto al sistema.

Descripción: A Quienes puedan agregar proyectos al sistema se les debe entregar un formulario en el cual se solicita la información del proyecto y se especifica a qué dependencia estará asignada, así como cuales son los empleados que estarán asignados a este proyecto. También es muy importante que los empleados asignados a dicho proyecto sean sólo de la dependencia a la cual fue asignado el proyecto.

Cuando se crea un proyecto es importante que se pueda colocar información anexa a dicho proyecto y eso se hace por medio del envío de archivos, para lo cual el sistema debe soportar almacenar los archivos, que las personas consideren importante como información anexa, se debe guardar un registro histórico de cuándo un usuario crea un proyecto, guardando su fecha y hora.

Referencias:

De requerimientos: R2, R8, R9, R10

De Casos: 020, 021,022

Nombre: **ListarProyecto**

Código: **020**

Actores: Súper usuario, Responsable, Consultas

Función: Listar los proyectos que están inscritos en el sistema.

Descripción: A cualquier usuario que pueda listar proyectos, se le permitirá ver una lista completa de proyectos y luego el mismo usuario puede seleccionar cualquiera de dichos proyectos para ver más información sobre él en específico.

Se debe guardar un registro histórico de cuándo un usuario lista proyectos guardando su fecha y hora, y el usuario que listó los proyectos.

Referencias:

De requerimientos: R2, R8, R9, R10
De Casos: 013

Nombre: **EditarProyecto**

Código: **021**

Actores: Súper usuario, Responsable

Función: Editar La información pertinente a un proyecto.

Descripción: Al usuario que tenga el permiso para editar proyectos se debe mostrar una lista de todos los proyectos creados en el sistema, luego que él seleccione alguno de los proyectos, se debe mostrar toda la información de ese proyecto, con posibilidad de edición y luego se debe actualizar dicho registro se debe guardar un registro histórico de cualquier empleado que edite algún proyecto, con fecha hora y una descripción de lo que ese usuario hizo.

Referencias:

De requerimientos: R2, R8, R9, R10,
De Casos:

Nombre: **EliminarProyecto**

Código: **022**

Actores: súper usuario, responsable

Función: Eliminar un proyecto del sistema.

Descripción: Los usuarios que pertenezcan al perfil responsable o súper usuario, tienen la posibilidad de eliminar cualquier proyecto, dado que a veces se puede hacer necesario esto, se debe mostrar una lista de los proyectos actuales y luego el usuario selecciona qué proyecto va a eliminar, además luego de seleccionado, se debe eliminar también todo lo que en el sistema quede de dicho proyecto, desvincular los empleados, borrar archivos etc. Por ultimo se debe guardar un registro histórico de cuándo un usuario elimina un proyecto, guardando su fecha y hora, y el usuario que ejecutó dicha acción.

Referencias:

De requerimientos: R2, R8, R9, R10
De Casos: 019, 021,020.

Nombre: **AgregarFunciones**

Código: **023**

Actores: responsable, súper usuario

Función: Agregar Funciones a la base de datos.

Descripción: A los usuarios que les sea permitido agregar funciones se les debe presentar la opción de un formulario de ingreso en el cual Debe tener dos partes

1: Los datos específicos de las funciones.

2: Las dependencias que aplican para tener dicha función

Por lo tanto al guardar una función se deben seleccionar las dependencias para las cuales esa función será activa, se debe guardar un registro histórico de cuándo un usuario agrega una función, guardando su fecha y hora, y el usuario que ejecutó dicha acción.

Referencias:

De requerimientos: R2, R9, R10.

De Casos: 023, 024, 025, 026.

Nombre: **VerFunciones**

Código: **024**

Actores: Responsable, súper usuario, Consultas

Función: Listar las funciones.

Descripción: Listar funciones es una opción en la cual, los empleados que puedan ejecutarla, puedan ver una lista de todas las funciones que tienen las diferentes dependencias pero con la opción de ver una lista de dependencias por cada una de las funciones listadas.

Se debe guardar un registro histórico de cuándo un usuario ve la lista de funciones, guardando su fecha y hora, y el usuario que ejecutó dicha acción

De requerimientos: R2, R9, R10

De Casos: 023, 024, 025, 026

Nombre: **EditarFunciones**

Código: **025**

Actores: Súper usuario, Responsable

Función: Editar La información pertinente a una función.

Descripción: Al usuario que tenga el permiso para editar funciones se debe mostrar una lista de todos las funciones creadas en el sistema, y se debe presentar la posibilidad de que dicho usuario elija una función y pueda cambiar la información de dicha función o cambiar las dependencias a las cuales afecta dicha función, se debe guardar un registro histórico de cualquier empleado que edite algún proyecto, con fecha hora y una descripción de lo que ese usuario hizo.

Referencias:

De requerimientos: R2, R9, R10.

De Casos: 023, 024, 025,026.

Nombre: **EliminarFunciones**

Código: **026**

Actores: Súper usuario, Responsable

Función: Eliminar Funciones del sistema.

Descripción: Para eliminar una función del sistema se deben listar las funciones existentes, y luego se debe seleccionar la que se debe eliminar, de igual manera se debe eliminar cualquier registro que use esa función, se debe desvincular las dependencias que tenían esa función, además, se debe guardar un registro histórico de cuándo un usuario elimina alguna función, guardando su fecha y hora.

Referencias: R2, R9, R10

De requerimientos:

De Casos: 023, 024,025

Nombre: **AgregarCargos**

Código: **027**

Actores: responsable, súper usuario

Función: Agregar Cargos a la base de datos.

Descripción: A los usuarios que les sea permitido agregar Cargos se les debe presentar la opción de un formulario de ingreso en el cual se ingresan los datos para un nuevo cargo, se debe guardar un registro histórico de cuándo un usuario agrega un cargo guardando su fecha y hora, y el usuario que ejecutó dicha acción.

Referencias:

De requerimientos: R2, R9, R10.

De Casos: 028, 029, 030.

Nombre: **VerCargos**

Código: **028**

Actores: Responsable, súper usuario, Consultas

Función: Listar los Cargos.

Descripción: Listar cargos es una consulta que se ejecuta sobre la base de datos y arroja una lista de los diferentes cargos que en la base de datos existen, y que están disponibles para ser usados por los diferentes empleados de la base de datos.

Se debe guardar un registro histórico de cuándo un usuario ve la lista de cargos, guardando la fecha y hora, y el usuario que ejecutó dicha acción

De requerimientos: R2, R9, R10

De Casos: 027, 029, 030

Nombre: **EditarCargos**

Código: **029**

Actores: Súper usuario, Responsable

Función: Editar La información pertinente a una Cargo.

Descripción: Editar cargos, es una opción que permite al usuario que tenga el perfil requerido para esta operación, listar completamente los cargos disponibles en nuestra base de datos, y de esa manera se le permite al usuario elegir entre uno de los diferentes cargos existentes, para que ese cargo salga en campos editables, y por ultimo el usuario pueda actualizar la información de dicho registro se debe guardar un registro histórico de cualquier empleado que edite algún Cargo, con fecha hora y una descripción de lo que ese usuario hizo.

Referencias:

De requerimientos: R2, R9, R10.

De Casos: 027, 028,030.

Nombre: **EliminarCargos**

Código: **030**

Actores: Súper usuario, Responsable

Función: Eliminar Funciones del sistema.

Descripción: Para eliminar una función del sistema se debe listar las funciones existentes, y luego se debe seleccionar la que se debe eliminar, de igual manera

se debe eliminar cualquier registro que use esa función, se deben desvincular las dependencias que tenían esa función, además, se debe guardar un registro histórico de cuando un usuario elimina alguna función, guardando su fecha y hora.

Referencias: R2, R9, R10

De requerimientos:

De Casos: 027, 028,029

Nombre: **AgregarProfesiones**

Código: **031**

Actores: responsable, súper usuario

Función: Agregar Profesiones a la base de datos.

Descripción: A los usuarios que les sea permitido agregar profesiones se les debe presentar la opción de un formulario de ingreso en el cual se ingresan los datos para una nueva profesión, se debe guardar un registro histórico de cuándo un usuario agrega una profesión guardando su fecha y hora, y el usuario que ejecutó dicha acción.

Referencias:

De requerimientos: R2, R9, R10.

De Casos: 032

Nombre: **VerProfesiones**

Código: **032**

Actores: Responsable, súper usuario, Consultas

Función: Listar las Profesiones.

Descripción: Listar Profesiones es una consulta que se ejecuta sobre la base de datos y arroja una lista de los diferentes profesiones que existen en la base de datos y que están disponibles para ser usadas en la gestión de la base de datos.

Se debe guardar un registro histórico de cuándo un usuario Ve la lista de cargos, guardando la fecha y hora, y el usuario que ejecutó dicha acción

De requerimientos: R2, R9, R10

De Casos: 031

Nombre: **Salir**

Código: **033**

Actores: Responsable, súper usuario, Consultas

Función: Salir del sistema.

Descripción: Salir del sistema es una opción que permite eliminar cualquier posibilidad de seguir ejecutando acciones en la base de datos con el usuario actual, se debe eliminar cualquier indicio de reconocimiento del usuario y solicitar nuevamente una conexión.

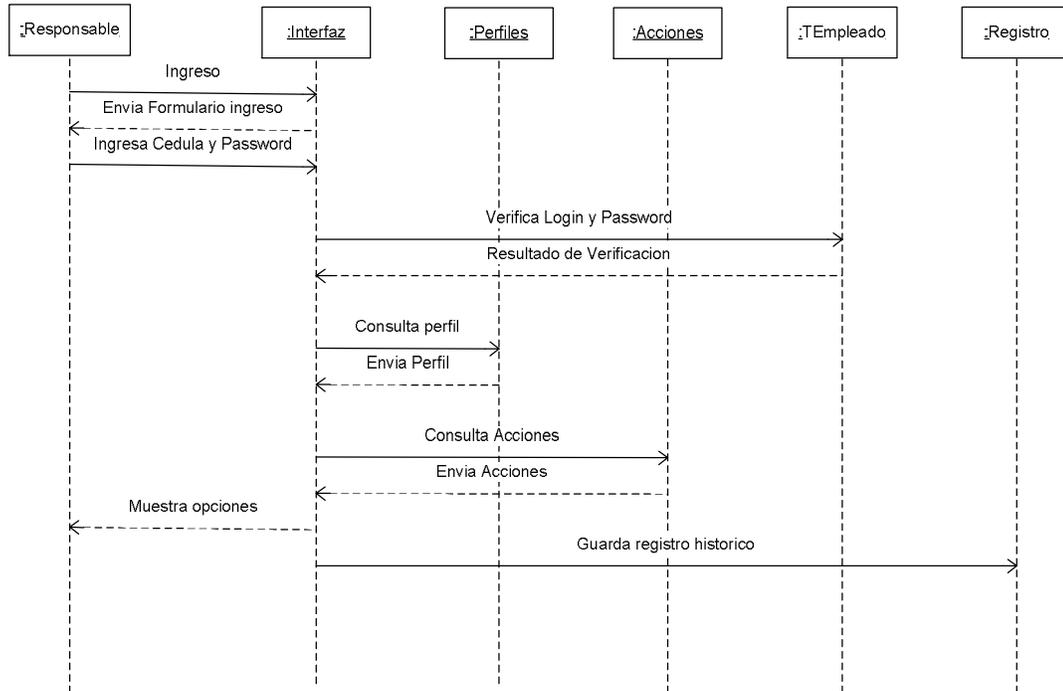
Se debe guardar un registro histórico de cuándo un usuario se desconecta del sistema.

De requerimientos: R9, R10

De Casos: 023, 024, 025, 026

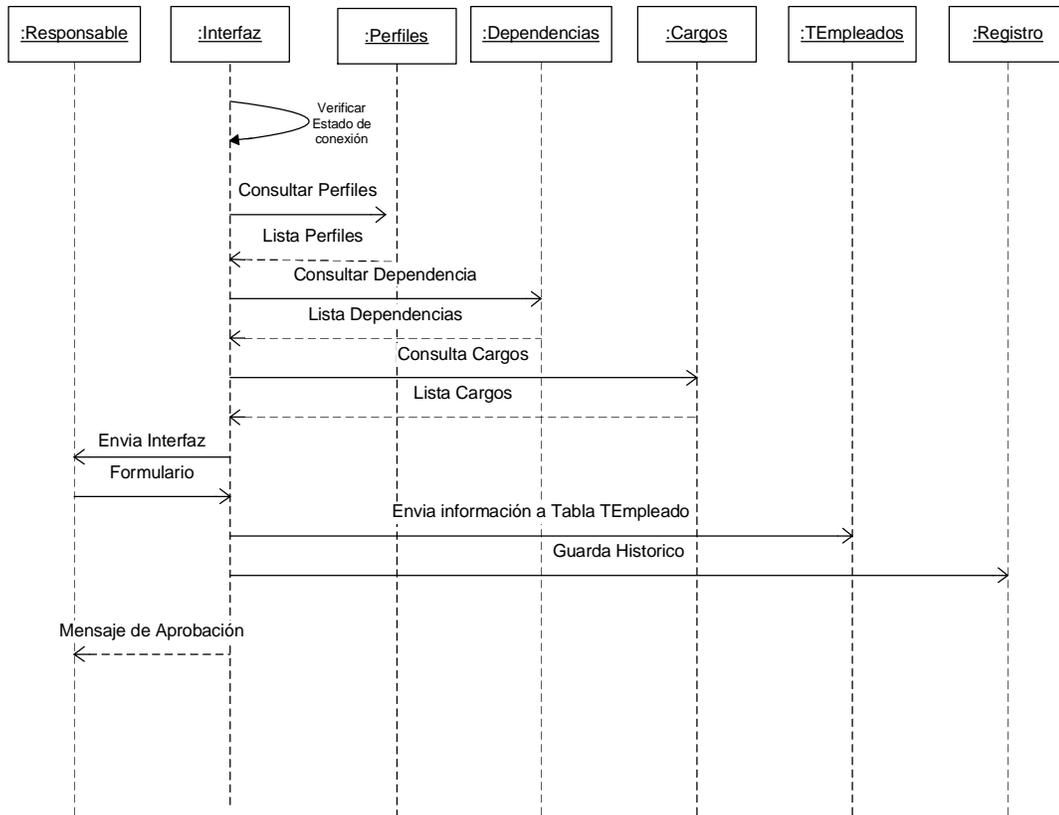
3. DIAGRAMA DE SECUENCIAS

3.1 Entrada al sistema



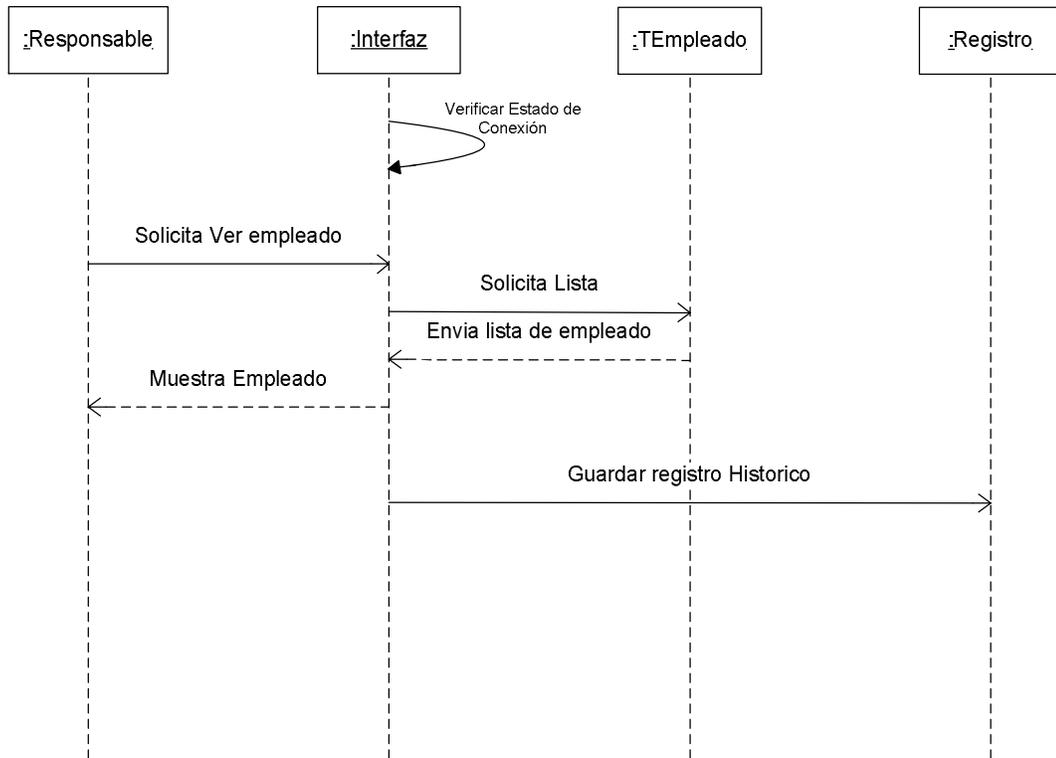
Para ingresar al sistema se envía un formulario de ingreso se llenan los datos se digita login y password se verifican, se consultan los perfiles y las acciones y se guarda un registro histórico.

3.2. Agregar Empleado



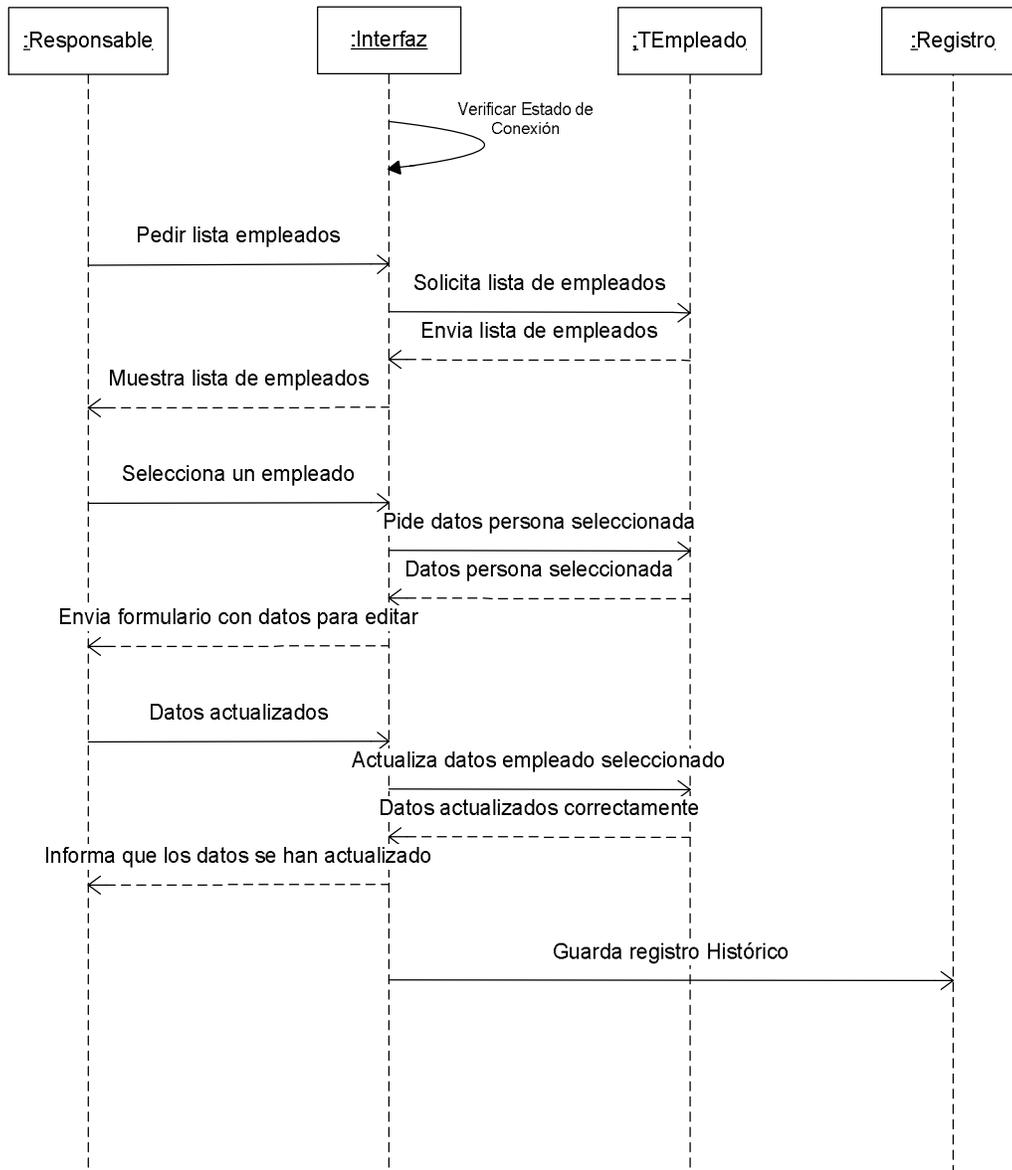
Para agregar a un empleado se presenta la opción de un formulario de ingreso en el cual se debe guardar un registro histórico de cualquier empleado que use esta opción en la base de datos, así como la hora y fechas de la transacción al guardar un empleado se debe seleccionar a qué dependencia será asignado, este usuario debe ser asignado a uno de los perfiles que existen, debe ser ingresado como Activo, y se le debe escoger uno de los cargos disponibles en la Base de datos.

3.3 Ver Empleado



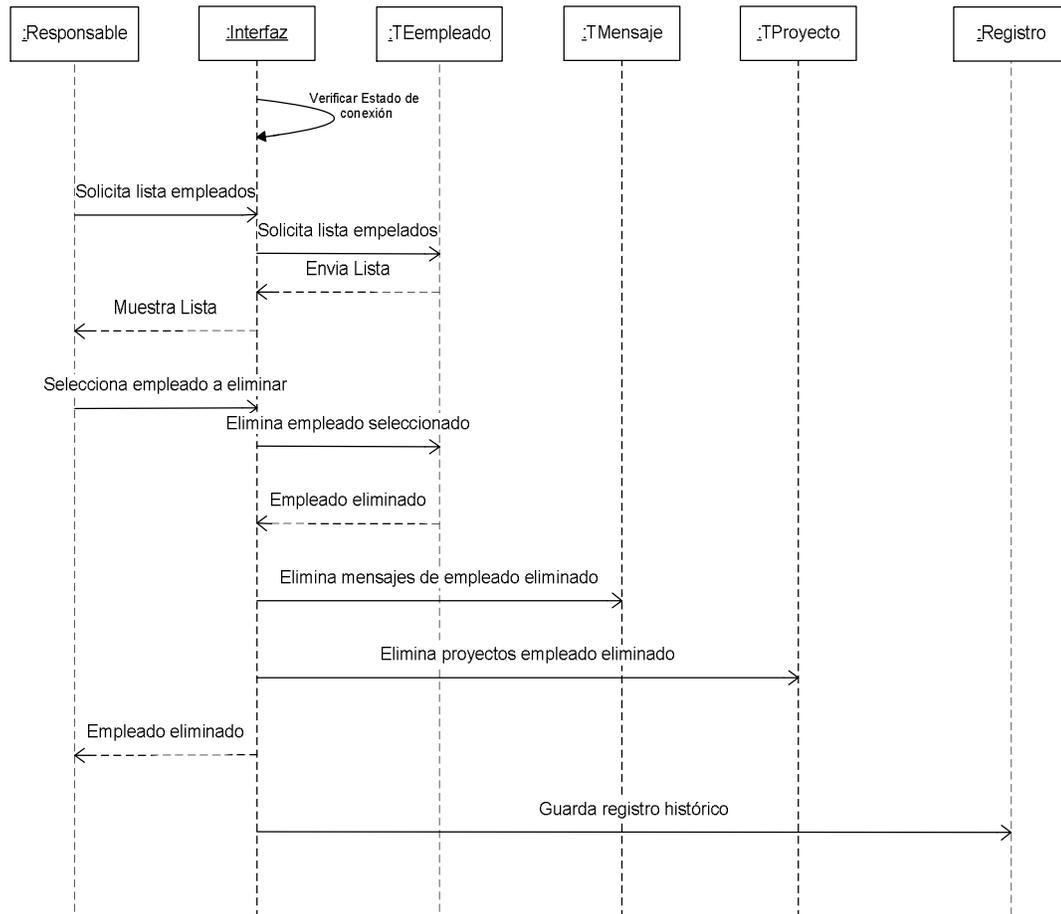
Para ver la lista de empleados se les debe presentar la opción de ver todos los empleados y se muestra la lista, se debe guardar un registro histórico de cualquier empleado que use esta opción en la base de datos

3.4. Editar Empleado



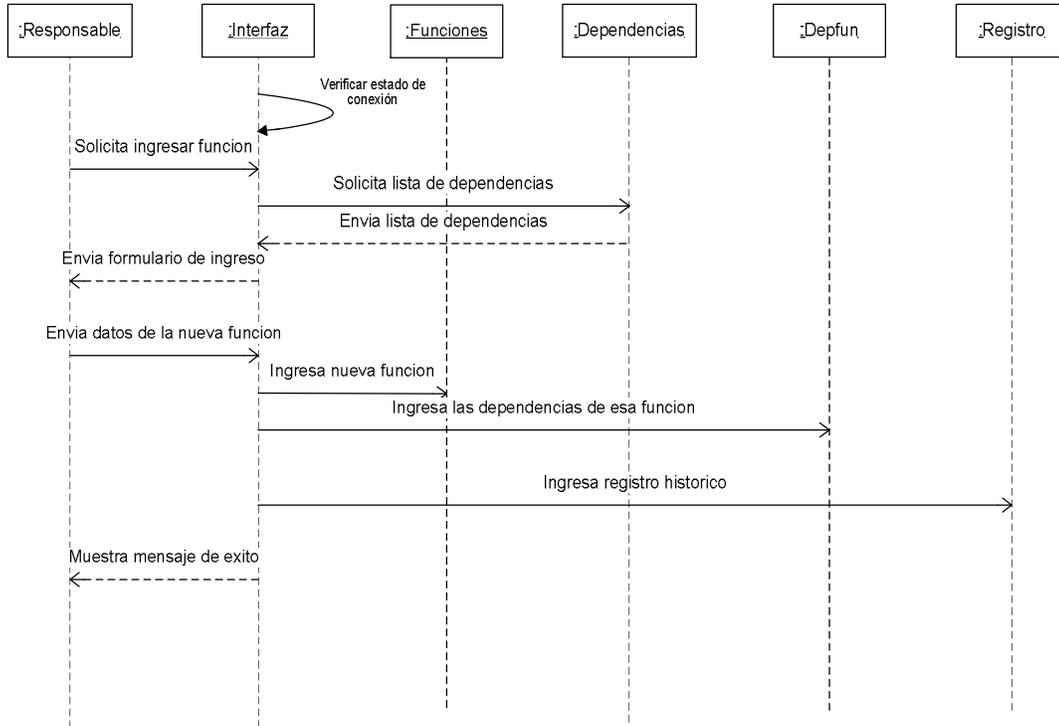
Para cambiar los datos de un empleado se debe verificar el empleado solicitando una lista de empleados, se selecciona el empleado y sus datos a editar, se editan los datos y se guarda en un registro histórico tanto los datos editados como los de la persona que lo edito.

