

1.1 Introducción

Actualmente la tecnología está evolucionando a pasos agigantados gracias a las computadoras, pues tienen gran capacidad de memoria y el acceso a los datos e información se realiza de manera sencilla y rápida. Logrando que las entidades reemplacen el procesamiento manual de la información por el procesamiento automático, utilizando Sistemas de Información de acuerdo a sus requerimientos.

Debido a la velocidad de las comunicaciones empleando redes de ordenadores (Internet o Intranet) y a la agilidad de algunos de los procesos, se consigue acelerar importantemente los procesos, logrando información oportuna para la toma de decisiones.

Gracias a la fácil accesibilidad al World Wide Web, a través de Internet, podemos conocer información, de diferentes tipos de organismos, acerca de sus actividades y establecer comunicación con las personas interesadas.

El propósito del presente trabajo es analizar y diseñar e implementar un Sistema Web de Gestión Académica, para la Unidad Educativa Humberto Portocarrero de Fe y Alegría del Nivel Secundario. Utilizando metodología RUP y tecnologías Web adecuadas para el desarrollo del mismo.

El Sistema, ofrecerá información académica durante las 24 horas del día, pudiendo acceder a esta: Director (a), Secretaria, Profesores, padres de familia o apoderados y Alumnos. Logrando de esta manera intervenir oportunamente en la toma de decisiones.

1.2. Antecedentes.

Fe y Alegría es una entidad privada no lucrativa y está registrada oficialmente ante UNICEF y UNESCO.

Fe y Alegría concibe y ofrece servicios educativos públicos de calidad a los sectores discriminados por aspectos económicos, socio-políticos, étnico-culturales, de género, generacionales y de discapacidad, para potenciar su desarrollo personal y participación social.

Hoy en día, Fe y Alegría no cuentan con sistemas de gestión académica, pero aprovecha las Tecnologías de la Información y Comunicación (TIC), a través de un proyecto de Aulas Telemáticas.

La Unidad Educativa Humberto Portocarrero de Fe y Alegría en Tarija, cuenta con equipos computarizados, los mismos son utilizados para la capacitación de alumnos en el manejo de las computadoras, y en otros casos se hace un mal uso, pensando que se trata sólo de una máquina de escribir sofisticada, donde sólo realizan cartas, listas, etc.

El Sistema Web de Gestión Académica para la Unidad Educativa Humberto Portocarrero de Fe y Alegría del Nivel Secundario propuesto, ayudará a tomar decisiones favorables para la institución.

A continuación mencionaremos y describiremos brevemente, algunos sistemas Web de gestión académica.

Claroline (Classroom Online)

Descargar y usar claroline no implica costo alguno.

Claroline¹ es un sistema de gestión de cursos basados en Web. Permite a los profesores/as crear y administrar Web de cursos desde un navegador (explore, netscape...). Sin ser un “campo virtual”, permite disponer con una administración sencilla, de un espacio de encuentro donde se comparte herramientas con un grupo de estudiantes, en un “aula” complementaria a sus clases, accesible las 24 horas del día.

Se puede descargar el software de <http://www.claroline.net/download.html>, existe una versión para Windows y otra para Unix y Linux.

Este sistema tiene muchas funciones, sin embargo no cubre todos los requerimientos académicos esenciales que debe tener un sistema académico para colegios, como ser: registro de notas, registro de asistencia, procedimiento para desplegado de pagos de mensualidades, módulo que muestre la visión del colegio; por ese motivo sería poco útil para los colegios de nuestro medio.

¹ <http://www.claroline.net/>

IntraSchool

IntraSchool es una aplicación completa de gestión académica que le permite a la institución educativa monitorear sus procesos de gestión diarios de una manera eficiente y en línea. IntraSchool cumple con las principales prácticas de seguridad para la protección de datos².

La forma de introducir, consultar y gestionar la información es sencilla y rápida, y en ningún caso supone cambios en los procesos. Es una forma segura de transmitir la información, evitando cualquier pérdida.

Sistac

El Sistac es un sistema de gestión para colegios y establecimientos educacionales en general, a través de la implementación de un conjunto de sistemas modulares desarrollados íntegramente por profesionales peruanos³.