3.5. Eliminar Empleado



Eliminar un usuario es una opción que permite a quienes tengan el perfil necesario para hacerlo desaparecer el registro de la base de datos, pero se debe tener en cuenta que al eliminar un empleado se eliminan los mensajes que ha enviado y también se le desvincula de los proyectos en los cuales esta inscrito en ese momento también al igual que en todos los casos de uso se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

3.6 Agregar Funciones

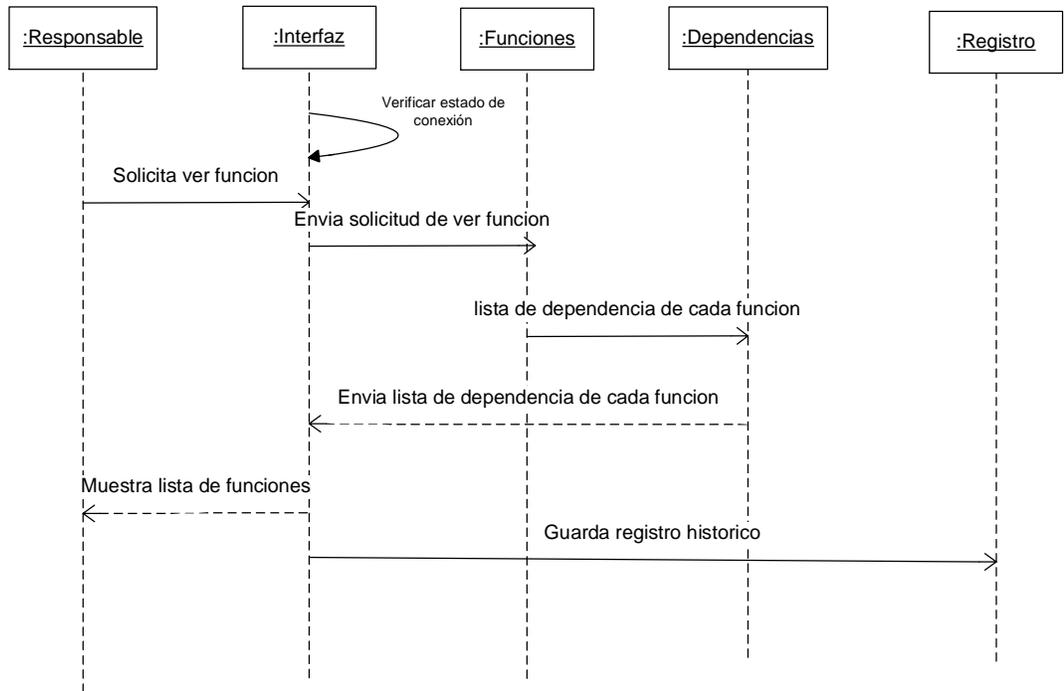


Los usuarios que les sea permitido agregar funciones se les debe presentar la opción de un formulario de ingreso en el cual Debe tener dos partes

- 1: Los datos específicos de las funciones.
- 2: Las dependencias que aplican para tener dicha función

Por lo tanto al guardar una función se deben seleccionar las dependencias para las cuales esa función será activa, se debe guardar un registro histórico de cuándo un usuario agrega una función, guardando su fecha y hora, y el usuario que ejecutó dicha acción.

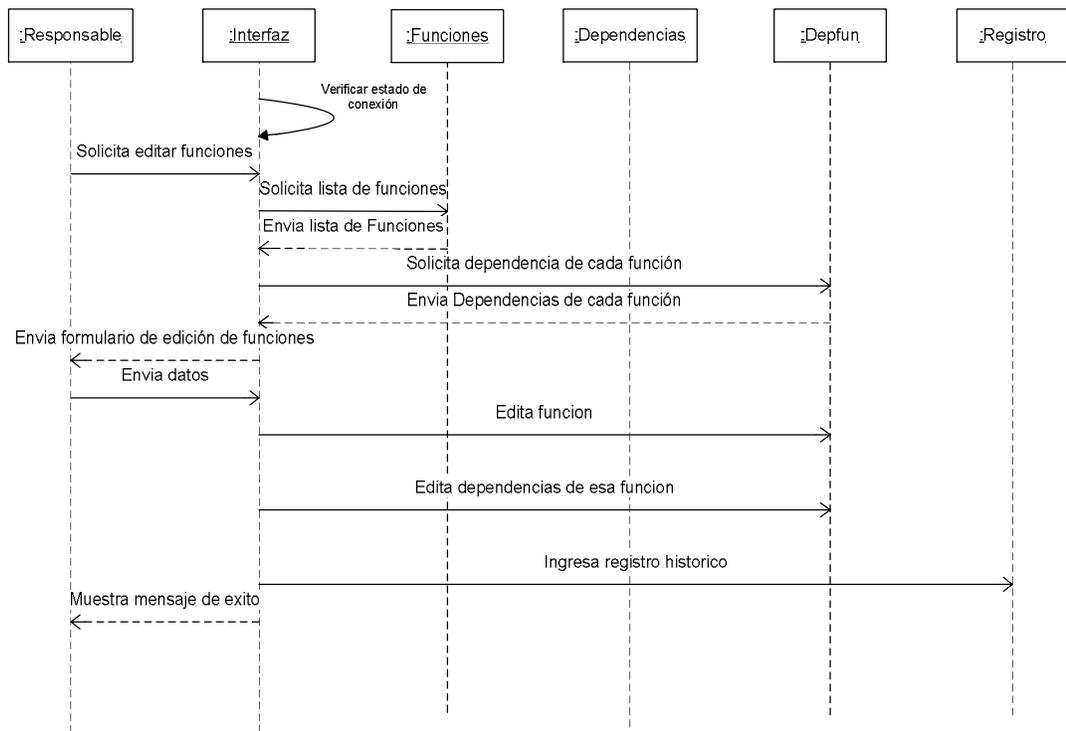
3.7 Ver Funciones



Listar funciones es una opción en la cual, los empleados que puedan ejecutarla, puedan ver una lista de todas las funciones que tienen las diferentes dependencias pero con la opción de ver una lista de dependencias por cada una de las funciones listadas.

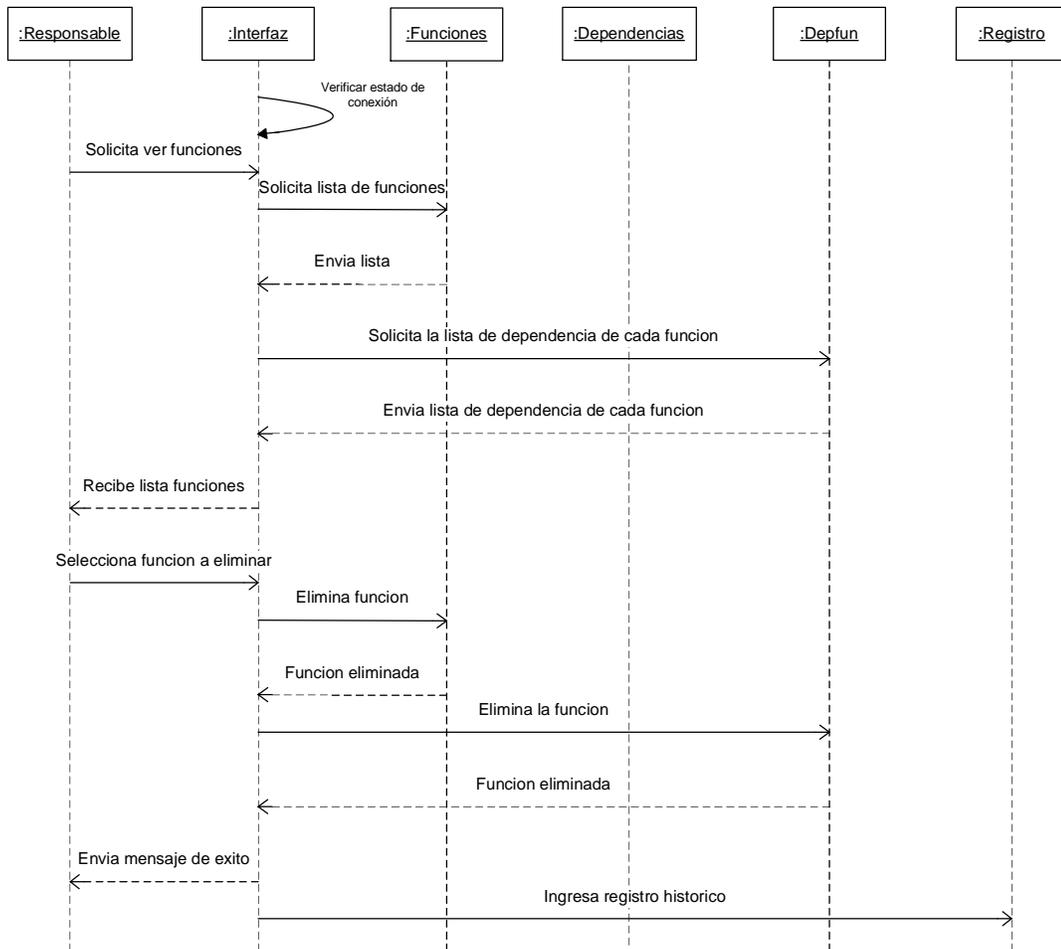
Se debe guardar un registro histórico de cuándo un usuario ve la lista de funciones, guardando su fecha y hora, y el usuario que ejecutó dicha acción

3.8 Editar Funciones



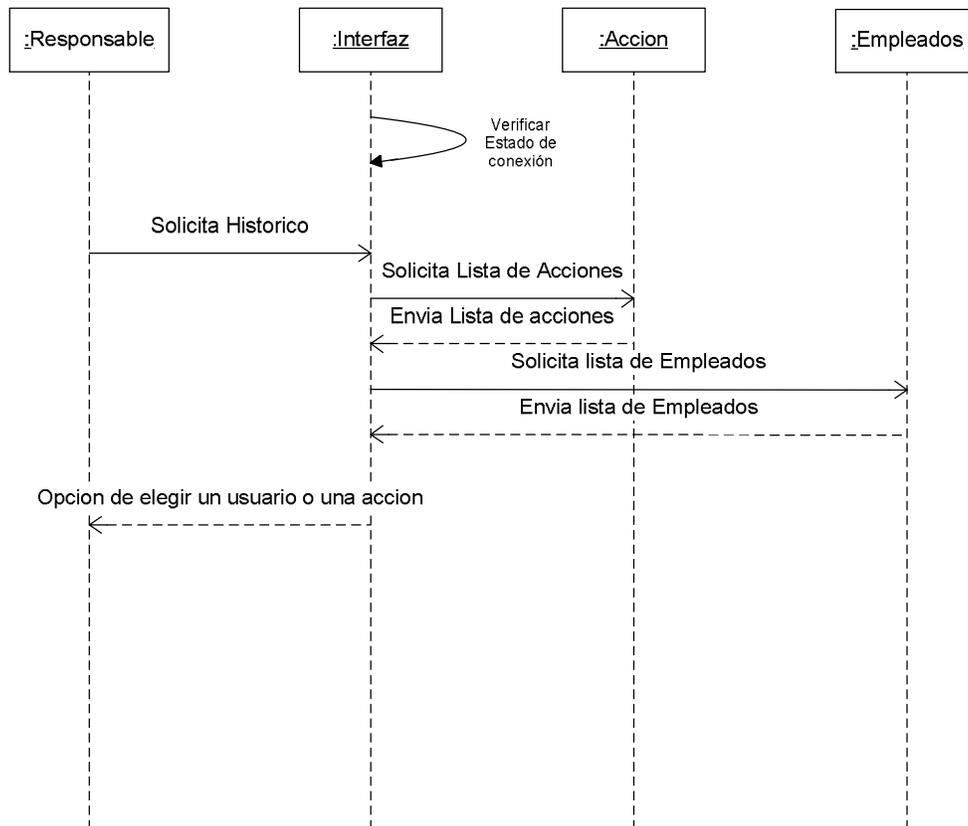
Para editar funciones se debe mostrar una lista de todas las funciones creadas en el sistema, y se debe presentar la posibilidad de que dicho usuario elija una función y pueda cambiar la información de dicha función o cambiar las dependencias a las cuales afecta dicha función, se debe guardar un registro histórico de cualquier empleado que edite algún proyecto, con fecha hora y una descripción de lo que ese usuario hizo.

3.9 Eliminar Función



Para eliminar una función del sistema se deben listar las funciones existentes, y luego se debe seleccionar la que se debe eliminar, de igual manera se debe eliminar cualquier registro que use esa función, se debe desvincular las dependencias que tenían esa función, además, se debe guardar un registro histórico de cuándo un usuario elimina alguna función, guardando su fecha y hora.

3.10. Histórico

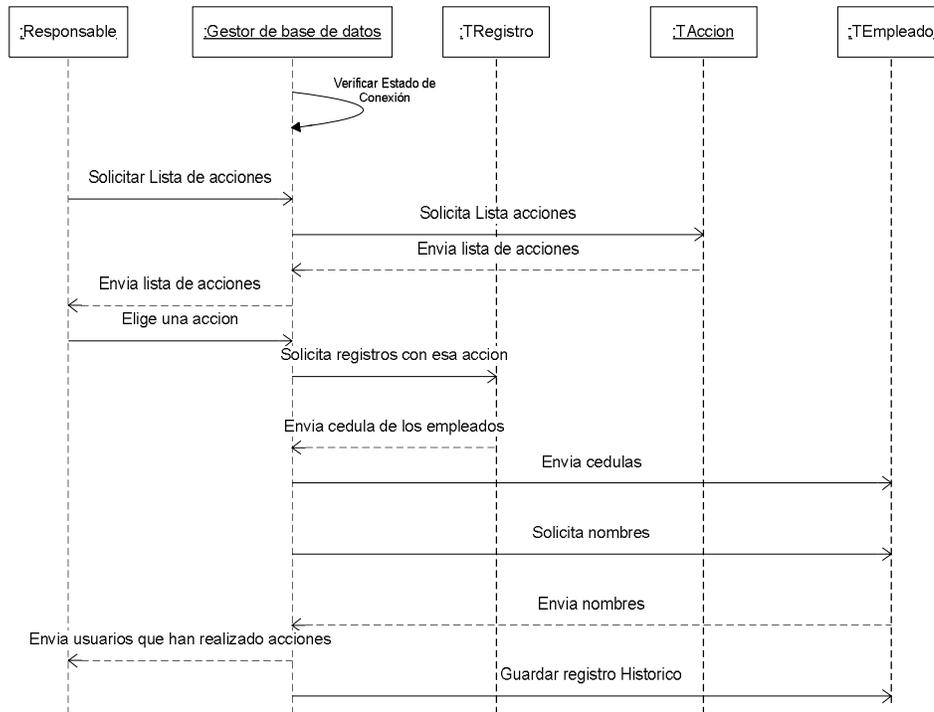


El histórico es una opción que permite Mostrar lo que un usuario determinado ha realizado sobre el sistema, o que usuarios han ejecutado una acción determinada, por tal motivo lo primero que se debe hacer es mostrar una opción en la cual la persona elija si quiere ver lo que un usuario ha realizado o los usuarios que han realizado una acción determinada.

Si el usuario elije la primera opción se debe desplegar una lista de los usuarios del sistema con el fin de que se pueda elegir uno de ellos y de esa manera poder ver todo el histórico de acciones de ese usuario.

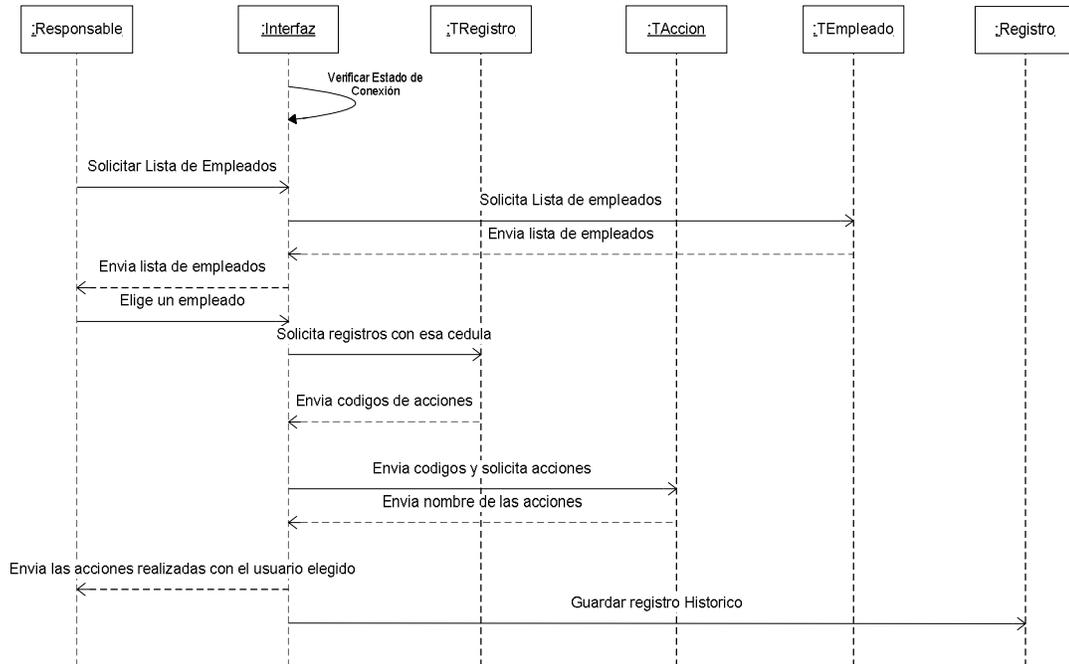
Si el usuario elije la segunda opción (Histórico de acciones) se debe mostrar una lista de las acciones disponibles en el sistema y seleccionar una de ellas para ver quienes de los usuarios que tienen login y password han ejecutado esa acción, además no sólo se debe decir qué acción ejecutó, si no que también se debe guardar una descripción de lo que hizo así como su fecha y la hora en que se ejecutó. Como Consultar el histórico es una acción, entonces se debe guardar un registro histórico de cualquier empleado que consulte ya sea el histórico de acciones o de un usuario del sistema.

3.11. Usuarios por acción



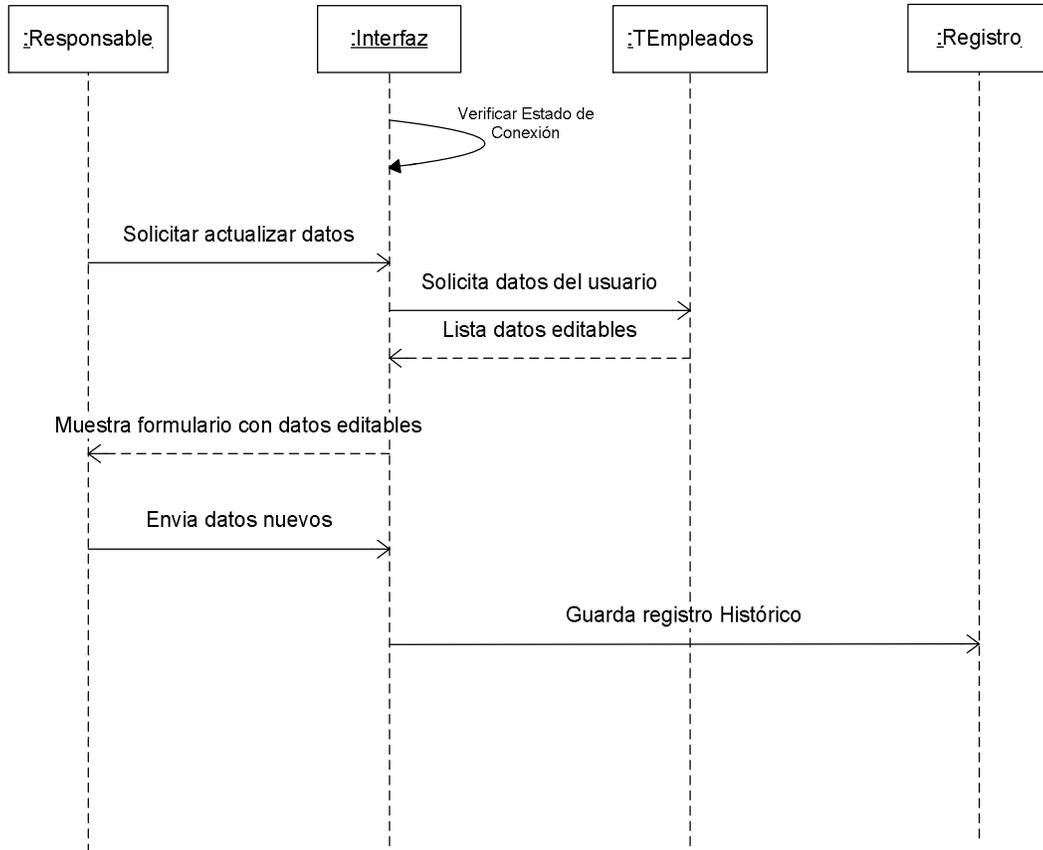
Se solicita una lista de los usuarios del sistema con el fin de que se pueda elegir uno de ellos y de esa manera poder ver todo el histórico de acciones de ese usuario.

3.12 Acciones por usuario



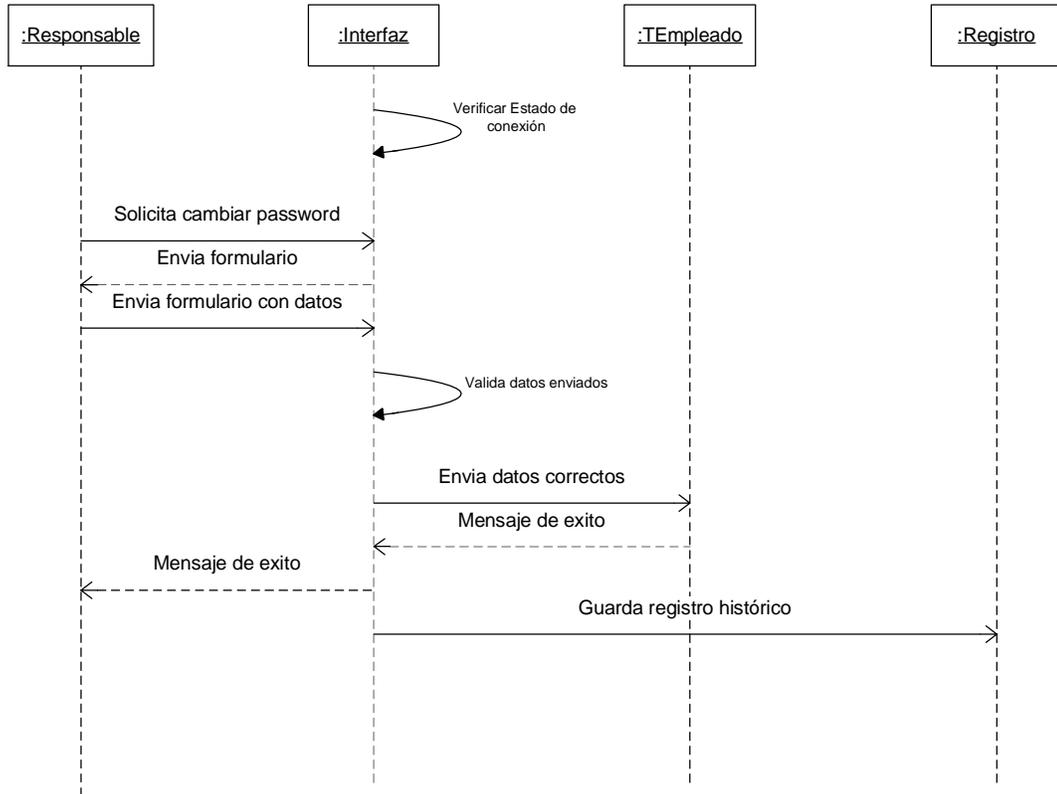
Se muestra una lista de las acciones disponibles en el sistema y seleccionar una de ellas para ver quienes de los usuarios que tienen login y password han ejecutado esa acción, además no sólo se debe decir qué acción ejecutó, si no que también se debe guardar una descripción de lo que hizo así como su fecha y la hora en que se ejecutó. Como Consultar el histórico es una acción, entonces se debe guardar un registro histórico de cualquier empleado que consulte ya sea el histórico de acciones o de un usuario del sistema.

3.13. Actualizar Información



Los usuarios deben tener la opción de actualizar su información personal, pero sólo alguna parte de la información, pues no podrá por ejemplo: activarse, desactivarse, cambiarse de dependencia, ya que esto lo hace el responsable de este usuario cuando edite información de usuario, él debe actualizar su información personal, tal como teléfono, celular, email, dirección, debe guardar un registro histórico de cuando un usuario haga algún tipo de edición de su información personal.

3.14. Cambiar Password

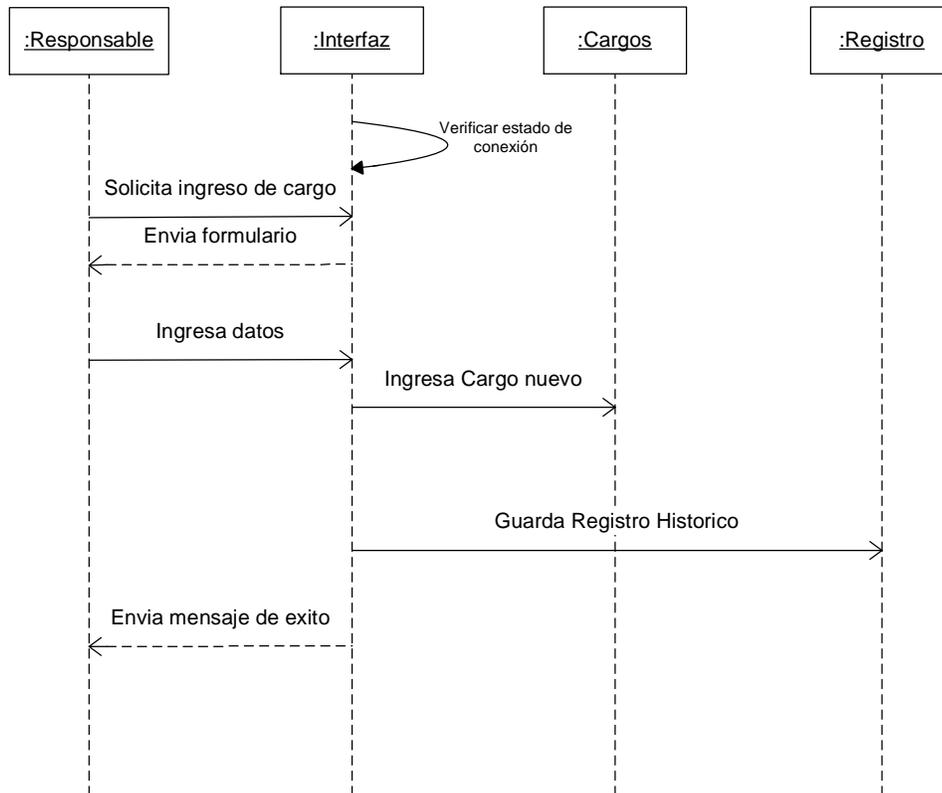


Para efectos de seguridad de la información y de la gestión de los diferentes usuarios del sistema, cada usuario debe tener un password, único y que además se debe guardar en la base de datos usando encriptación, ya que este password, no debe ser visible para nadie que tenga acceso a las tablas de la base de datos. A pesar de esto, cualquier usuario debe cambiar su password como política de la alcaldía y por ello cada usuario debe tener habilitada esta opción, en este caso se debe solicitar la siguiente información:

Password antiguo: con el fin de comprobar la identidad del usuario al momento de hacer un cambio.

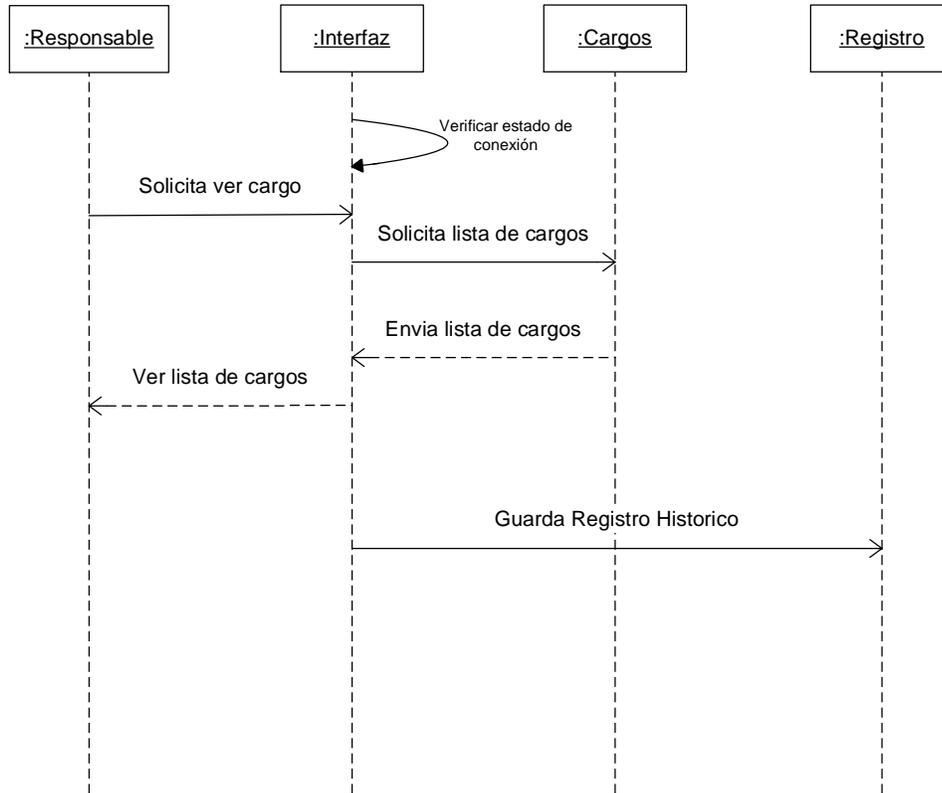
Nuevo password: El password que desea, el cual se debe solicitar dos veces para eliminar la probabilidad de cualquier error sintáctico en el momento de crear el password nuevo. Es muy importante validar esta información de manera correcta dado que este es un proceso crítico. Y se debe guardar un registro histórico de cuando un usuario cambia su password.

3.15 Agregar Cargo



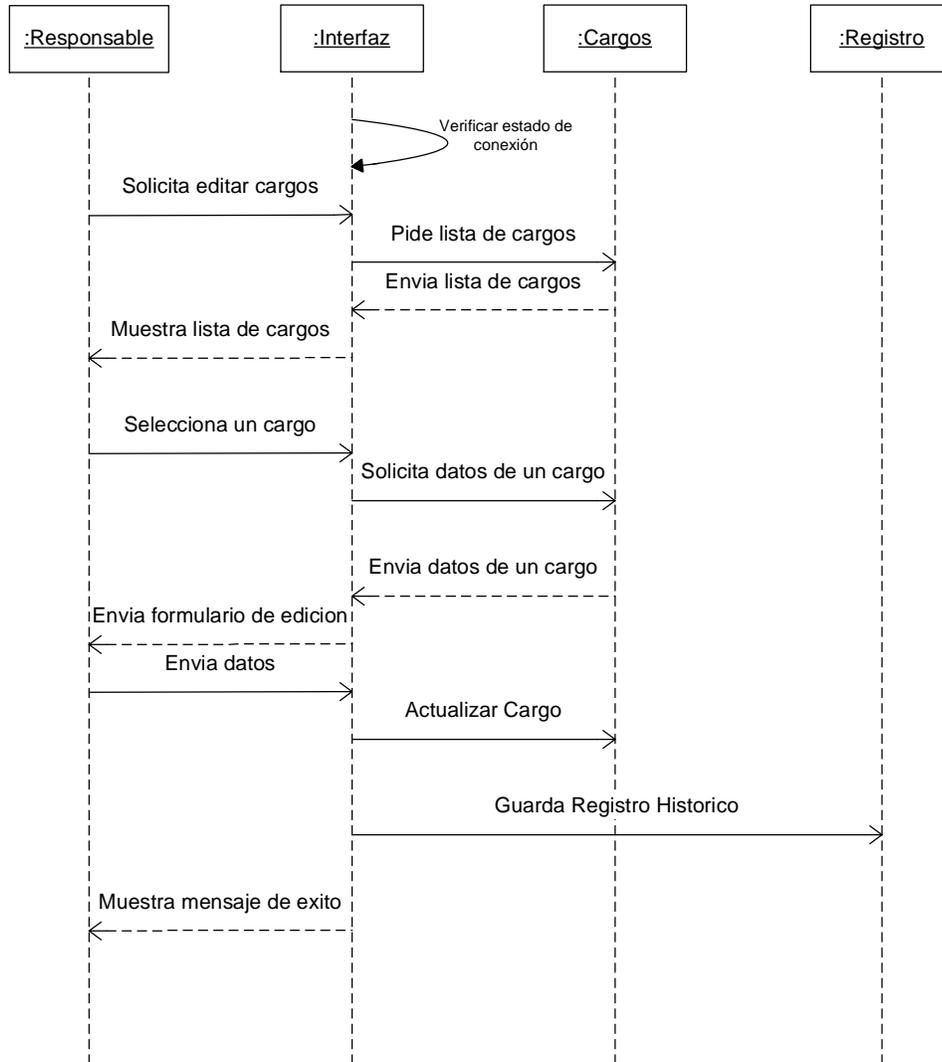
Para agregar un cargo se presenta la opción de un formulario de ingreso en el cual se ingresan los datos para un nuevo cargo, se debe guardar un registro histórico de cuándo un usuario agrega un cargo guardando su fecha y hora, y el usuario que ejecutó dicha acción.

3.16. Ver cargos



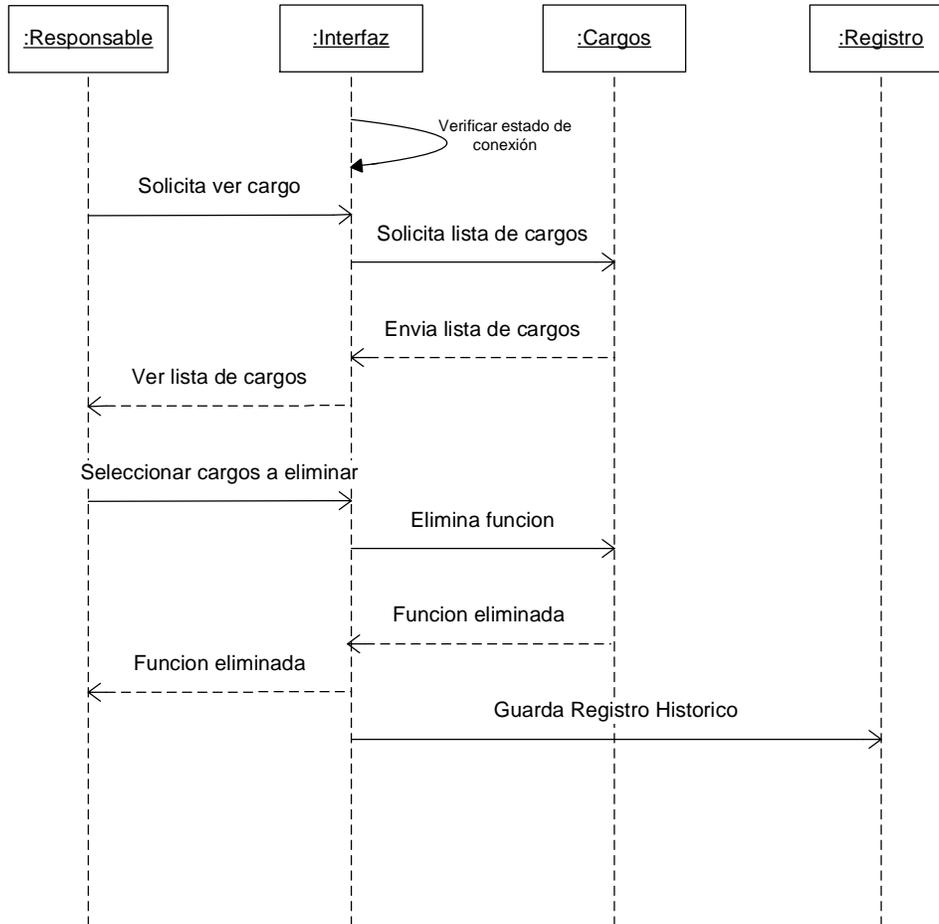
Ver cargos es una consulta que se ejecuta sobre la base de datos y arroja una lista de los diferentes cargos que en la base de datos existen, y que están disponibles para ser usados por los diferentes empleados de la base de datos. Se debe guardar un registro histórico de cuándo un usuario ve la lista de cargos, guardando la fecha y hora, y el usuario que ejecutó dicha acción

3.17. Editar Cargos



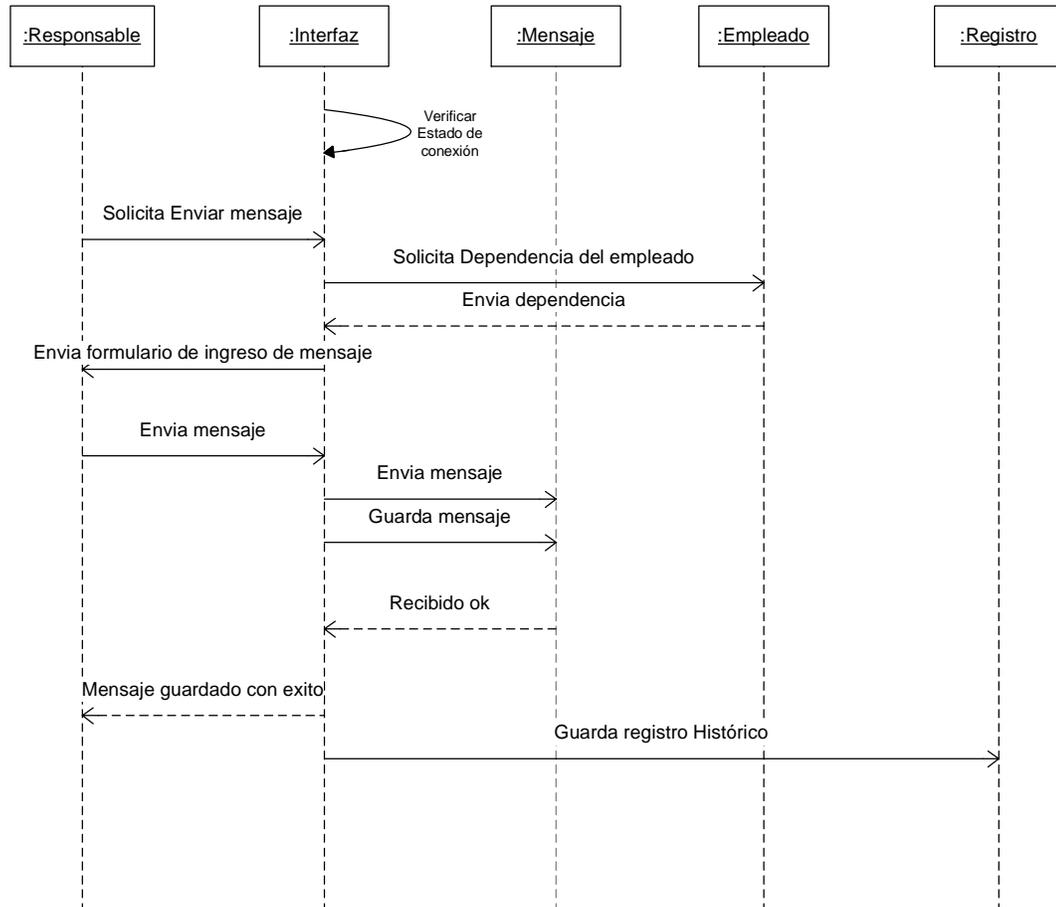
Editar cargos, es una opción que permite al usuario que tenga el perfil requerido para esta operación, listar completamente los cargos disponibles en nuestra base de datos, y de esa manera se le permite al usuario elegir entre uno de los diferentes cargos existentes, para que ese cargo salga en campos editables, y por ultimo el usuario pueda actualizar la información de dicho registro se debe guardar un registro histórico de cualquier empleado que edite algún cargo, con fecha hora y una descripción de lo que ese usuario hizo.

3.18 Eliminar Cargos



Para eliminar un cargo del sistema se debe listar los cargos existentes, y luego se debe seleccionar la que se debe eliminar, de igual manera se debe eliminar cualquier registro que use esa función, se deben desvincular las dependencias que tenían esa función, además, se debe guardar un registro histórico de cuando un usuario elimina alguna función, guardando su fecha y hora.

3.18 Crear Mensaje



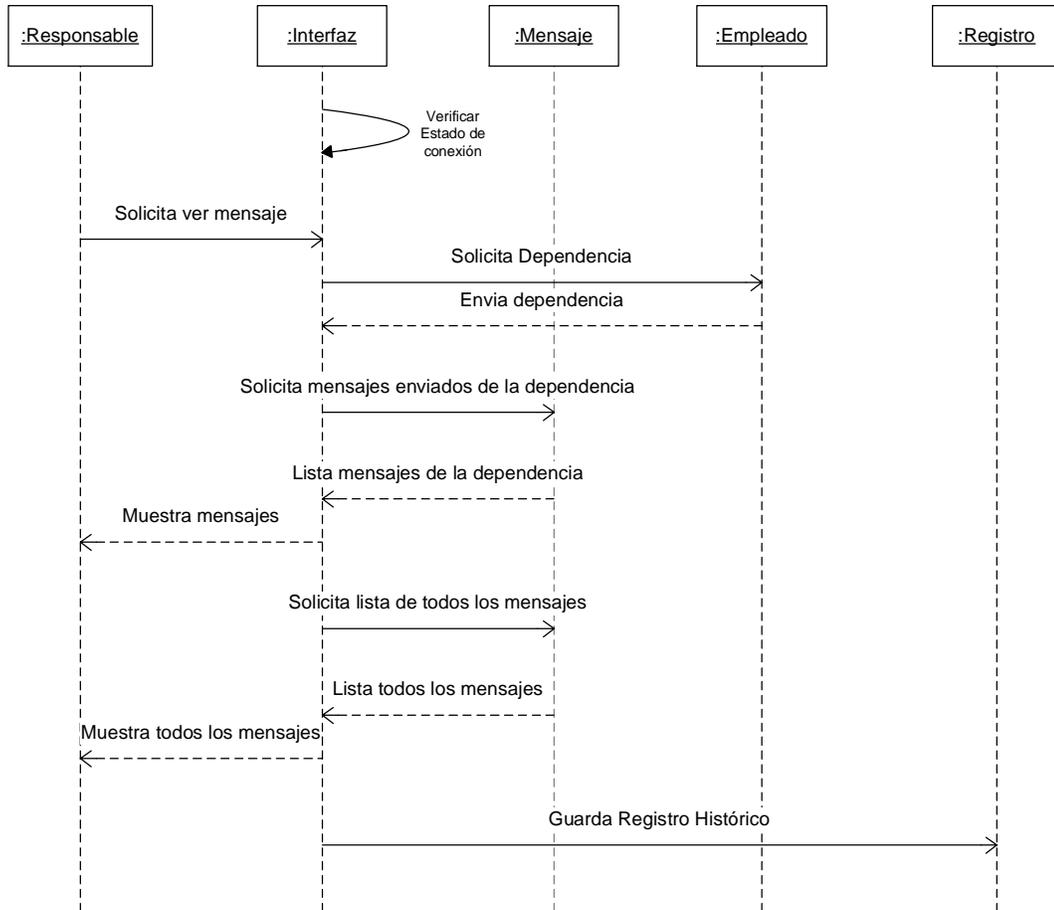
Para enviar mensajes se debe presentar la opción, de enviar un mensaje general que puede tener uno de dos ámbitos:

Para la dependencia, cualquier empleado de una dependencia debe estar en capacidad de enviar mensajes a sus compañeros de dependencia

Para la alcaldía en general: cualquier empleado puede estar en capacidad de enviar un mensaje a todos los empleados de la alcaldía.

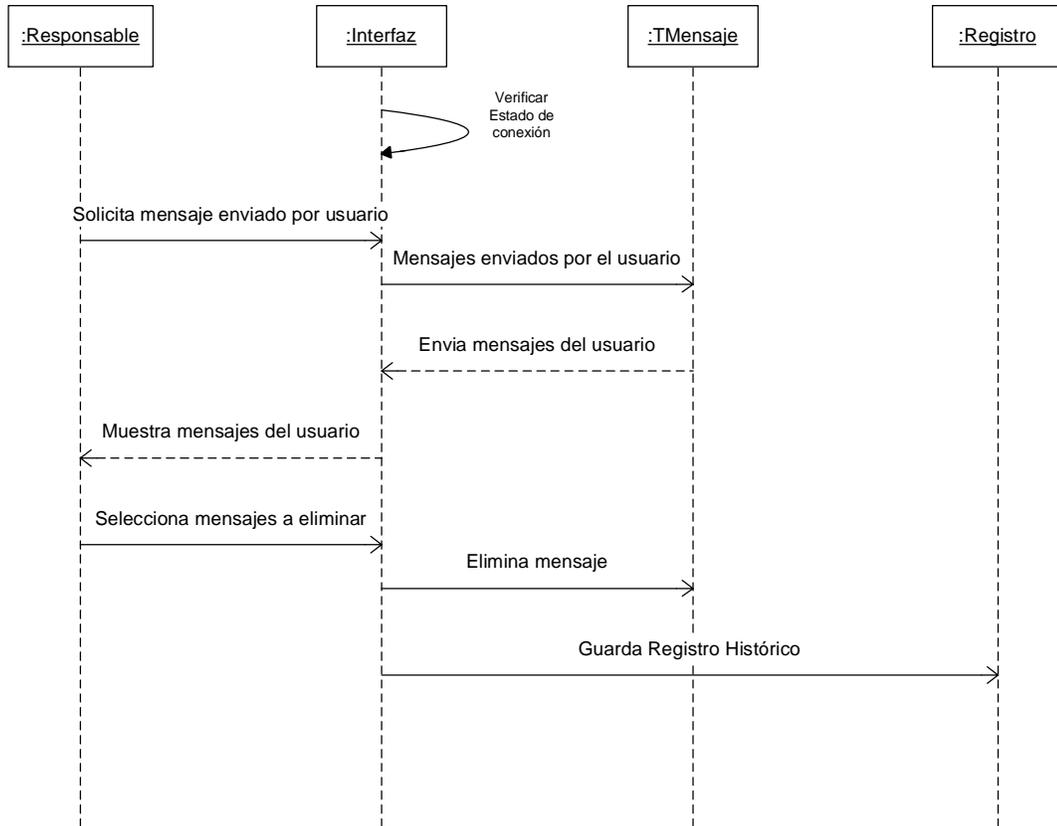
Un mensaje se envía por medio de un formulario y el tipo de ámbito o grupo al cual se le ve a enviar dicho mensaje debe ser de escogencia del usuario, se debe guardar un registro histórico de cuando un usuario deja un mensaje de cualquier ámbito, guardando su fecha y hora.