Sistac, permite que los profesores ingresen las notas bajo diferentes modalidades directamente al sistema, llevar registros de inasistencia, conducta, datos médicos, datos de orientación, históricos, datos completos de alumnos, apoderados, tutores y cuerpo docente y administrativo, generación de todo tipo de informes, estadísticos, oficiales, cálculo de promedios, etc.

SchoolTrack

SchoolTrack es un sistema avanzado que permite la gestión propia de los colegios de manera simple. No sólo permite manipular las áreas académicas de un colegio sino también la gestión de las finanzas, admisiones, biblioteca, enfermerías, etc. Además, permite a padres y alumnos el acceso a la información a través de Internet.

Una de las principales ventajas de SchoolTrack es su flexibilidad, lo que lo convierte en un sistema único en el mercado, configurable casi al 100%. Esta ventaja nos permite atender a colegios completamente diferentes.

² <http://www.intraschool.net/info/beneficio.php>

³ <http://www.multiserviciose.com/sistac.html>

Por otro lado, Colegium no vende el software sino que lo entrega mediante un sistema de suscripción que incluye formación a usuarios, soporte y actualizaciones.

La Facultad de Ciencias y Tecnología UMSS

La Facultad de Ciencias y Tecnología de la Universidad Mayor de San Simón en la ciudad de Cochabamba⁴, ha implementado un sistema Web, el mismo que cuenta con enlaces al sistema de información académica para los estudiantes de la UMSS. La administración del sistema se la realiza mediante roles de usuarios y manejo de sesiones.

Coimata

El sistema coimata⁵ ha sido implementado por la Universidad Autónoma Juan Misael Saracho de la ciudad de Tarija. Este sistema ha sido desarrollado con software libre y de código abierto por el equipo de la Dirección de Tecnologías de Información y Comunicación (DTIC) con el objetivo de automatizar los procesos académicos y administrativos de la Universidad.

Coimata, es un sistema Web autogestionable que contiene una base de datos que almacena información académica acerca de la comunidad universitaria, plantel docente y administrativo. Los estudiantes y docentes de la Universidad pueden acceder al sistema mediante una cuenta de usuario y clave.

1.3 Planteamiento del Problema.

Dilación en la elaboración de informes, registros, reportes y la forma inadecuada en difundir la información que ofrece la Unidad Educativa, Humberto Portocarrero de Fe y Alegría, gestionados de forma manual, ocasionando trámites innecesarios e ineficiencia en la toma de decisiones oportunas.

⁴ <http://www.fcyt.umss.edu.bo/>

⁵ <http://coimata.ujms.edu.bo>

1.4 Objetivos.

1.4.1 Objetivo General.

Analizar, diseñar e implementar un Sistema Web de gestión académica, para la Unidad Educativa Humberto Portocarrero de Fe y Alegría en la ciudad de Tarija (Nivel Secundario).

1.4.2 Objetivos Específicos

- Utilizar metodología RUP, en el desarrollo del sistema.
- Utilizar entrevistas, cuestionarios y el método Isac para conocer los procesos académicos que lleva a cabo la Unidad Educativa, e identificar los requerimientos específicos de la aplicación requerida.
- Utilizar UML Lenguaje Unificado de Modelado (Unified Modelling Language) para el modelado del sistema en las fases de análisis.
- Utilizar Postgresql para diseñar una base de datos.
- Ofrecer una interfaz amigable y fácil de aprender sin provocar cansancio y confusión al usuario.
- Utilizar: Php, Html, JavaScript.

1.5 Justificación.

1.5.1 Justificación Social.

El Sistema Web de Gestión Académica para el Colegio Humberto Portocarrero de Fe y Alegría en la ciudad de Tarija (Nivel Secundario) propuesto, ofrecerá diversos beneficios a la sociedad como ser:

Con este Sistema Web, los usuarios como ser: Director (a), secretaria, profesores, regente, tutores o padres de familia y estudiantes tendrán acceso a información académica las 24 horas del día. Además ahorrarán tiempo en el tratamiento y búsqueda automática de dicha información.

1.5.2 Justificación Académica.

El desarrollo del presente trabajo nos va a permitir aplicar conocimientos adquiridos durante los años de nuestra Educación Universitaria, como también aplicar la investigación que iremos realizando durante el desarrollo del mismo.