3.19 Ver Mensaje



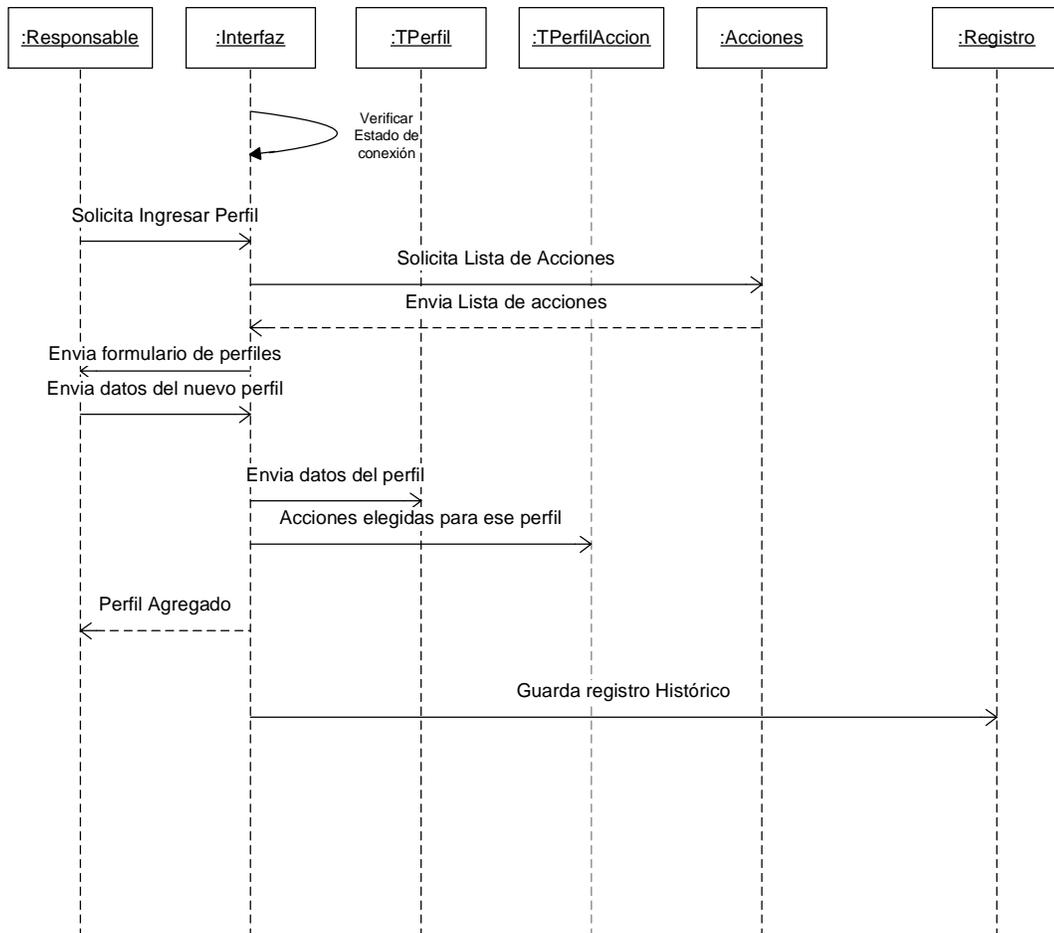
Para leer los mensajes enviados a un tablón de mensajes donde se discrimina, los mensajes enviados a la dependencia del usuario, para lo cual se debe detectar la dependencia de dicho usuario, y luego los mensajes enviados a todos los funcionarios en general de la alcaldía, se debe guardar un registro histórico de cuando un usuario lee los mensajes enviados a cualquier ámbito, guardando su fecha y hora.

3.19 Eliminar Mensaje



Para eliminar un mensaje primero se debe verificar la lista de mensajes enviado por cualquier usuario y de acuerdo a la pertinencia o a algún criterio de significado de dicho mensaje puede eliminarlo, para lo cual se muestra la lista de mensajes organizados por fecha y se seleccionan los que se van a eliminar, se debe guardar un registro histórico de cuando un usuario elimina los mensajes de cualquier otro usuario, guardando su fecha y hora.

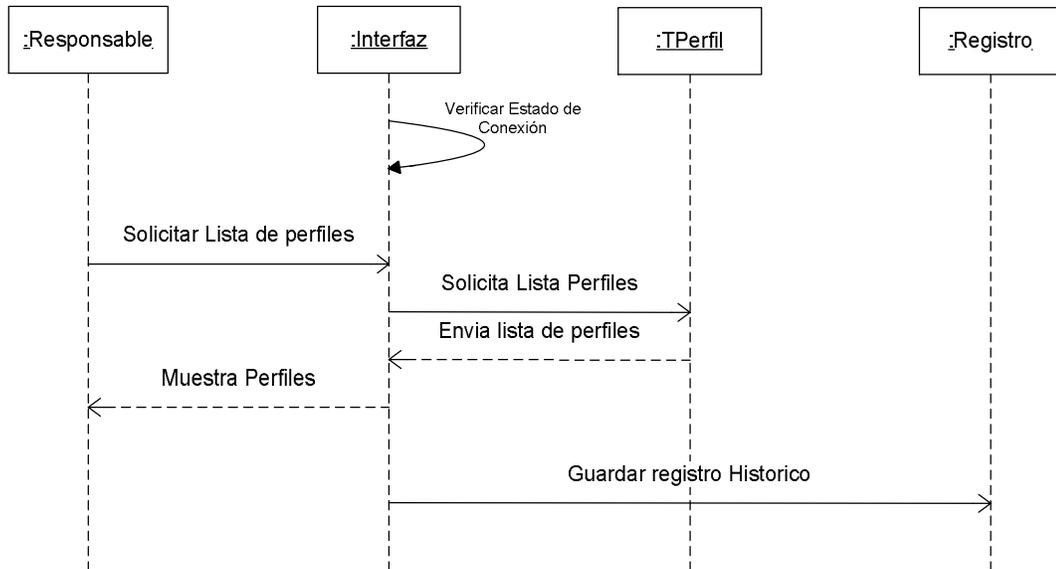
3.20 Crear Perfiles



Crear un perfil permite crear una agrupación lógica de funciones (acciones) que permite segmentar el tipo de usuario de acuerdo a lo que pueda hacer sobre el sistema: ejm

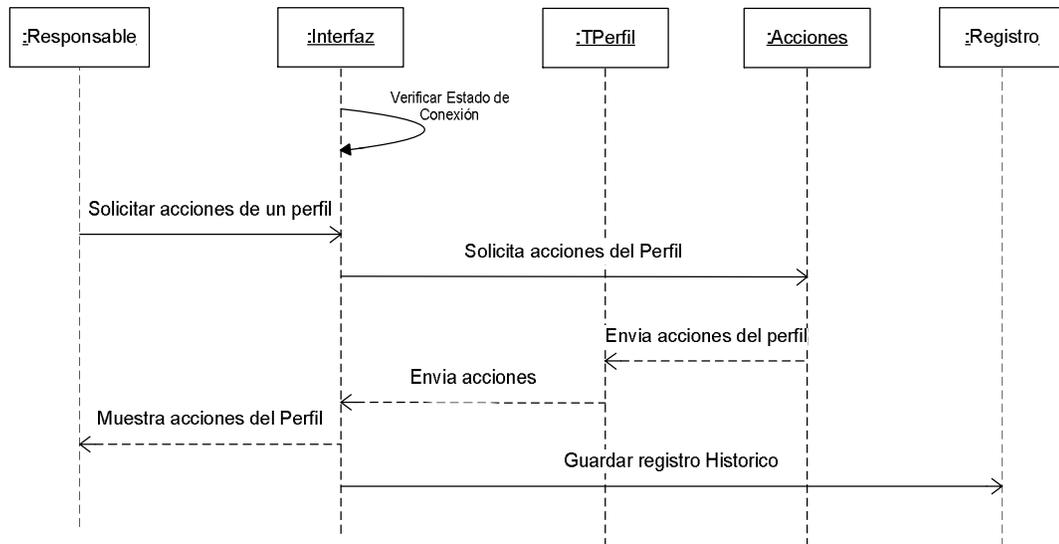
Andrés es de perfil “consultor” los consultores pueden, hacer X y Y pero no Z, al crear un usuario; a este usuario se le agrega a un perfil determinado y al igual que en todos los casos de uso se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

3.21. Ver Perfil



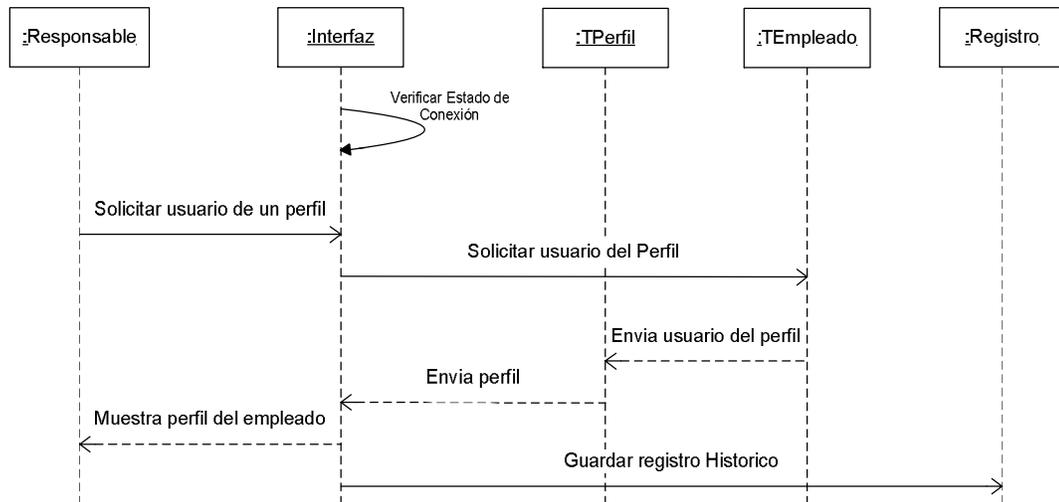
Ver un perfil permite ver la lista de perfiles creados, pero además es muy importante conocer que usuarios tiene un perfil determinado, también es necesario poder preguntar al sistema que acciones puede un perfil dado ejecutar y al igual que en todos los casos de uso se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

3.22 Acciones de un perfil



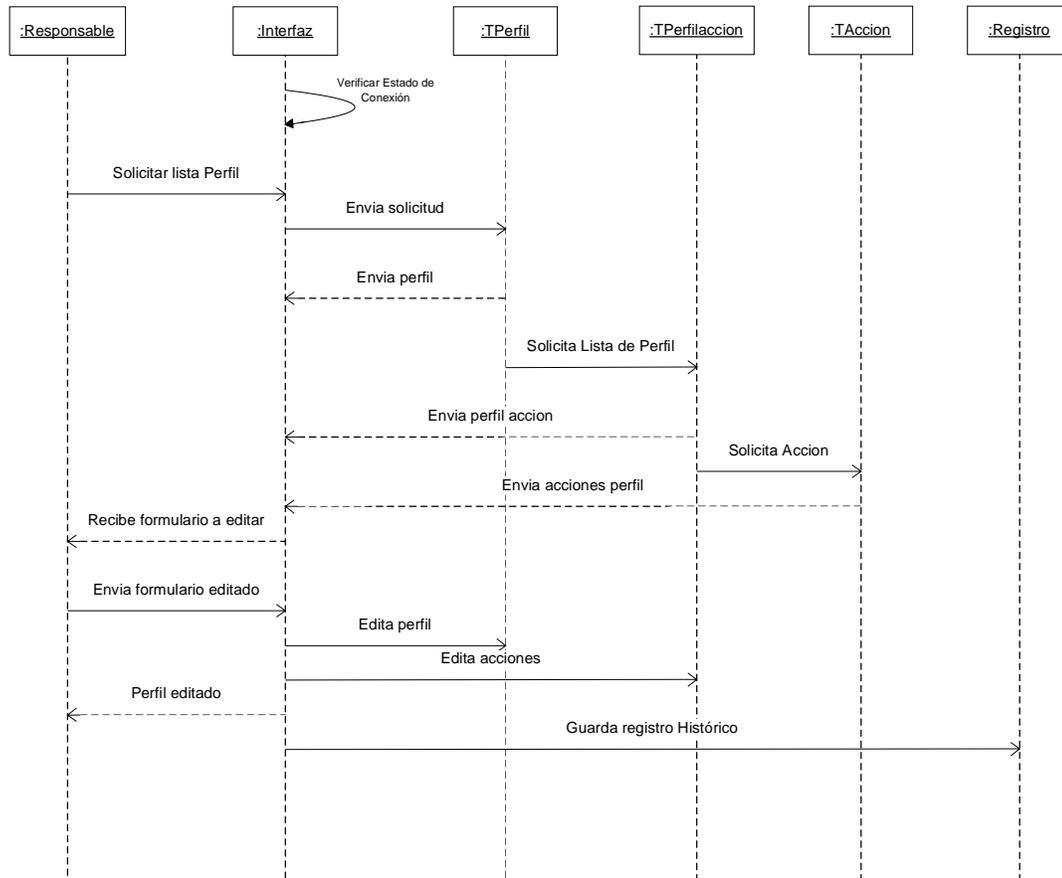
Para ver las acciones de un perfil se solicitan todas las acciones que existan por un perfil se consultan y se guardan en el registro histórico.

3.23 Usuarios por perfil



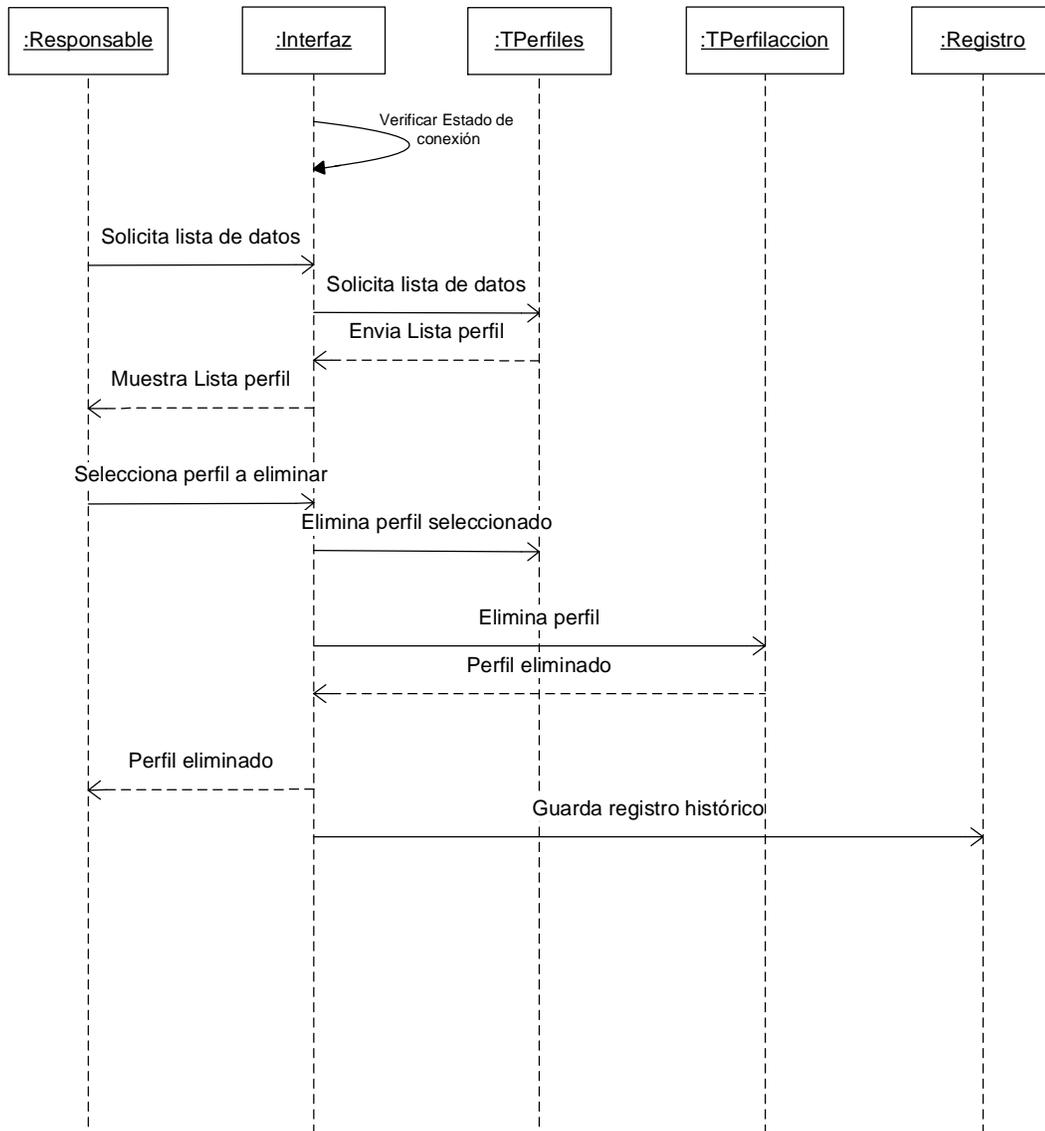
Para consultar los usuarios por perfil se solicitan los usuarios que pertenezcan a un perfil determinado se consultan y esta acción es guardada en un registro histórico.

3.24 Editar Perfiles



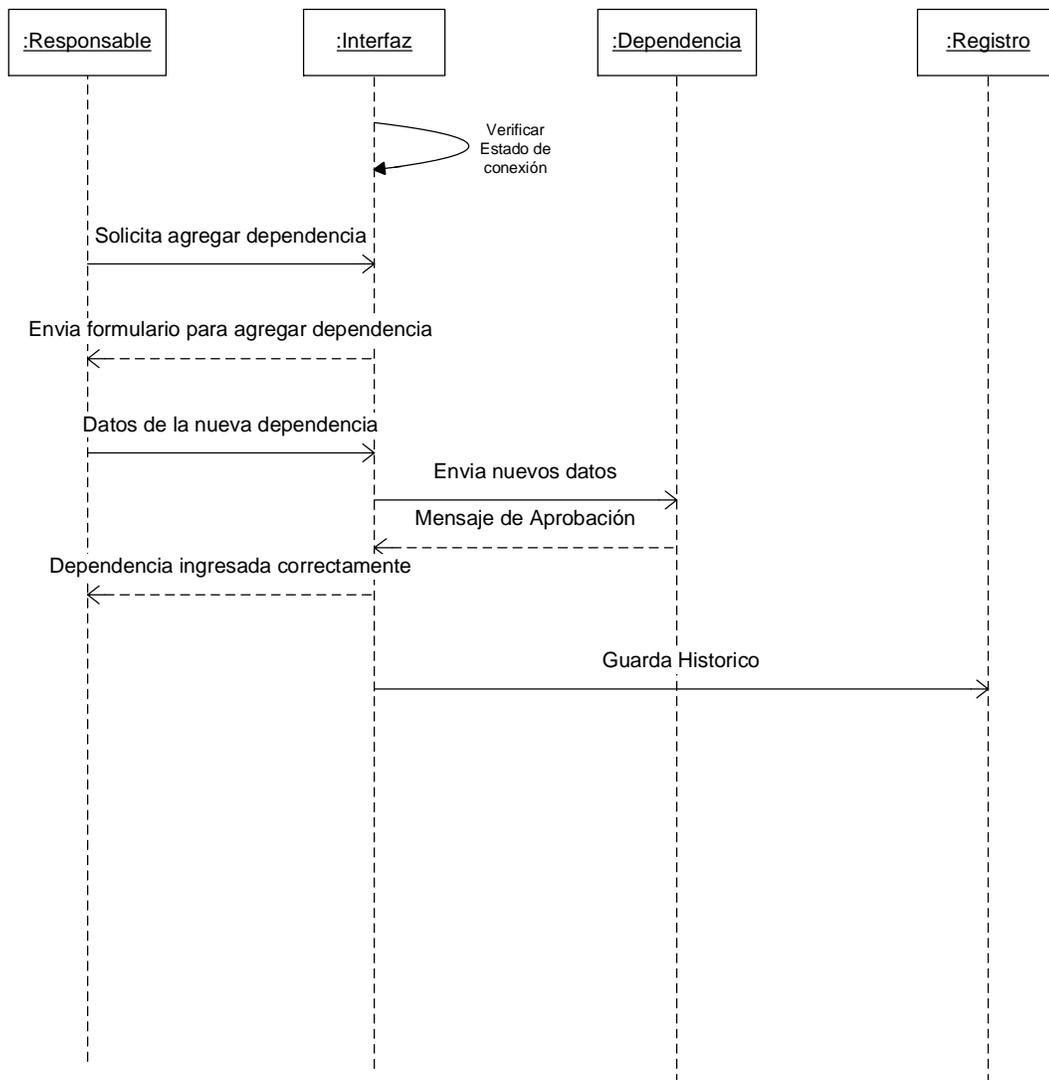
Para editar perfiles debe mostrar una lista de todos los perfiles existentes y de esa manera, el usuario debe seleccionar un perfil para editar, luego se le debe mostrar un formulario con los campos evitables, el cual tiene la información del perfil que se desea eliminar, se debe tener en cuenta que editar un perfil puede también implicar agregar o eliminar acciones que ese perfil puede realizar, por tal motivo se deben además mostrar las acciones de dicho perfil para que el usuario elija si lo desea dejar eliminar o agregar otra acción a dicho perfil, se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

3.25. Eliminar Perfiles



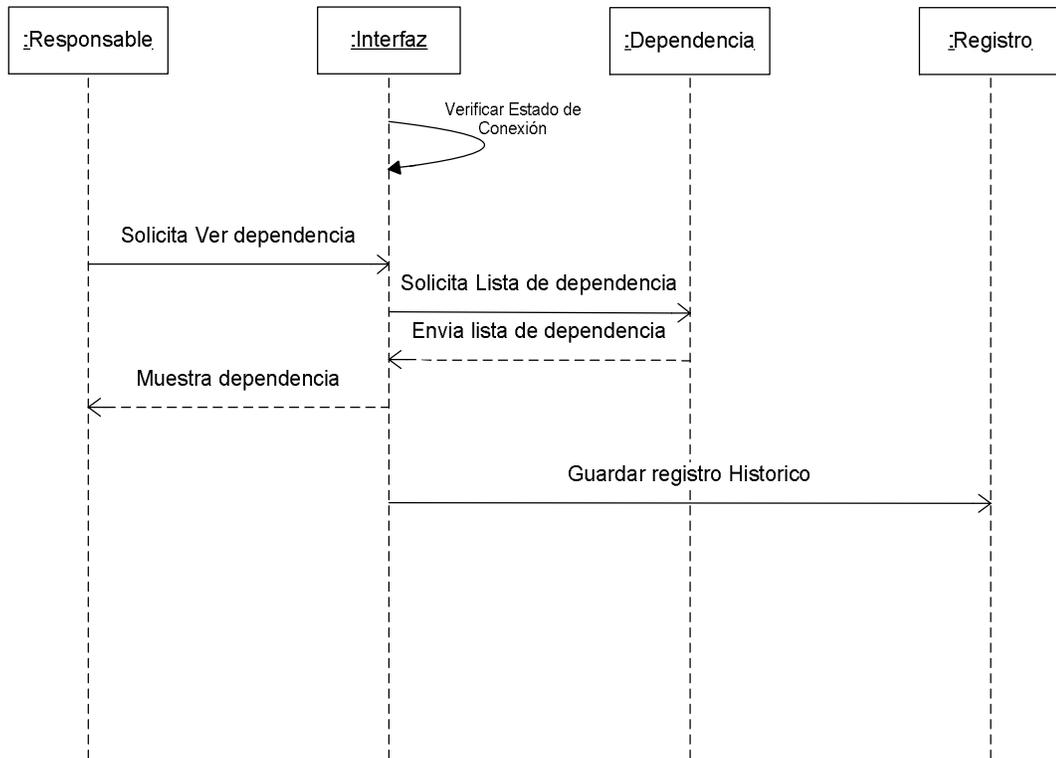
Para eliminar perfiles se debe permitir una lista de todos los perfiles existentes y de esa manera, el usuario debe seleccionar un perfil para eliminar, además se debe tener en cuenta que se debe eliminar cualquier rastro que hubiese existido en el sistema de dicho perfil, por ejemplo: en su relación con acciones, si hay algún usuario con el perfil que se desea eliminar entonces se debe mostrar el mensaje de que primero se debe cambiar el perfil de los usuarios X y Y dado que ese perfil va a ser eliminado, se debe guardar un registro histórico de cualquier empleado que elimine un perfil.

3.26. Agregar Dependencia



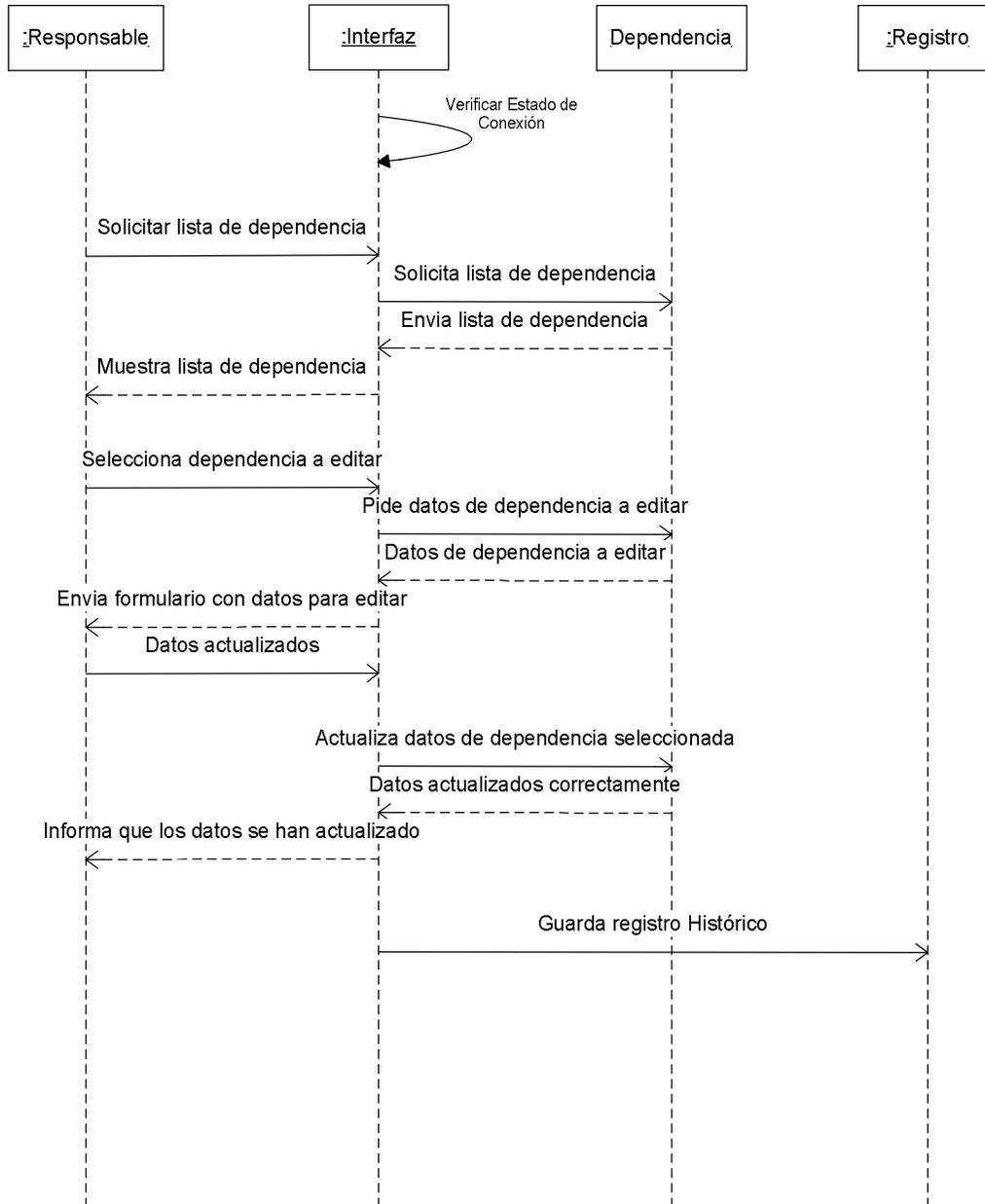
Para agregar dependencias a la base de datos, se debe presentar un formulario, desde el cual se ingresen los datos de una dependencia y de esta manera los próximos empleados al ser ingresados al sistema encontrarán la nueva dependencia dentro de las opciones a las cuales se asignará, Cuando alguien agregue un usuario se debe guardar un registro histórico guardando su fecha y hora, además quién agregó al sistema y quién ha sido agregado al sistema.

3.27 Ver Dependencia



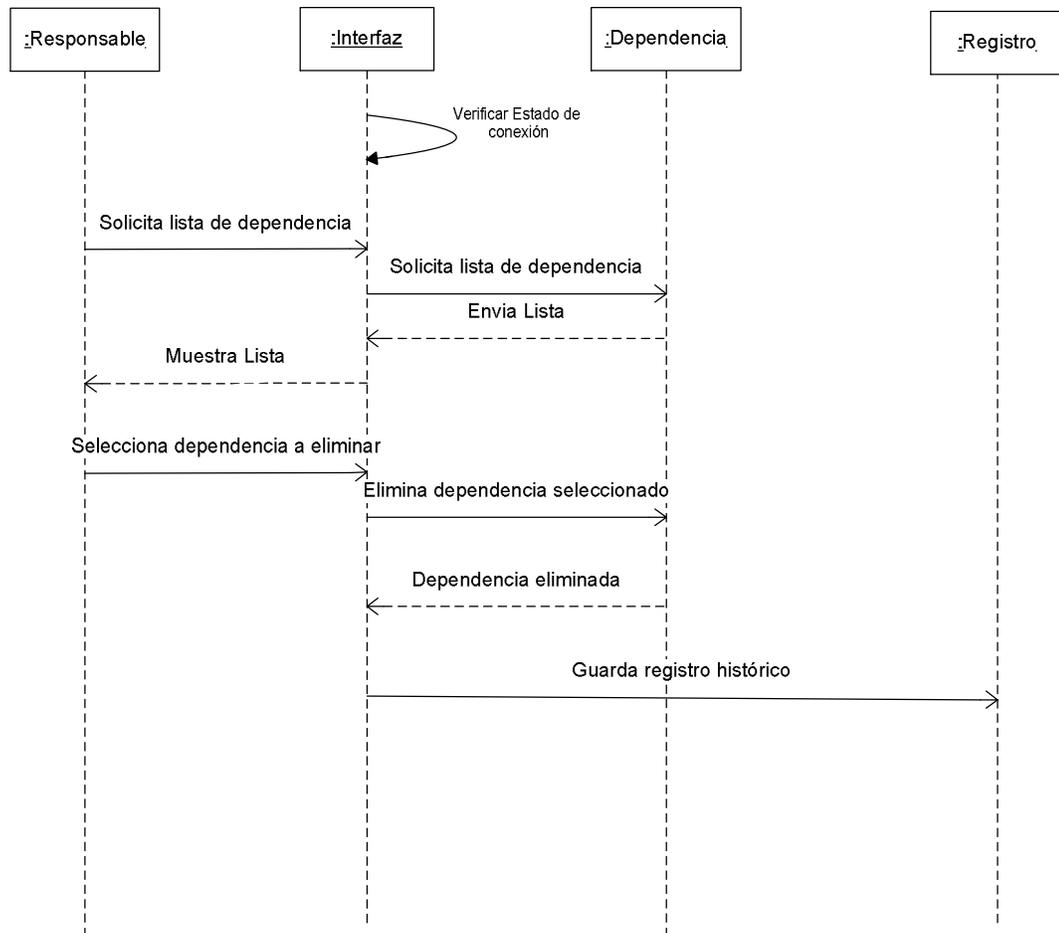
Para ver la lista de dependencias se les debe presentar la opción de ver todas las dependencias en una lista, se debe guardar un registro histórico de cualquier empleado que use esta opción en la base de datos

3.28 Editar Dependencia



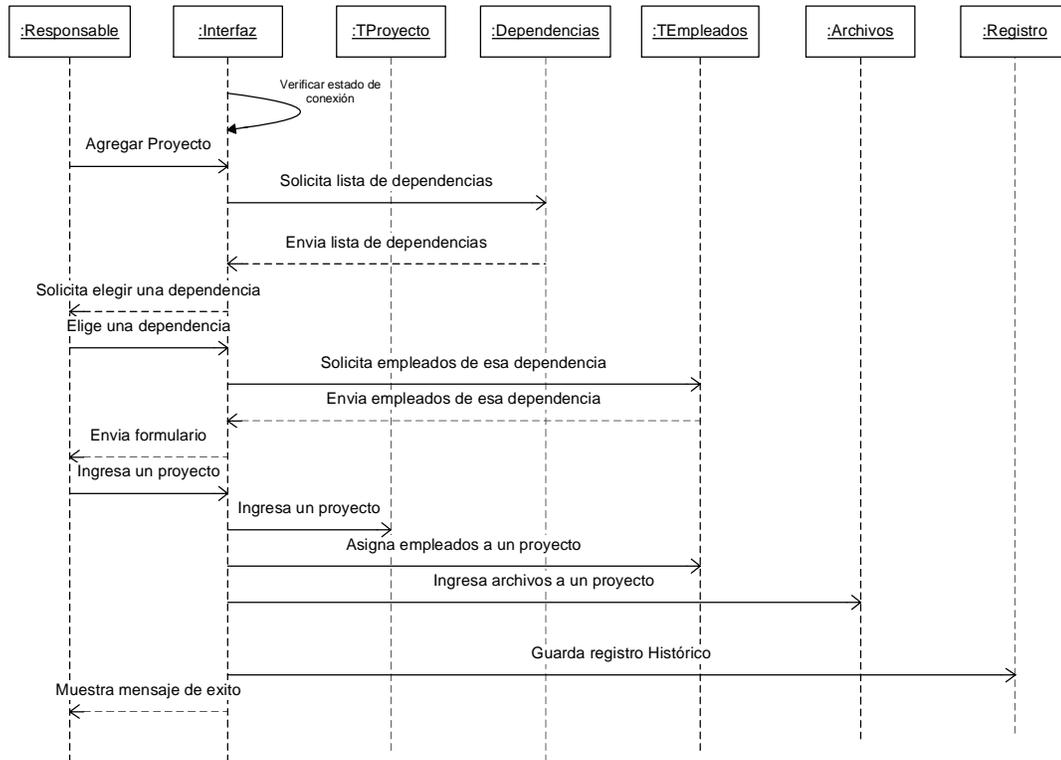
Para editar dependencias se debe mostrar una lista de todas las dependencias existentes y de esa manera, el usuario debe seleccionar una para editar, luego se le debe mostrar un formulario con los campos editables de ella, el cual tiene la información del perfil que se desea editar, se debe guardar un registro histórico de cualquier empleado que elimine a otro usuario.

3.29 Eliminar Dependencia



Para verificar la lista de dependencias, y seleccionar de entre ellas una o varias para eliminar, luego de seleccionadas se eliminan, se debe guardar un registro histórico de cuándo un usuario elimina alguna dependencia, guardando su fecha y hora.

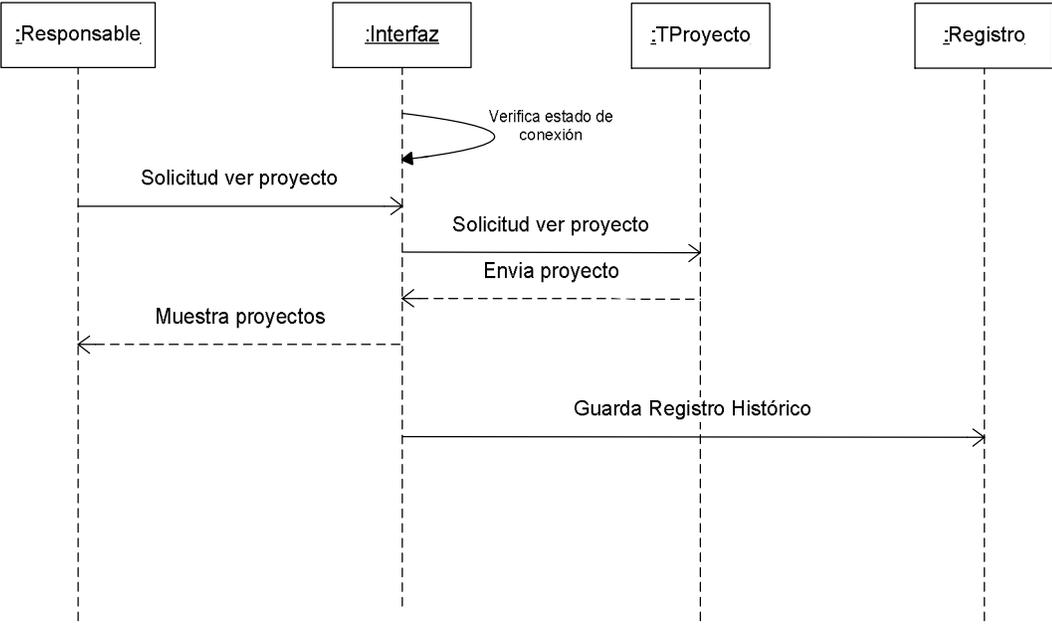
3.30 Agregar Proyectos



Para agregar un proyecto al sistema se entrega un formulario en el cual se solicita la información del proyecto y se especifica a qué dependencia estará asignada, así como cuales son los empleados que estarán asignados a este proyecto. También es muy importante que los empleados asignados a dicho proyecto sean sólo de la dependencia a la cual fue asignado el proyecto.

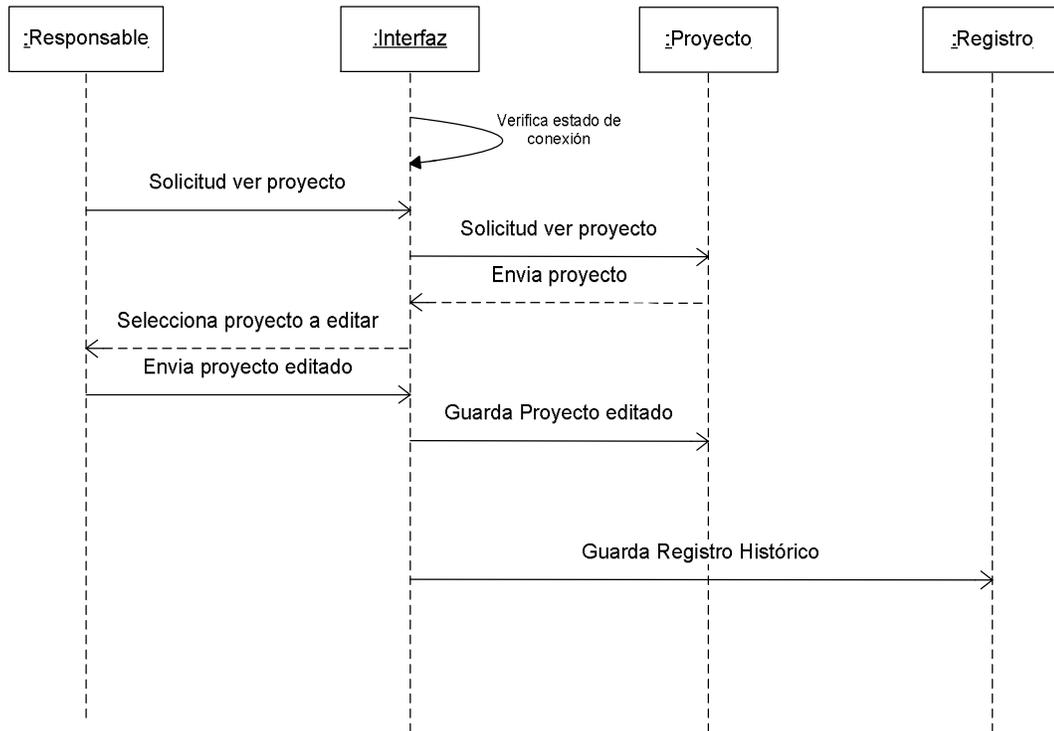
Cuando se crea un proyecto es importante que se pueda colocar información anexa a dicho proyecto y eso se hace por medio del envío de archivos, para lo cual el sistema debe soportar almacenar los archivos, que las personas consideren importante como información anexa, se debe guardar un registro histórico de cuándo un usuario crea un proyecto, guardando su fecha y hora.

3.31 Listar Proyectos



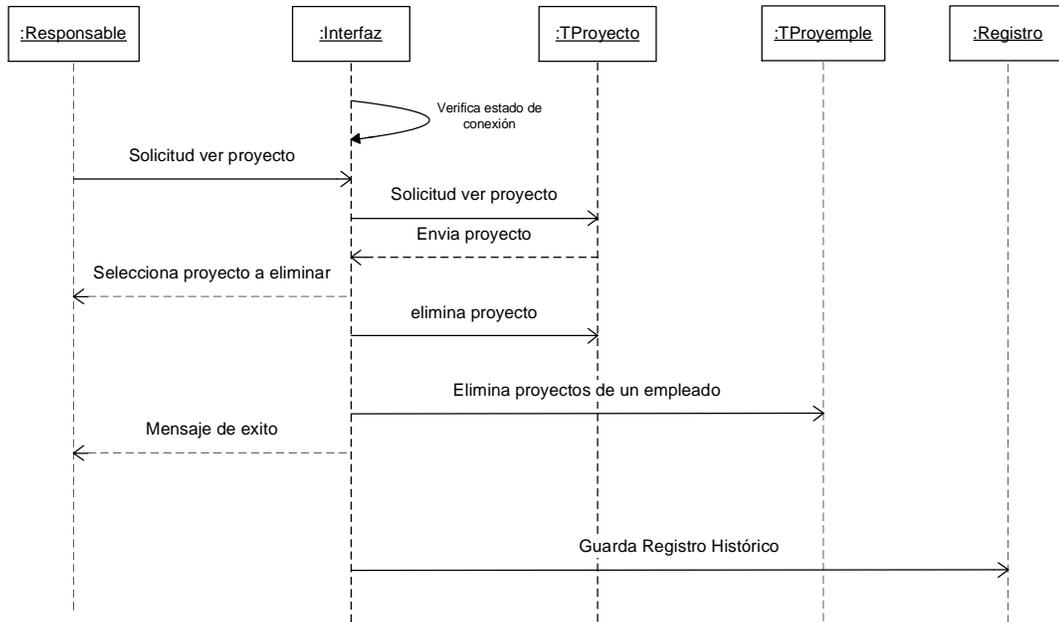
Para listar proyectos, se permite ver una lista completa de proyectos y luego el mismo usuario puede seleccionar cualquiera de dichos proyectos para ver más información sobre él en específico. Se debe guardar un registro histórico de cuándo un usuario lista proyectos guardando su fecha y hora, y el usuario que listó los proyectos.

3.32 Editar Proyecto



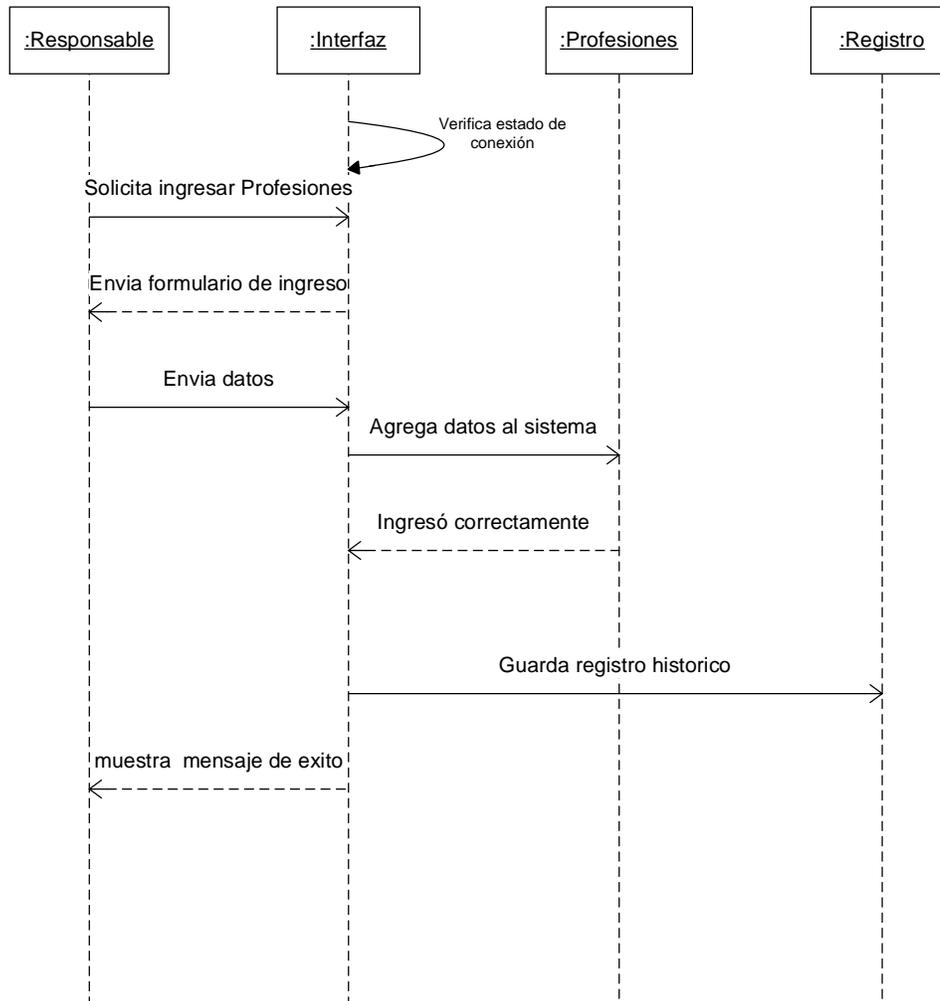
Para editar proyectos se debe mostrar una lista de todos los proyectos creados en el sistema, luego que él seleccione alguno de los proyectos, se debe mostrar toda la información de ese proyecto, con posibilidad de edición y luego se debe actualizar dicho registro se debe guardar un registro histórico de cualquier empleado que edite algún proyecto, con fecha hora y una descripción de lo que ese usuario hizo.

3.33 Eliminar Proyecto



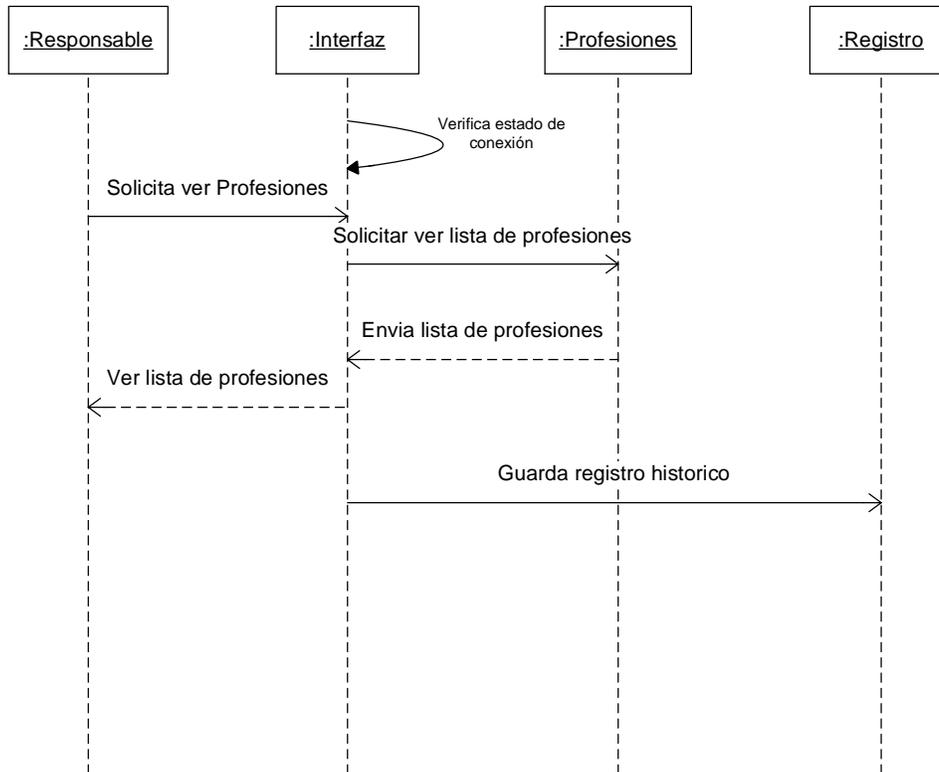
Para eliminar cualquier proyecto, se debe mostrar una lista de los proyectos actuales y luego el usuario selecciona qué proyecto va a eliminar, además luego de seleccionado, se debe eliminar también todo lo que en el sistema quede de dicho proyecto, desvincular los empleados, borrar archivos etc. Por último se debe guardar un registro histórico de cuándo un usuario elimina un proyecto, guardando su fecha y hora, y el usuario que ejecutó dicha acción.

3.34 Agregar Profesiones



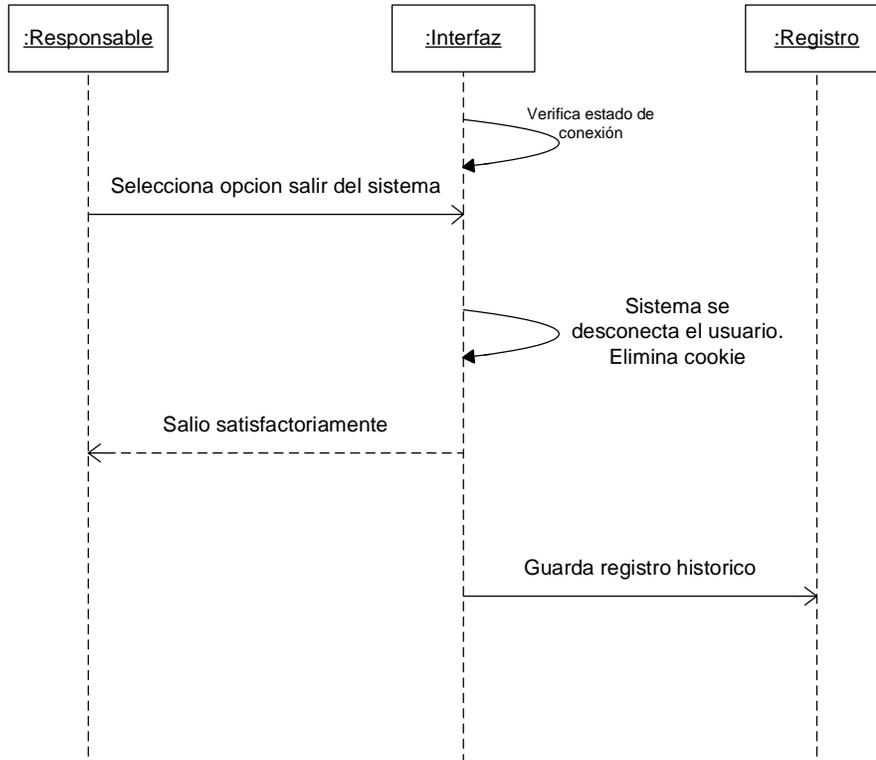
Para agregar profesiones se presentará la opción de un formulario de ingreso en el cual se ingresan los datos para una nueva profesión, se debe guardar un registro histórico de cuándo un usuario agrega una profesión guardando su fecha y hora, y el usuario que ejecutó dicha acción.