Este trabajo es presentado a consideración de la Universidad Autónoma “Juan Misael Saracho”, como requisito para optar el grado académico de Licenciatura en Ingeniería Informática.

1.5.3 Justificación Tecnológica.

El Sistema Web de Gestión Académica para el Colegio Humberto Portocarrero de Fe y Alegría en la ciudad de Tarija del nivel secundario, brinda un nuevo medio tecnológico de informatización, fácil de utilizar, de mantener y de actualizar, acorde con las nuevas exigencias tecnológicas.

Además que para su funcionamiento no precisa de un equipo de computación de última generación y para su operación se necesita tan sólo de un navegador.

1.6 Alcances y Limitaciones.

1.6.1 Alcances.

- Registro de estudiantes y docentes
- Registro del personal administrativo
- Automatización de inscripciones, estudiantes nuevos como antiguos.
- Registro de calificaciones de estudiantes
- Registro de asistencia de estudiantes
- Reportes de calificaciones y asistencia de estudiantes
- Búsquedas en la Base de Datos del sistema, para disminuir el tiempo en la accesibilidad.
- Administración de opciones

- Actualiza datos personales de los usuarios registrados.
- Habilita fechas para introducir calificaciones.
- Los usuarios del sistema podrán consultar información académica, utilizando como datos de entrada el nombre de usuario y password.
- Asigna materias a cursos y docentes.
- Administra los paralelos.

1.6.2 Limitaciones.

- No se gestionará información de carácter administrativa de la Unidad Educativa.
- No se realizará el control de asistencia del personal administrativo.
- No contempla la creación de foros, chat, noticias y eventos.
- No se transcribirá el contenido de todos los documentos (libros) considerados en este proyecto (sólo los necesarios para comprobar el funcionamiento del sistema).
- No se registrará la asistencia de los estudiantes de forma directa.

1.7 Explicación de las Alternativas de Solución

Implementar un Sistema Web Dinámico de Gestión Académica para la Unidad Educativa Humberto Portocarrero de Fe y Alegría del nivel secundario, que sirva como guía al Director(a), profesores, Secretaria, padres de familia o apoderados y Estudiantes, para estar informados oportunamente, desde cualquier lugar, simplemente conectándose a Internet.

2.1 Esquema Conceptual

2.1.1 Red

Generalmente, una red⁶, se define como el conjunto de computadoras interconectadas entre sí, mediante un sistema de cableado o medio de comunicación, con la finalidad de compartir información y los recursos incorporados en ella. Las redes pueden interconectarse con otras redes y contener sub-redes.

La topología más común, o configuración general de redes, incluye el bus, la estrella, y las topologías token ring. Las redes se pueden clasificar también en términos de la separación física entre nodos, como redes de área local (LAN, local area network), redes de área metropolitana (MAN, Metropolitan Area Network), y redes de área amplia (WAN, Wide Area Network).

Una cierta clase de redes puede también ser clasificada por el tipo de tecnología de la transmisión de datos que se emplea. Por ejemplo, una red TCP/IP (Transport Control Protocol/Internet Protocol), o una red del tipo SNA (Systems Network Architecture); si transporta voz, datos, o ambas clases de señales; por quién puede utilizar la red (pública o privada); por la naturaleza de sus conexiones (conmutada, dedicada o no dedicada, o por conexión virtual); y por los tipos de conexiones físicas (por ejemplo, fibra óptica, cable coaxial, y par trenzado sin blindaje UTP).

Las grandes redes de telefonía y las redes que usan su infraestructura (tal como el Internet) disponen de acuerdos para compartir e intercambiar recursos con otras compañías para formar redes mucho más grandes.

2.1.2 Internet

Internet⁷ fue desarrollado originariamente para los militares de Estados Unidos, y después se utilizó para el gobierno.

Es una combinación de hardware (ordenadores interconectados por vía telefónica o digital) y software (protocolos y lenguajes que hacen que todo funcione).