3.35 Ver Profesiones



Listar Profesiones es una consulta que se ejecuta sobre la base de datos y arroja una lista de las diferentes profesiones que existen en la base de datos y que están disponibles para ser usadas en la gestión de la base de datos. Se debe guardar un registro histórico de cuándo un usuario Ve la lista de cargos, guardando la fecha y hora, y el usuario que ejecutó dicha acción

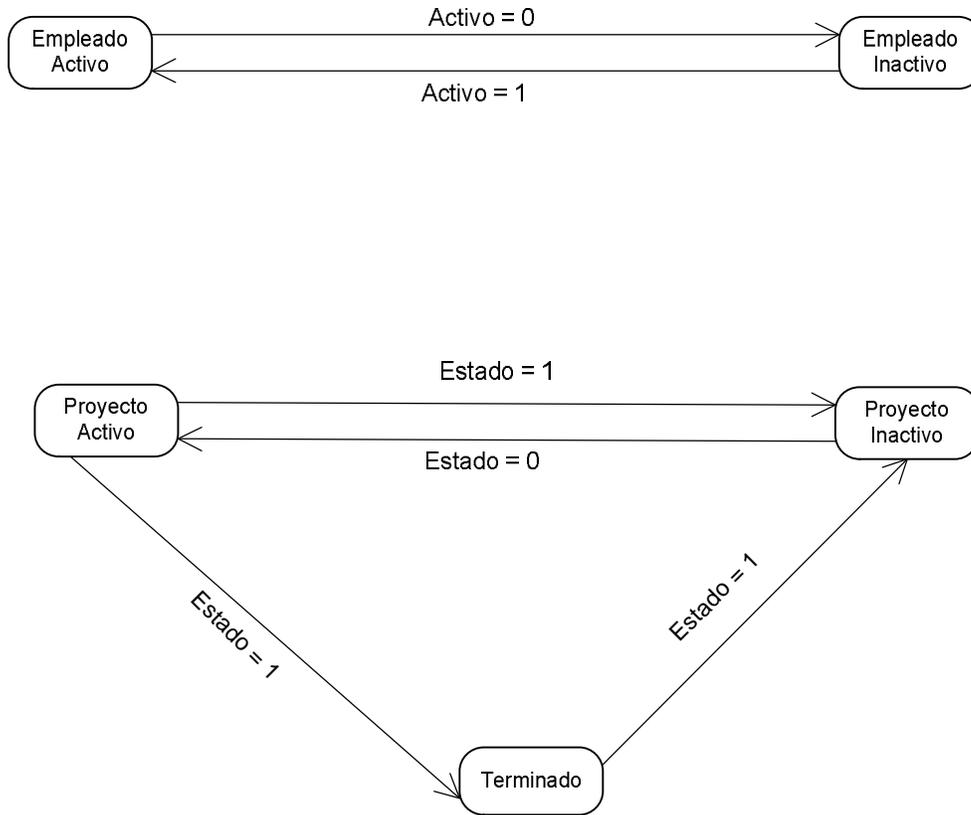
3.36 Salir



Salir del sistema es una opción que permite eliminar cualquier posibilidad de seguir ejecutando acciones en la base de datos con el usuario actual, se debe eliminar cualquier indicio de reconocimiento del usuario y solicitar nuevamente una conexión.

Se debe guardar un registro histórico de cuándo un usuario se desconecta del sistema.

4. DIAGRAMA DE ESTADOS



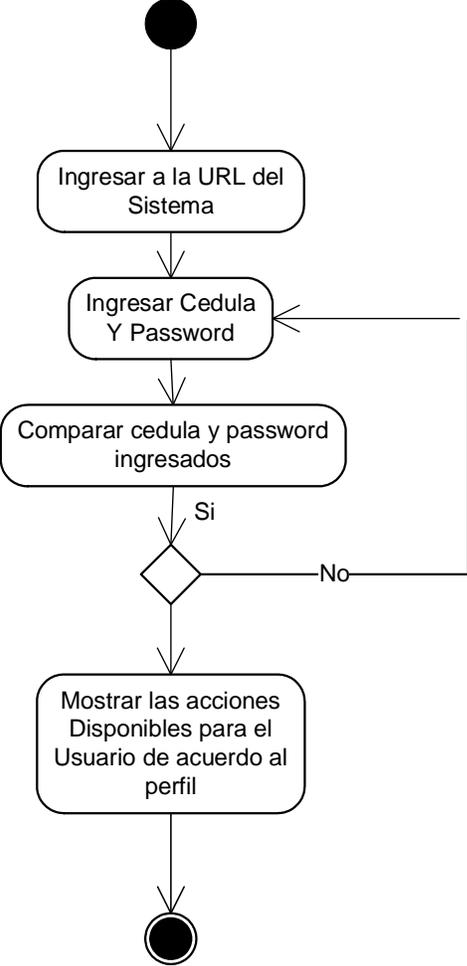
Mediante este diagrama se describe los posibles estados que puedan tener un empleado o proyecto

El empleado puede estar activo a inactivo dentro del sistema dependiendo de los horarios que cumpla ya que estará activo solo en las horas que le sean asignadas por el súper usuario.

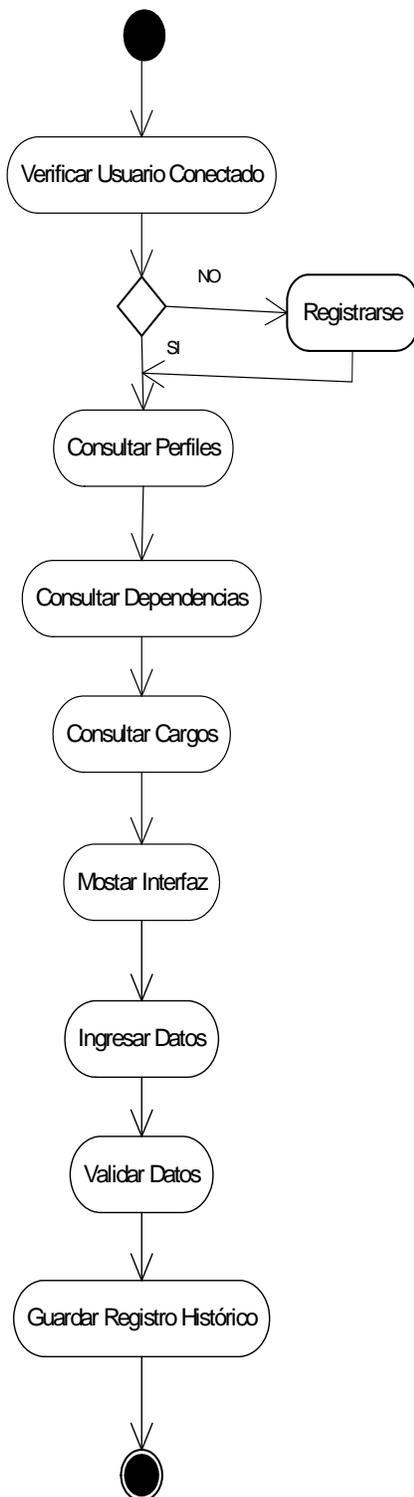
Un proyecto tiene tres etapas según su ejecución y tiempo de desarrollo estas etapas pueden ser activo cuando el proyecto se encuentra en vigencia. Inactivo cuando no se ha iniciado, se ha detenido por algún inconveniente, o se ha finalizado. Finalizado cuando se ha cumplido con el desarrollo total de dicho proyecto