⁶ <http://es.wikipedia.org/wiki/red>

⁷ <http://es.wikipedia.org/wiki/internet>

Es una infraestructura de redes a escala mundial (grandes redes principales -tales como MILNET, NSFNET, y CREN - redes más pequeñas que conectan con ellas) que conecta a la vez a todos los tipos de ordenadores

Internet es una red mundial de computadoras, que mediante la comunicación entre todas estas computadoras a través de diferentes medios como cables telefónicos, redes de fibras ópticas, microondas, antenas parabólicas, satélites y hasta cables submarinos. Para que estas computadoras se entiendan unas con otras sin importar su marca, estructura o tecnología, se usa un protocolo o lenguaje común en Internet llamado TCP/IP. El TCP (Transport Control Protocol) o protocolo de control de transporte se encarga de asegurar que la información se transporte correctamente entre dos computadores que se conectan en la red empleando para ello diferentes técnicas de detección y corrección de pérdida de datos. El protocolo IP (Internet protocol) se encarga de encontrar en la red mundial el computador con el que se desea hacer una conexión y de manejar junto con el TCP todo lo relacionado a la comunicación entre las dos terminales.

Los usuarios de Internet, pueden realizar conversaciones directamente con otros usuarios en otras localidades usando el IRC (Internet Relay Chat).

2.1.3 Web

World Wide Web, o simplemente Web⁸, es el universo de información accesible a través de Internet, una fuente inagotable del conocimiento humano.

El componente más usado en el Internet es definitivamente el Web. Su característica sobresaliente es el texto remarcado, un método para referencias cruzadas instantáneas.

La exploración en el Web se realiza por medio de un software especial denominado Browser o Explorador. La apariencia de un Sitio Web puede variar ligeramente dependiendo del explorador que use. Así mismo, las versiones más recientes disponen de una funcionalidad mucho mayor tal como animación, realidad virtual, sonido y música.

⁸ <http://es.wikipedia.org/wiki/web>

2.1.3.1 Página Web

Una página Web es la unidad básica del World Wide Web.

Una página de Internet o página Web⁹ es un documento electrónico que contiene información específica de un tema en particular y que es almacenado en algún sistema de cómputo que se encuentre conectado a la red mundial de información denominada Internet, de tal forma que este documento pueda ser consultado por cualquier persona que se conecte a esta red mundial de comunicaciones y que cuente con los permisos apropiados para hacerlo.

Una página Web tiene la característica peculiar de que el texto se combina con imágenes para hacer que el documento sea dinámico y permita que se puedan ejecutar diferentes acciones, una tras otra, a través de la selección de texto remarcado o de las imágenes, acción que nos puede conducir a otra sección dentro del documento, abrir otra página Web, iniciar un mensaje de correo electrónico o transportarnos a otro Sitio Web totalmente distinto a través de sus hipervínculos.

Estos documentos pueden ser elaborados por los gobiernos, instituciones educativas, instituciones públicas o privadas, empresas o cualquier otro tipo de asociación, y por las propias personas en lo individual.

2.1.3.2 Sitio Web

Un sitio Web es un conjunto de archivos electrónicos y páginas Web que son accesibles mediante el protocolo HTTP de Internet¹⁰. Este conjunto de páginas tiene como objetivo el desarrollo de algún tema, prestar servicios o entregar contenidos asociados al entorno de la página.

Un sitio Web debe ser codificado por lo general en HTML, siendo navegable mediante un navegador de páginas Web. También se pueden crear páginas dinámicas usando servidores Web que posean soporte de algún lenguaje de programación. (Por ejemplo: PHP, ASP, JSP, etc.).

⁹ http://es.wikipedia.org/wiki/pagina_web

¹⁰ http://es.wikipedia.org/wiki/sitio_web

2.1.3.3 Portal Web

Un portal Web¹¹ es un sitio que recibe un alto tráfico y que está dirigido a resolver necesidades específicas de un grupo de personas.

Un portal Web puede ser un Centro de Atención a los clientes y prospectos de venta de su empresa, estos se pueden complementar con herramientas que le ayuden a levantar pedidos, atender los problemas de sus clientes, brindar correos electrónicos, motores de búsqueda, evaluaciones en línea, dar capacitación a distancia, etc.