5. DIAGRAMAS DE ACTIVIDAD

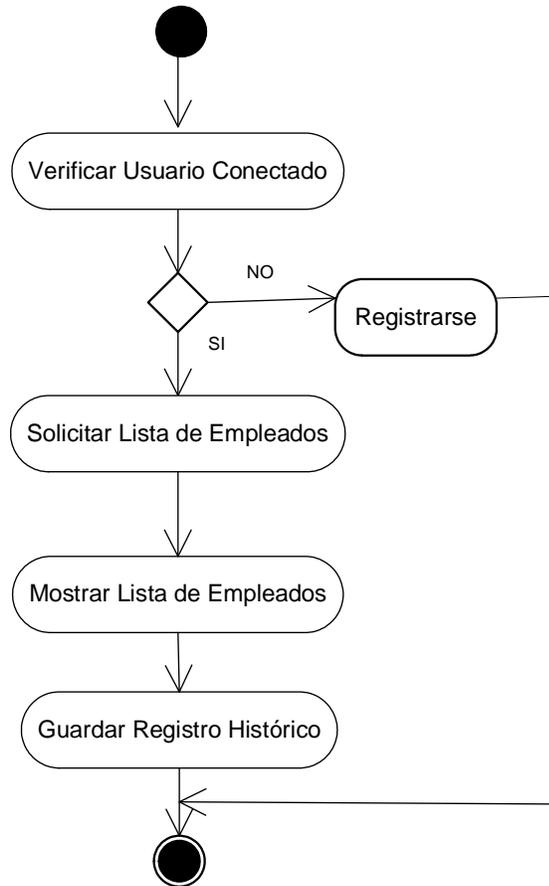
5.1 Ingreso al sistema



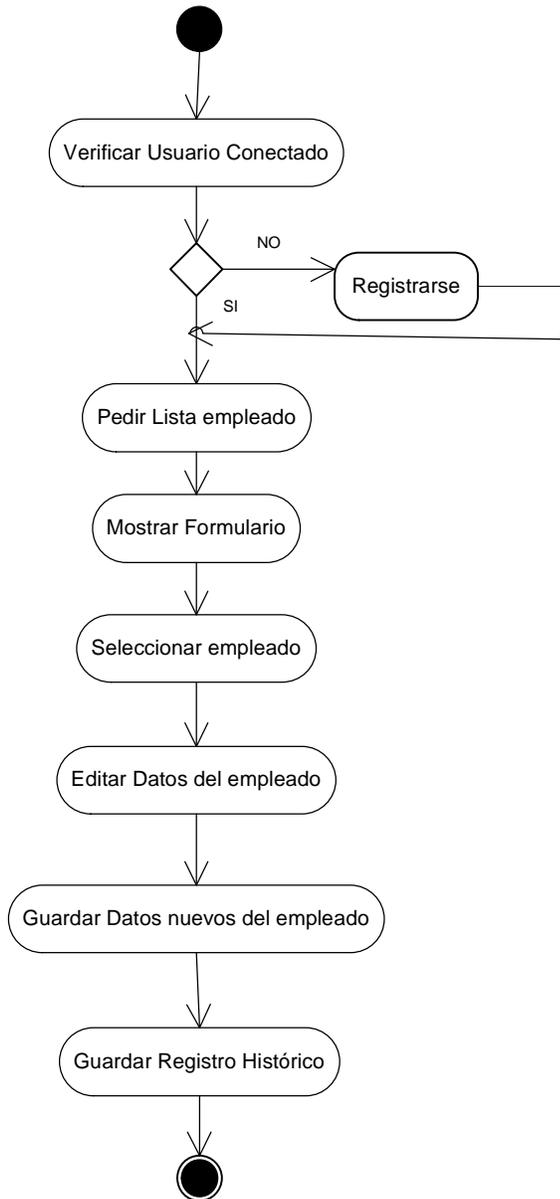
5.2 Agregar Empleados



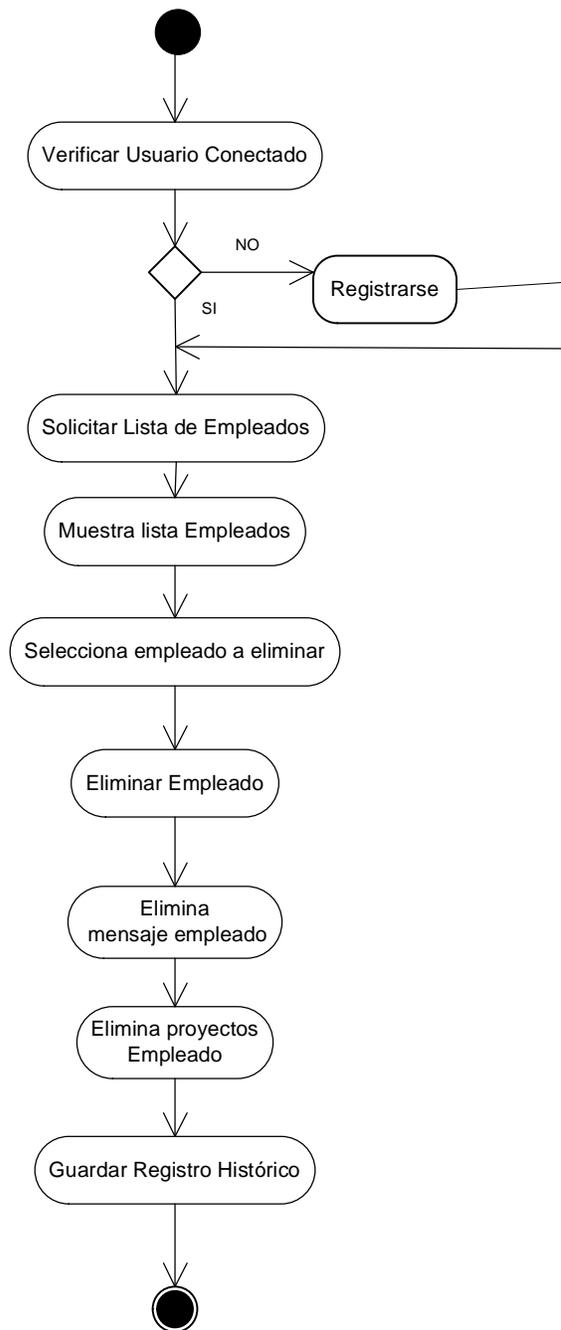
5.3 Ver Empleados



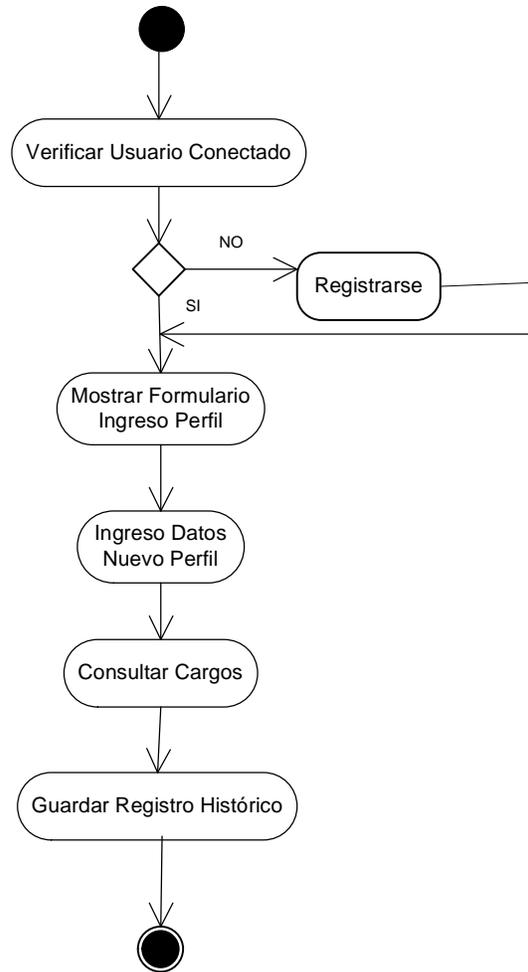
5.4 Editar Empleados



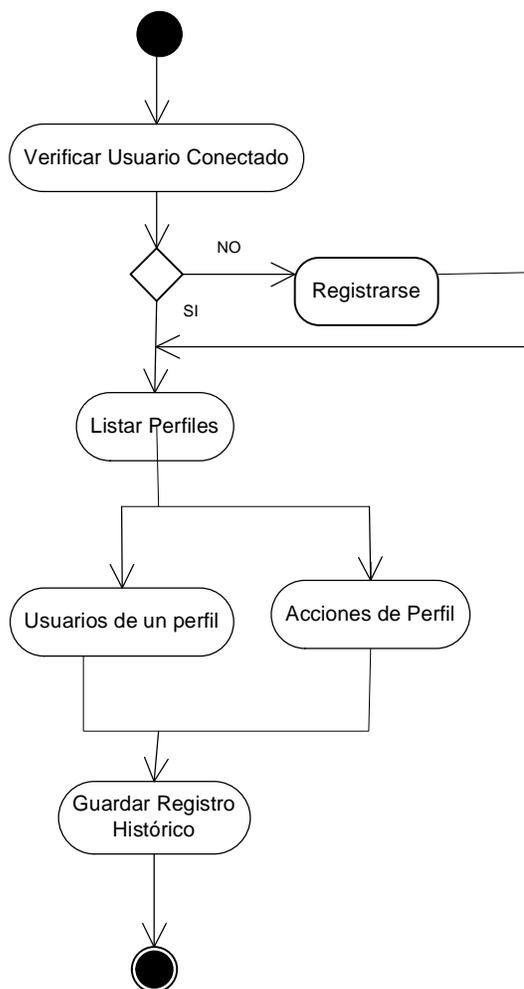
5.5 Eliminar Empleado



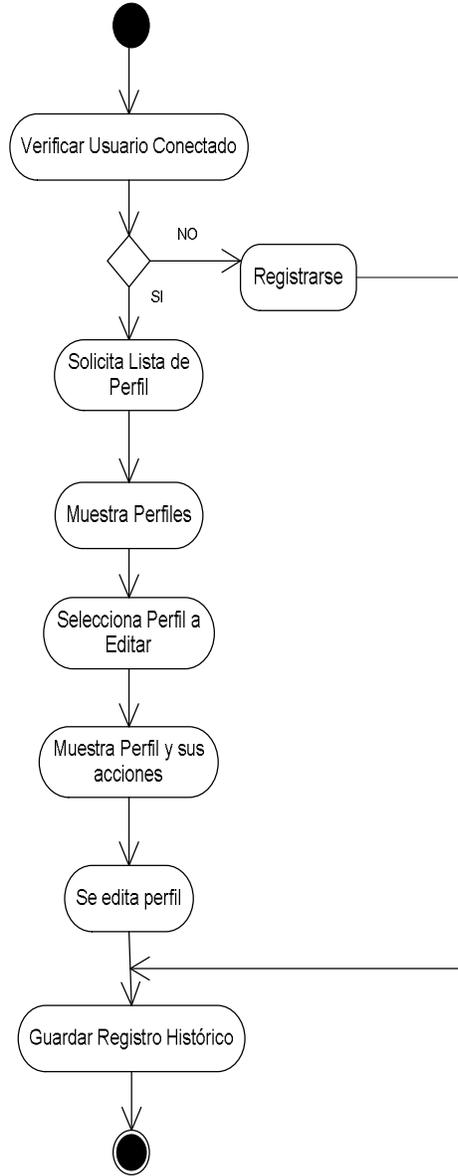
5.6 Crear Perfil



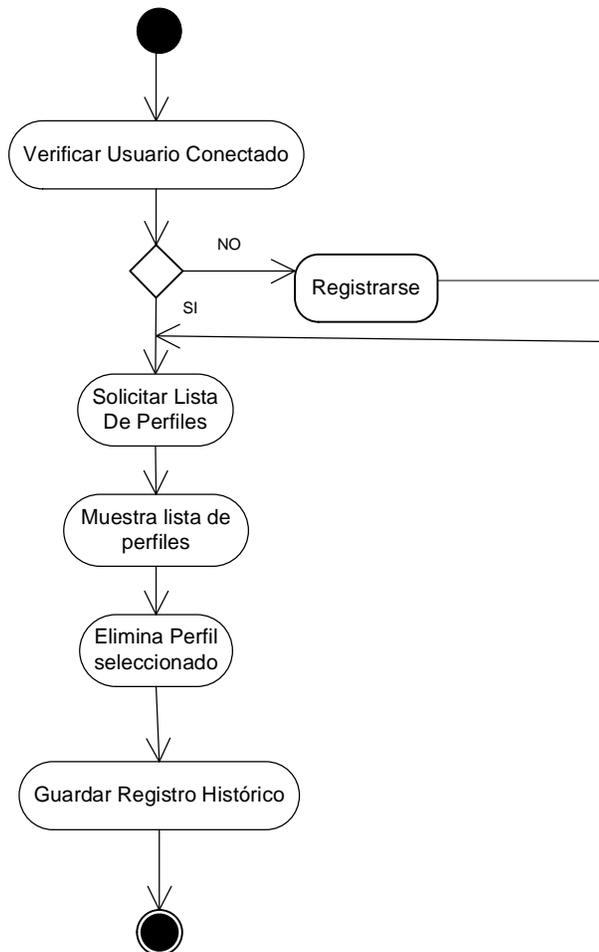
5.7 Ver Perfil



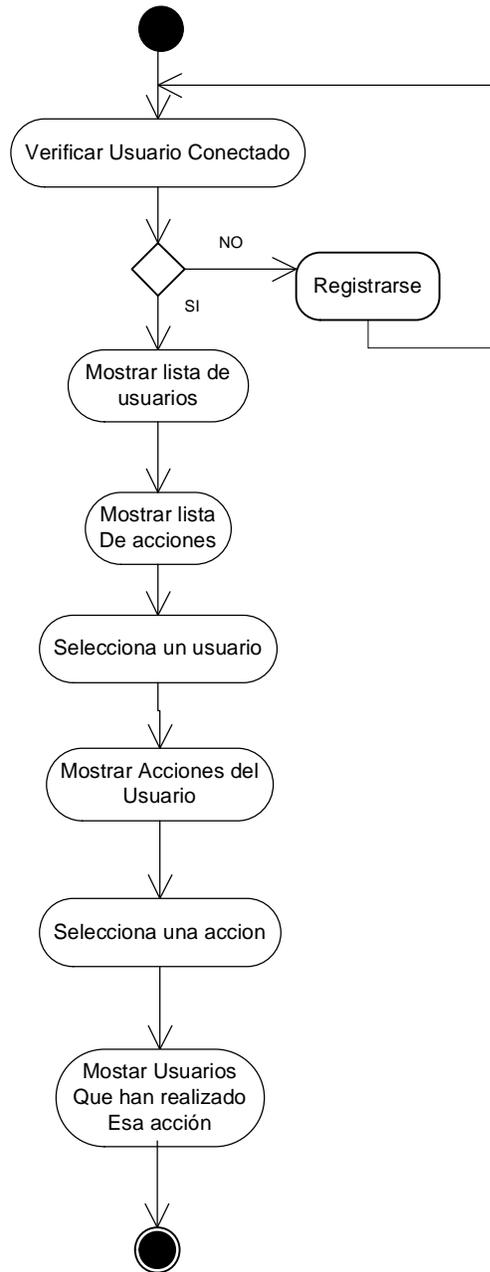
5.8 Editar Perfiles



5.9 Eliminar Perfiles



5.10 Histórico



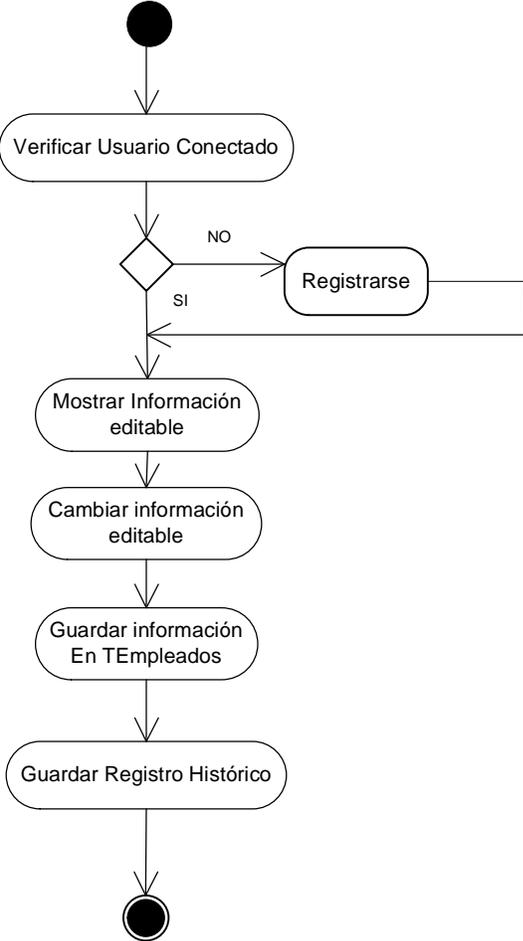
5. 11. Histórico Por usuario



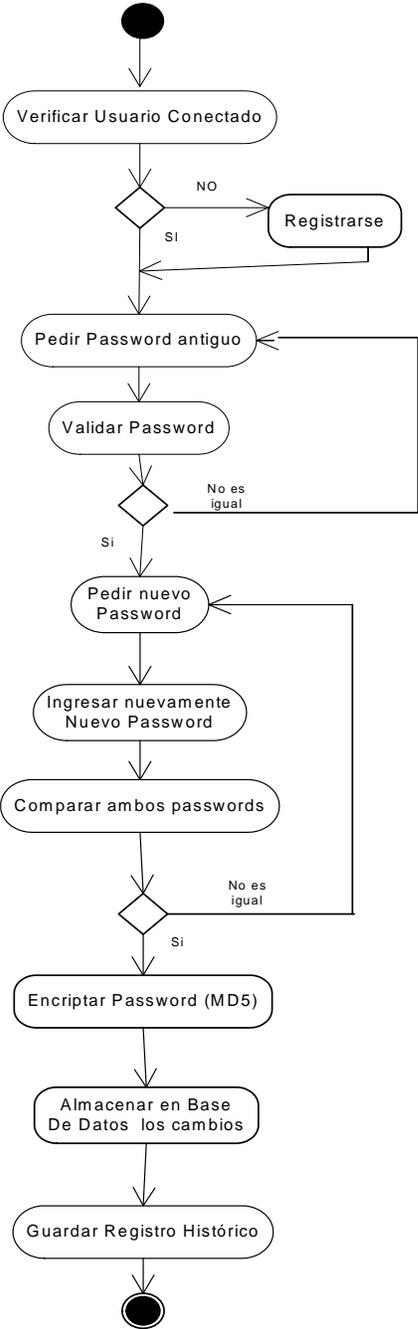
5.12. Histórico – Usuarios por acción



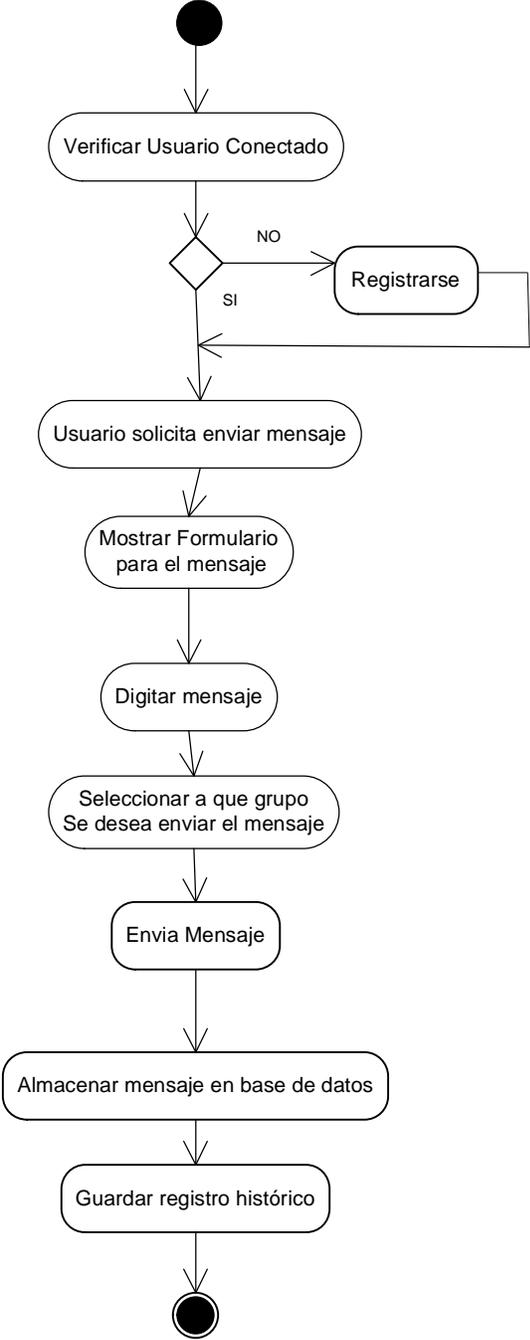
5. 13. Actualizar Información



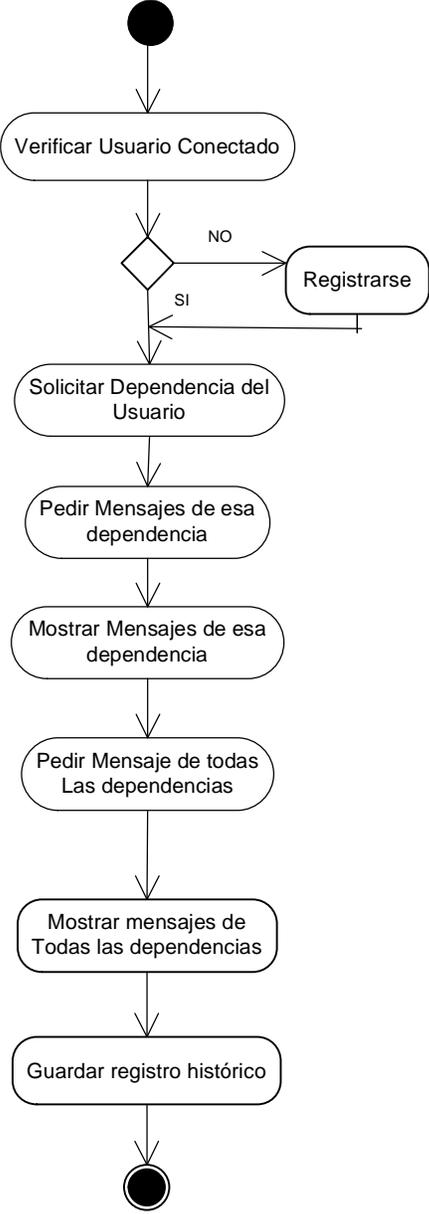
5.14. Cambiar Password



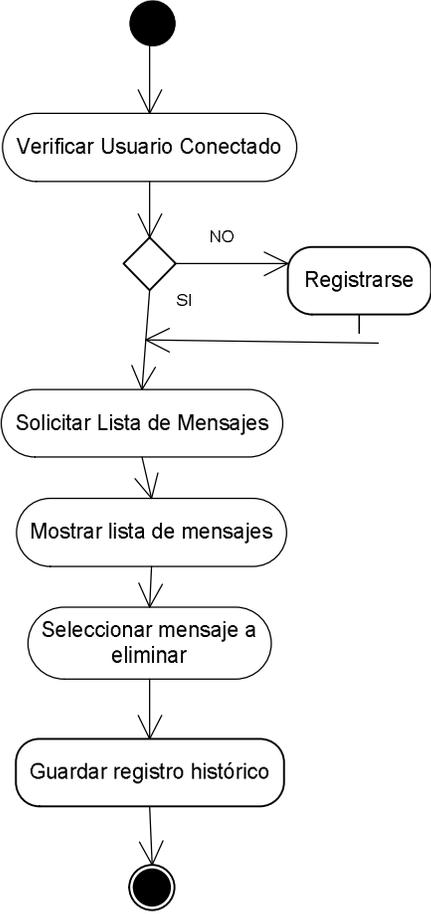
5.15. Dejar Mensaje



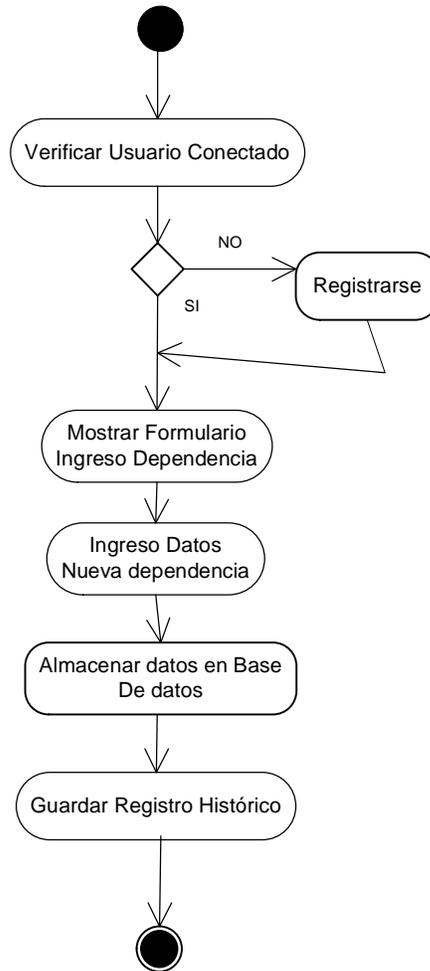
5. 16. Ver Mensaje



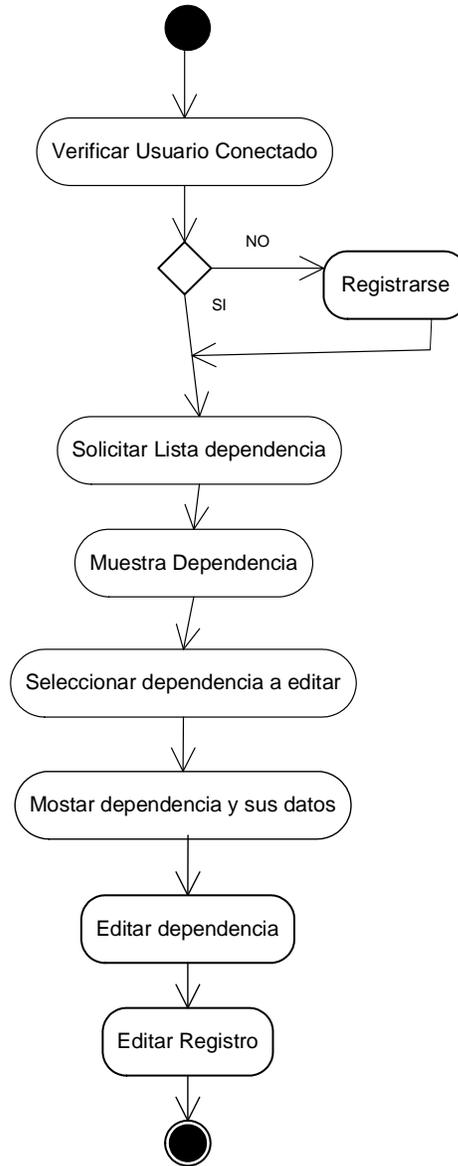
5.17. Eliminar Mensaje



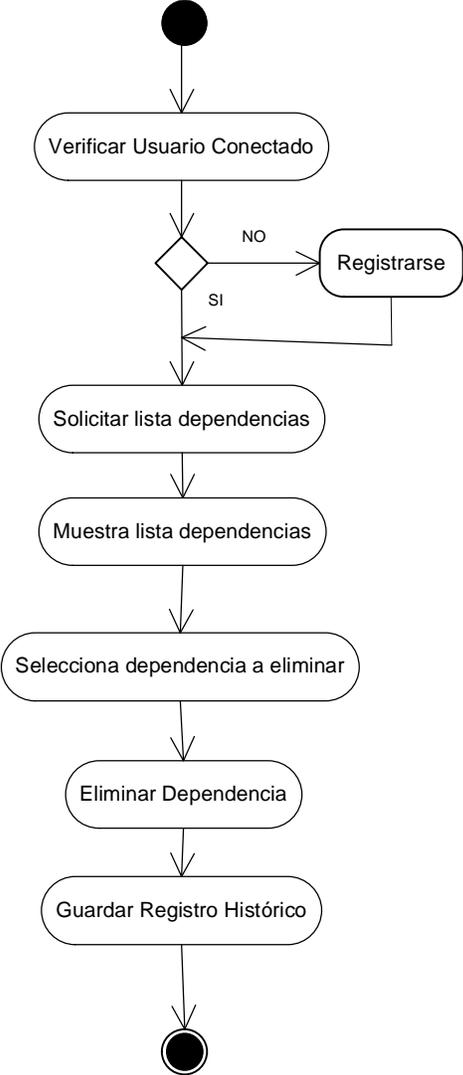
5.18 Adicionar Dependencia



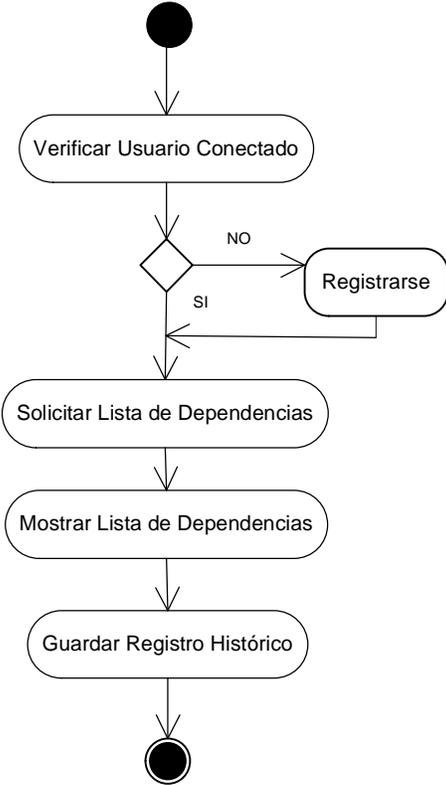
5.19 Editar Dependencia



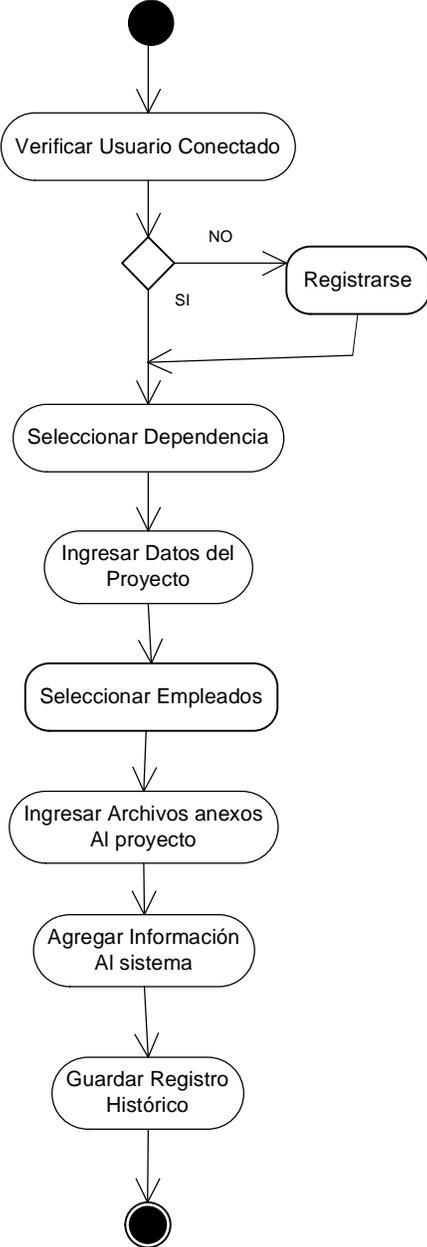
5.20 Eliminar Dependencia



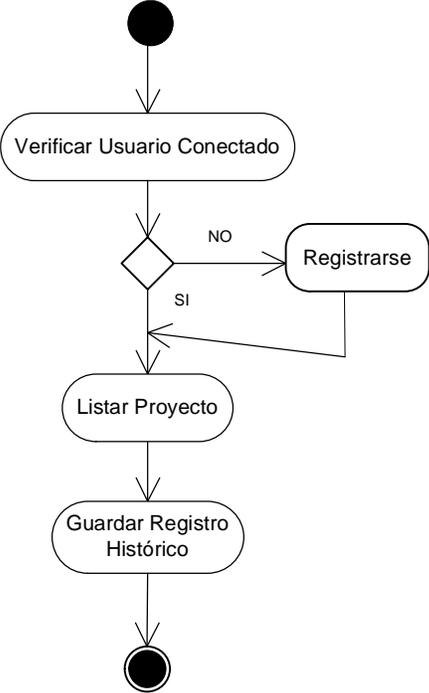
5.21 Listar Dependencias



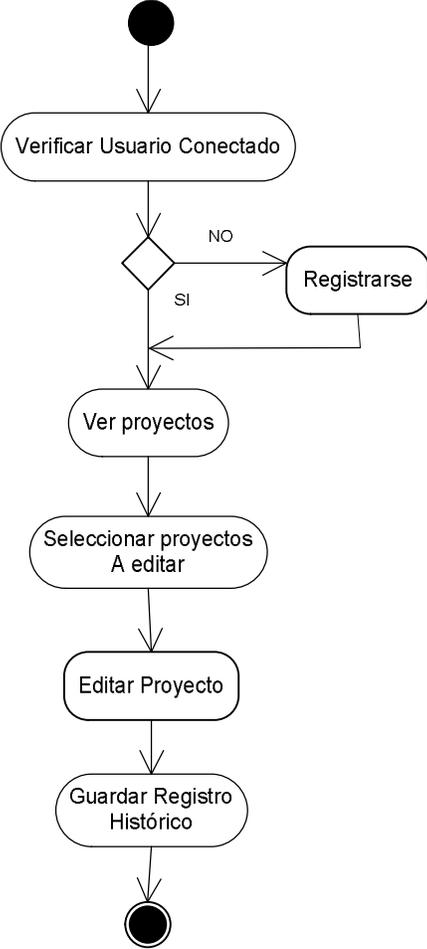
5.22 Agregar Proyecto



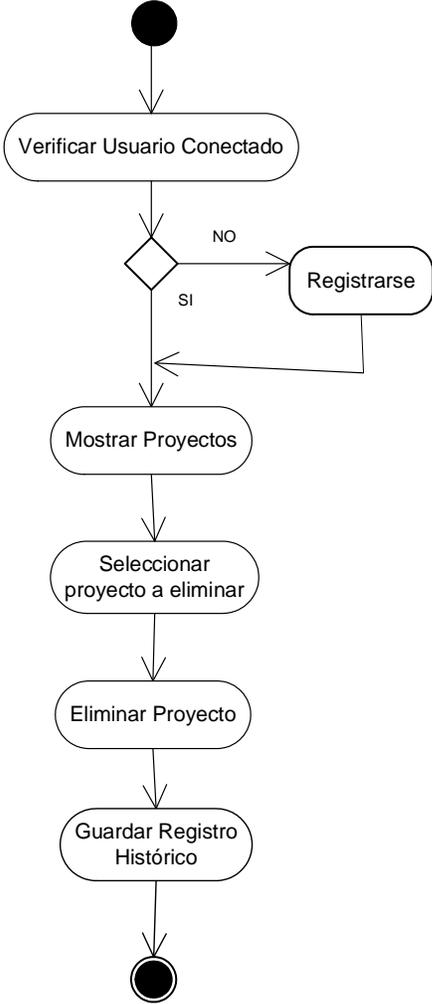
5.23 Listar Proyecto



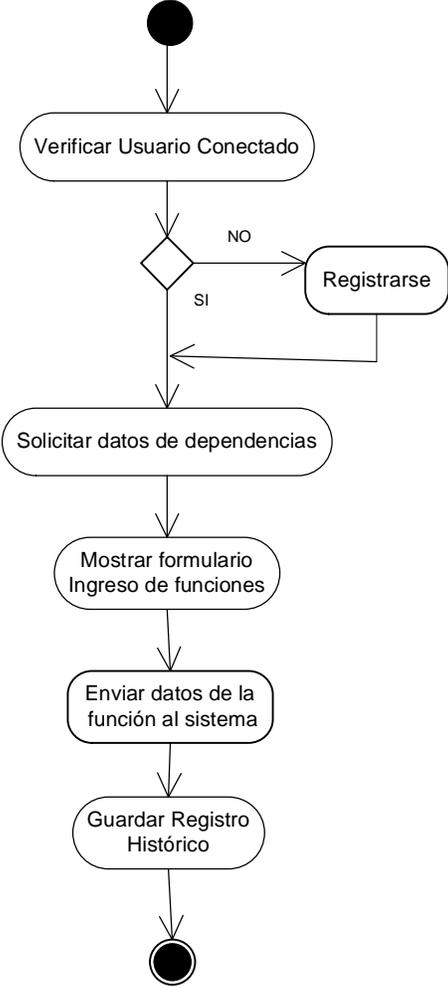
5.24 Editar Proyecto



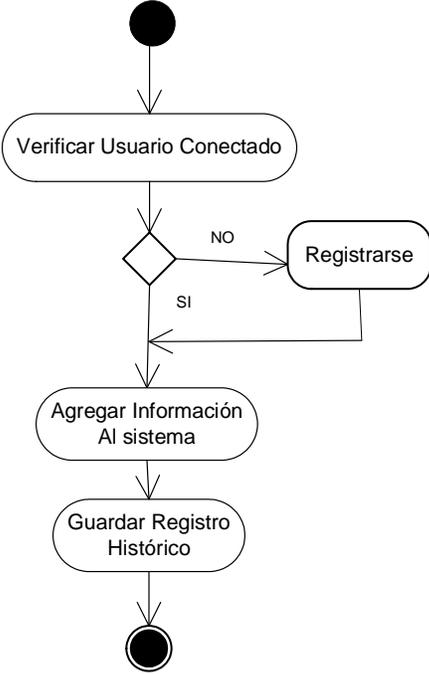
5.25 Eliminar Proyecto



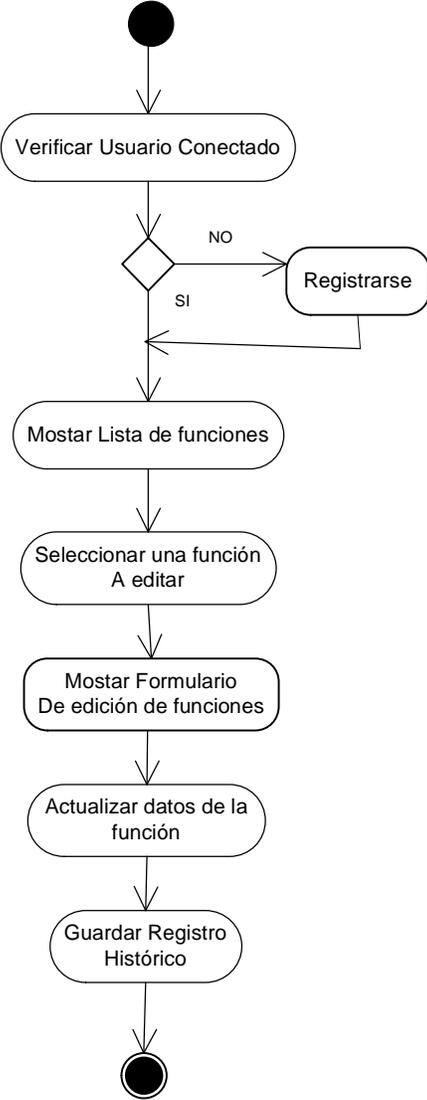
5.26. Agregar Funciones



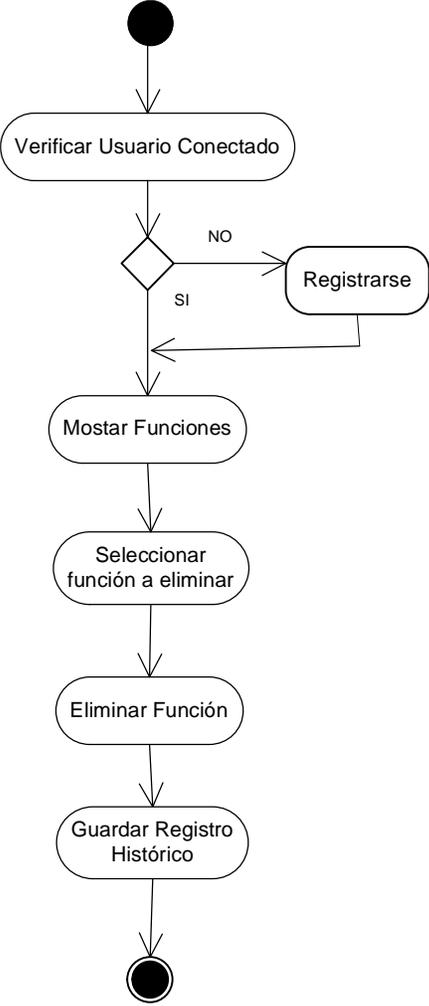
5.27. Ver funciones



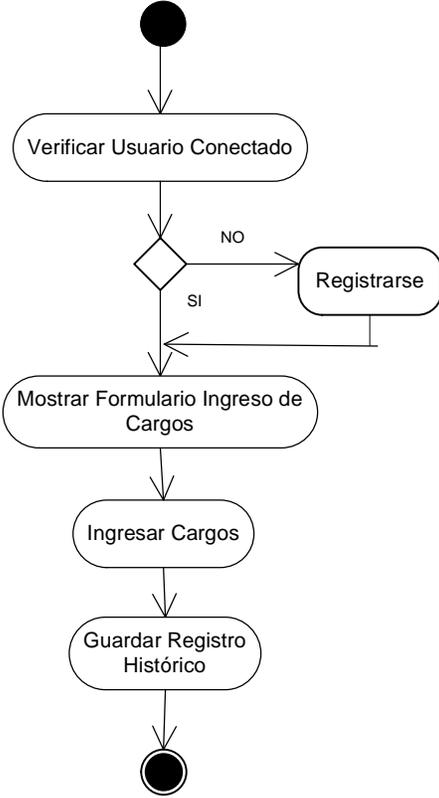
5.28 Editar Funciones



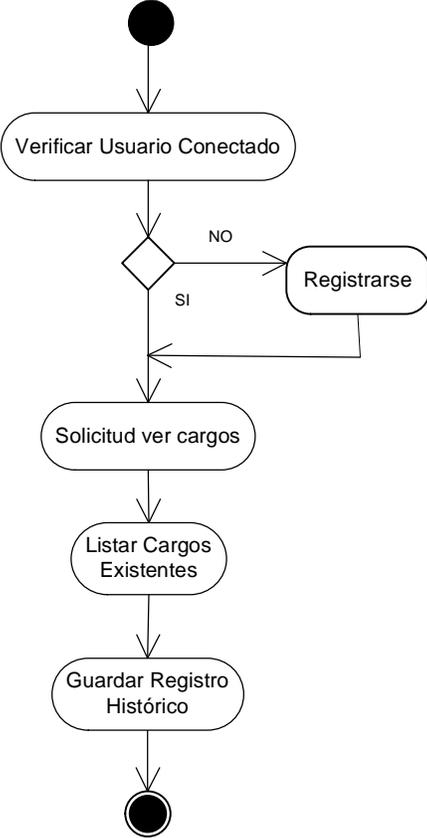
5.29 Eliminar Función



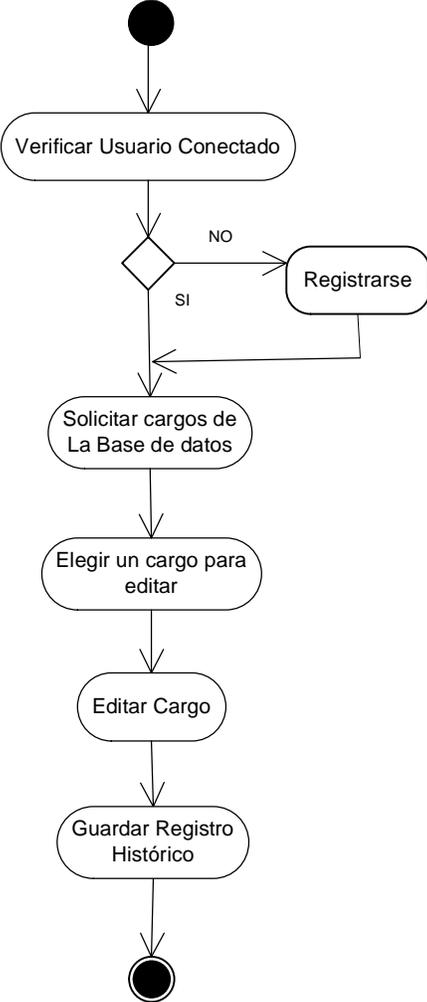
5.30. Agregar Cargos



5.31. Ver Cargos



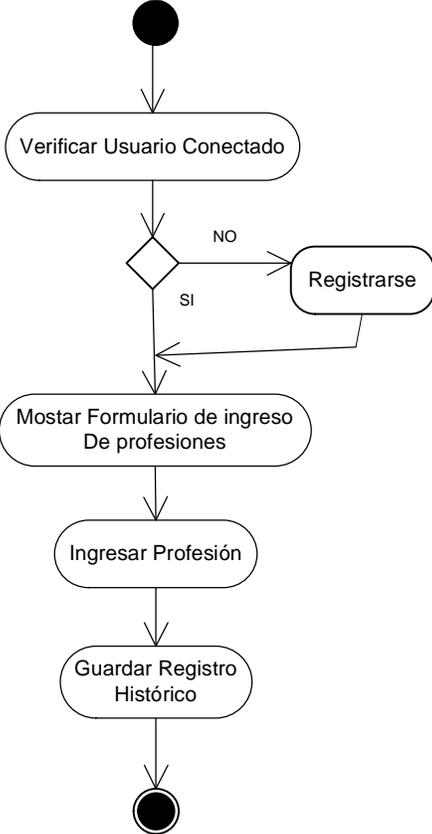
5.32. Editar Cargos



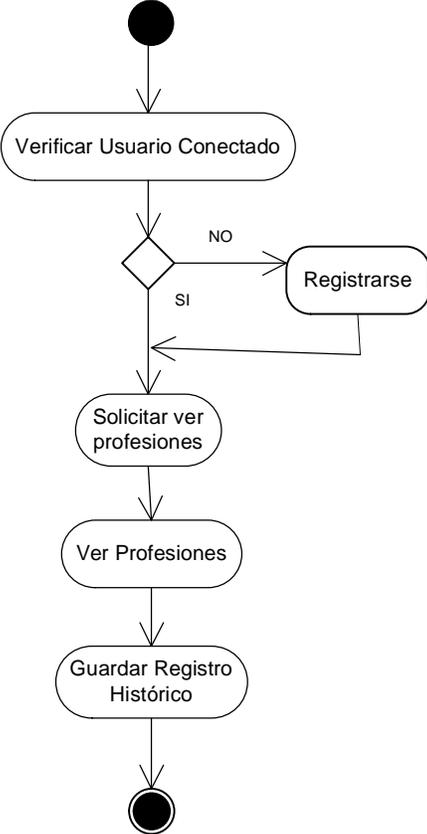
5.33 Eliminar Cargos



5.34. Agregar Profesiones



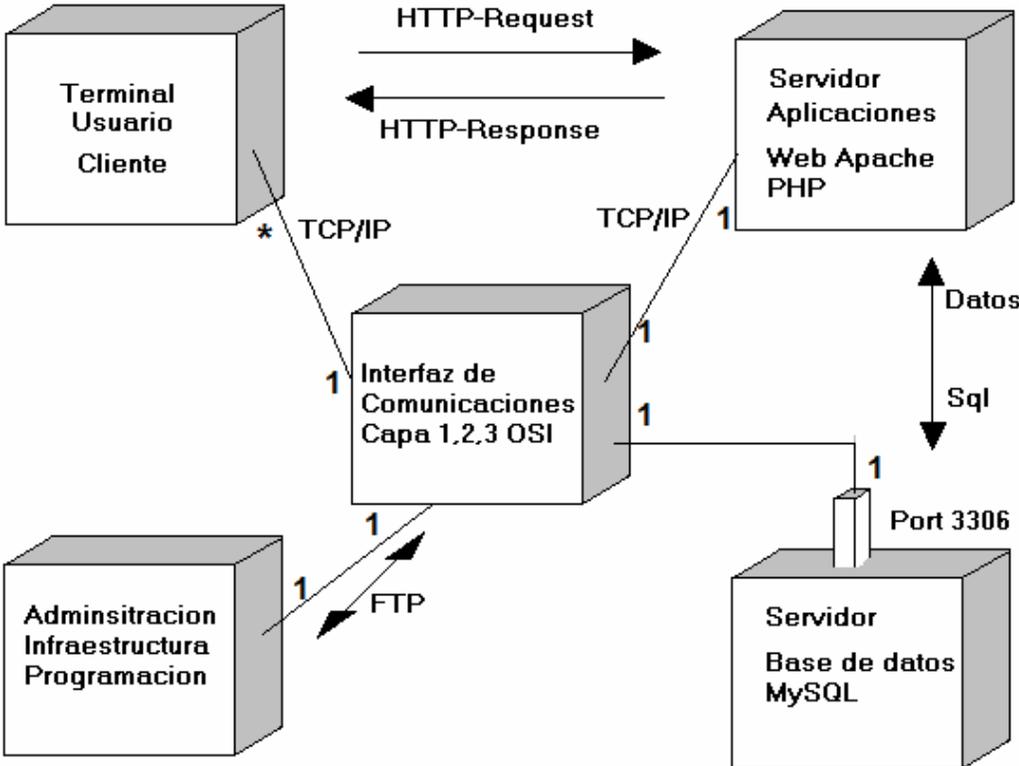
5.35. Ver Profesiones



5.36. Salir



6. DIAGRAMA DE DESPLIEGUE



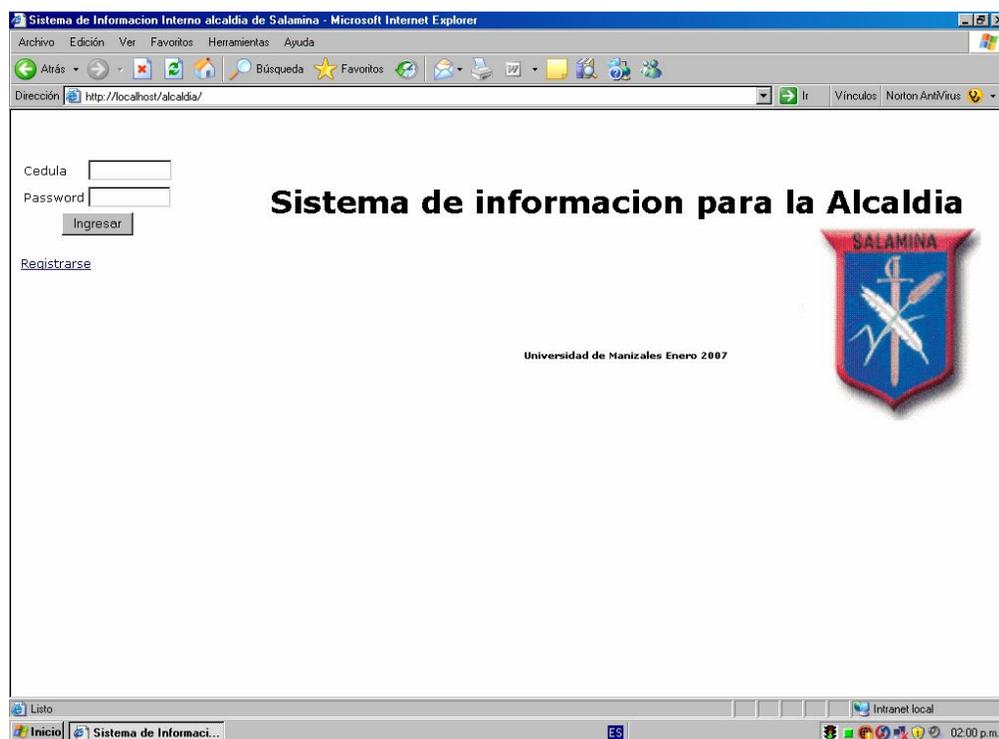
ANEXO C

MANUAL DEL USUARIO

El sistema de información para la Alcaldía de Salamina es mediante la ubicación del recurso URL (dirección Web) por medio de un navegador de Internet cualquiera que soporte HTML 4.

Se debe ubicar la dirección <http://servidor/alcaldia> en la cual **servidor** es el dominio en el cual se encuentra instalado el servidor Web con capacidades de comunicación con la base de datos MySQL y los archivos de la aplicación.

Al entrar a esta dirección Web se tiene la siguiente interfaz de usuario:



Se tienen tres perfiles predeterminados y cuando el usuario ingresa al sistema con su **cedula** y **password** correctamente se presenta una lista de enlaces a las acciones permitidas a ese usuario en de acuerdo a su perfil.

Existen de manera determinada tres perfiles: SuperUsuario, **Responsable**, **Cliente**.

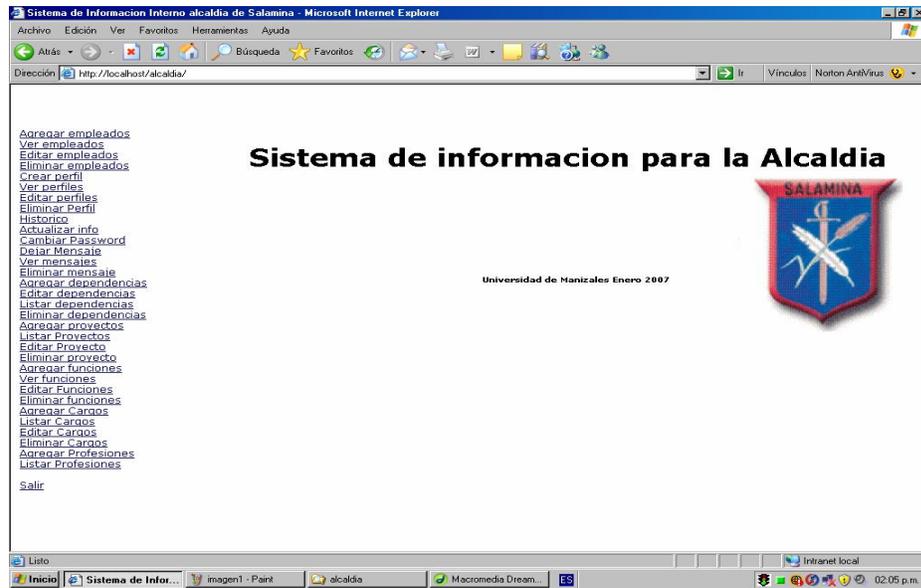
Cuando un usuario “entra” al sistema de acuerdo a su perfil se muestran las **acciones permitidas** para ese usuario de acuerdo a su perfil en el lado derecho de la pantalla.

El conjunto de acciones disponibles en el sistema son

- 001 Agregar empleados
- 002 Ver empleados
- 003 Editar empleados
- 004 Eliminar empleados

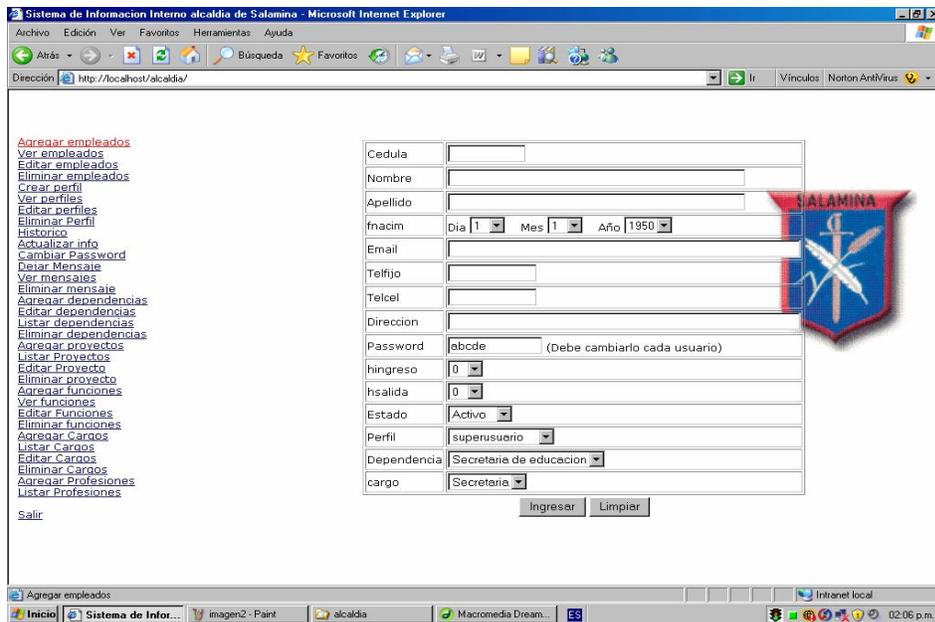
- 005 Crear perfil
- 006 Ver perfiles
- 007 Editar perfiles
- 008 Eliminar Perfil
- 009 Historico
- 010 Actualizar info
- 011 Cambiar password
- 012 Dejar mensaje
- 013 editar mensaje
- 014 eliminar mensaje
- 015 Agregar dependencias
- 016 Editar dependencias
- 017 listar dependencias
 - listar funciones de una dependencia
 - listar proyectos de una dependencia
 - listar empleados de una dependencia
- 018 Eliminar dependencias
- 019 agregar proyectos
- 020 listar proyectos
- 021 editar proyecto
- 022 Eliminar proyecto
- 023 Agregar funciones
- 024 Ver funciones
- 025 editar funciones
- 026 Eliminar funciones
- 027 agregar cargos
- 028 listar cargos
- 029 Editar cargos
- 030 Eliminar cargos
- 031 Agregar profesiones
- 032 listar profesiones
- 033 Salir

Luego de ingresar al sistema se presenta una lista de opciones en el lado derecho de la siguiente manera:



Ingreso de datos

Para el ingreso de datos al sistema agregar empleados, Crear perfil ,Dejar mensaje, Agregar dependencias , Agregar proyectos , Agregar funciones, Agregar cargos, Agregar profesiones. Se utilizan formularios de ingreso de datos como el siguiente:



ANEXO D

MANUAL DE INSTALACIÓN

Pasos para instalar la aplicación en un servidor

- 1 Instalación del servidor.
- 2 Ejecución de consultas SQL (Creación de la base de datos)
- 3 Instalación de los programas
- 4 Probar la aplicación

1 Instalación del Servidor: Para instalar la aplicación de gestión de la información de la Alcaldía de Salamina en un servidor, se necesita instalar el conjunto de herramientas

Apache, PHP, MySql.

Se pueden instalar aparte o se puede utilizar una herramienta que los instale de manera conjunta, para ello se pueden utilizar una herramienta como la utilizada para la creación del proyecto

VERTRIGO: Aplicación de software que empaqueta las herramientas PHP, MySql, Apache, SqlLite, PhpMyadmin, en un solo instalador, es un proyecto creado y listado dentro del repositorio de proyectos: SourceForge es cual puede ser descargado desde <http://www.sf.net> o directamente desde el sitio Web del proyecto:

<http://vertrigo.sf.net> desde la cual se puede descargar el instalador para diferentes sistemas operativos.

El cual al ser ejecutado muestra la siguiente pantalla



El instalador de la aplicación está en el Vertrigo Server 1.17 el cual está en el CD anexo a este documento.

Se debe seguir el proceso de instalación de manera completa y el servidor queda instalado en la maquina elegida para tal efecto.

Aunque Cualquier otra herramienta puede ser utilizada, siempre y cuando se instale

PHP: Como herramienta de Programación

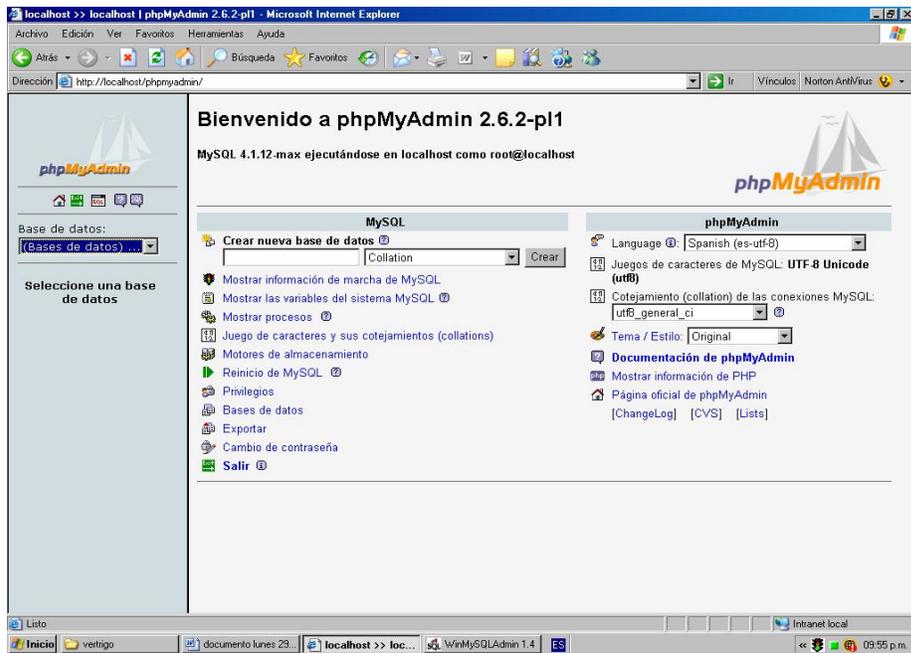
Apache: Como servidor Web

MySql Como base de datos

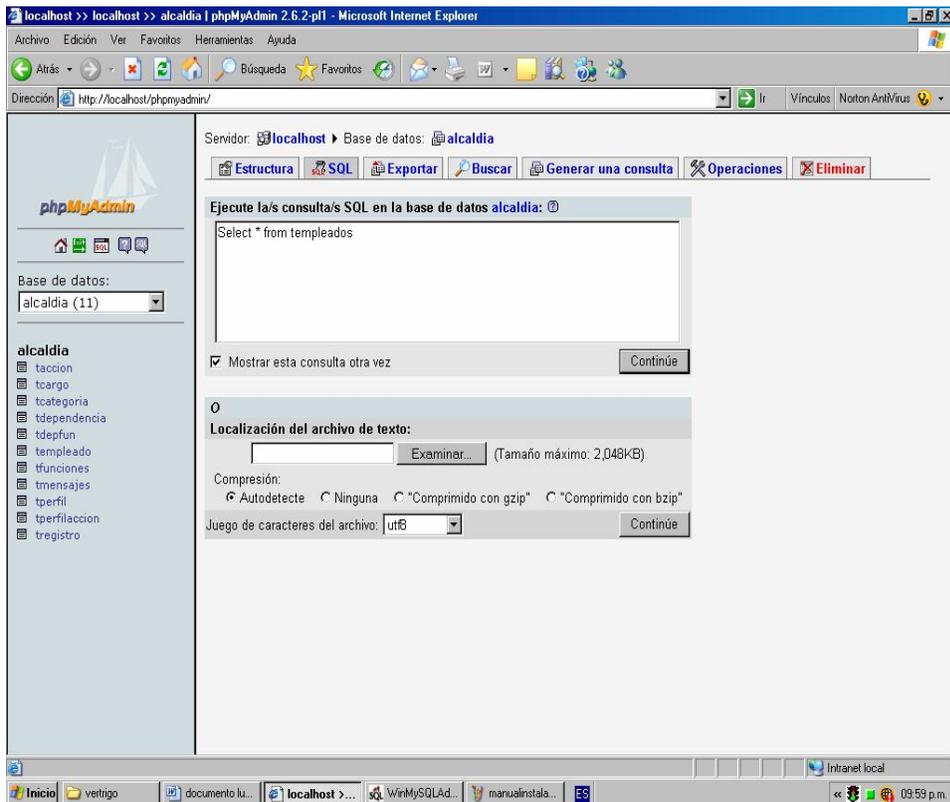
En nuestro caso también queda instalado el phpmyadmin el cual es una aplicación desarrollada en PHP que gestiona de manera sencilla bases de datos en MySql, en nuestro caso nos ayudará a gestionar la base de datos **alcaldía**

2 Ejecución de consultas SQL (Creación de la base de datos)

Se Utiliza la aplicación PhpMyadmin accedendo el url <http://127.0.0.1/phpmyadmin>



Y desde la opción de SQL:



Se ingresa el archivo de texto (**bdalcaldia.txt**) en el cual esta la consulta SQL para la creación de la base de datos, donde se crean las tablas y se adicionan los registros utilizados en la fase de desarrollo.

3. Instalación de los programas.

Se debe crear una carpeta con el nombre **alcaldía** en la ruta **c:\program files\Vertrigo Serv\Apache\htdocs**

Los diferentes programas (Códigos Scripts en PHP) deben ser ubicados en la ruta **c:\program files\Vertrigo Serv\Apache\htdocs\alcaldia**

De esa manera para invocar la aplicación solo se debe ubicar la URL del navegador en

<http://127.0.0.1/alcaldia>

Previamente se debe verificar que en los códigos (.php), la instrucción de conexión a la base de datos **mysql_connect**, se tenga el login y el password de la base de datos de MySql en los parámetros 2 y 3

De la siguiente manera:

mysql_connect("\$servidor:\$puerto","angela","xyzw")

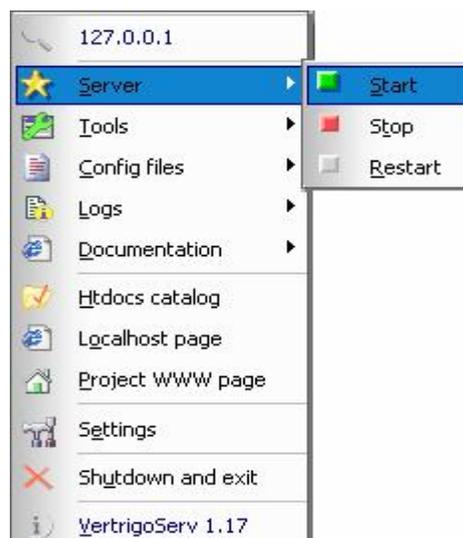
En este caso se tiene

login: angela

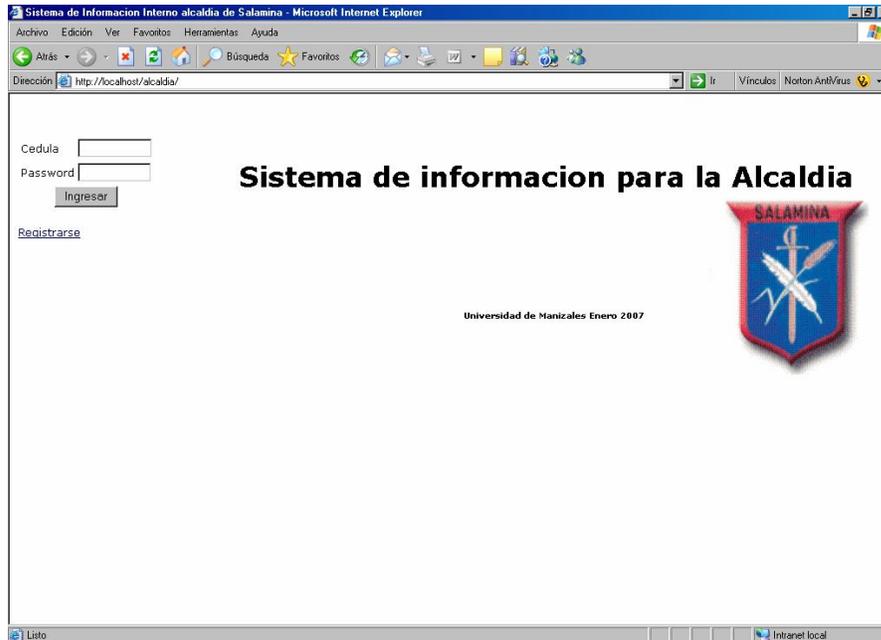
Password :xyzw

4. Probar la aplicación

Se debe iniciar (start) el servidor apache y el motor de base de datos, eso se hace desde la siguiente opción



Ahora se debe verificar que el servidor esta funcionando y la aplicación perfectamente instalada <http://127.0.0.1/alcaldia>. A lo cual el navegador debe responder con la siguiente interfaz



La cual es una pantalla que invita a ingresar al sistema a cualquiera de los actores