Los portales normalmente tienen programación que requiere muchos recursos computacionales y por su alto tráfico generalmente se hospedan en servidores dedicados

Un portal Web es un sitio Web cuyo objetivo es ofrecer al usuario, de forma fácil e integrada, el acceso a una serie de recursos y de servicios, entre los que suelen encontrarse buscadores, foros, compra electrónica, etc.

2.1.4 Base de Datos

Una base de datos es un conjunto de datos que pertenecen al mismo contexto almacenados sistemáticamente para su uso posterior¹².

Una base de datos puede definirse como “un conjunto auto descriptivo de registros integrados”¹³.

En la actualidad, y en gran parte gracias a la tecnología y recursos disponibles provenientes de campos como la informática y la electrónica, las bases de datos pueden adquirir diversas formas, ofreciendo un amplio rango de soluciones al problema de almacenar datos.

Las aplicaciones más usuales son para la gestión de empresas e instituciones públicas.

¹¹ http://es.wikipedia.org/wiki/portal_web

¹² “Fundamentos de Base de Datos”: Korth Siverschatz

¹³ “Procesamiento de Base de Datos”: David M. Kroenke. Cap.1

2.1.5 Validación de Datos

Es esencial que los datos de entrada, junto con las peticiones de transacciones, sean válidos.

Los problemas que deben considerarse son los siguientes:

- Validación de datos de entrada; es decir datos faltantes, longitud de campo incorrecta, Los datos tienen una composición inaceptable, los datos están fuera de rango, los datos son inválidos, los datos no concuerdan con datos almacenados¹⁴.
- Validación de las transacciones de entrada; es decir enviar datos erróneos. datos enviados por una persona no autorizada. Pedir al sistema que ejecute una función inaceptable.

2.1.6 Seguridad en los Sistemas Informáticos

¿Contra qué debemos protegernos?

A menudo sólo pensamos en evitar que alguien pueda acceder a nuestros datos fraudulentamente. En cambio, la seguridad informática abarca un entorno más amplio no solo debemos evitar el acceso indebido a nuestros archivos y bases de datos. También hemos de proteger los mismos contra cualquier otro tipo de desastre.

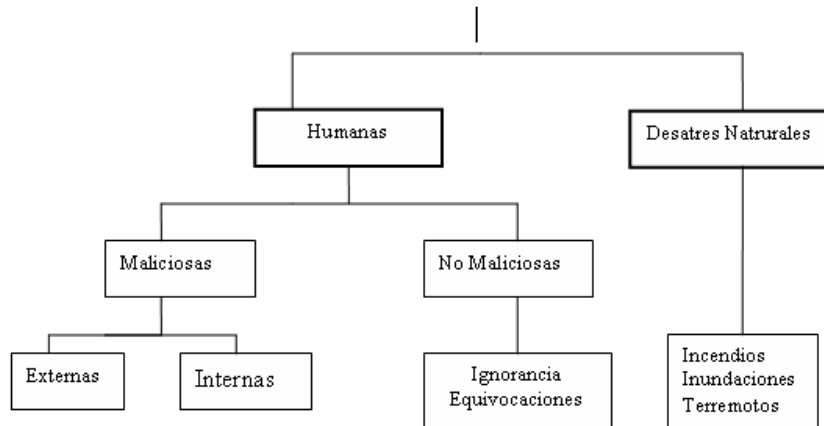
En Cualquier sistema informático, existen tres elementos básicos a proteger el hardware, el software y los datos. Además, generalmente se habla de un cuarto elemento llamado fungible: papel, cintas magnéticas, disquetes, cd-roms.

De los cuatro, los datos que manejan nuestro sistema serán los más importantes, ya que son el resultado del trabajo realizado. En caso de un desastre, el daño al hardware, software o a los elementos fungibles, pueden adquirirse nuevamente.

En cambio, los datos obtenidos en el transcurso del tiempo son difíciles o imposibles de recuperar: hemos de pasar obligatoriamente por un sistema de copias de seguridad y aún así, es difícil devolver los datos a su forma anterior al daño.

¹⁴ Análisis y Diseño de Sistemas: Kendall & Kendall

Amenazas para la seguridad.



2.1.6.1 Seguridad Física

La seguridad es “la administración del riesgo”. Es prácticamente imposible eliminar todas las posibles deficiencias que puede presentar nuestro sistema de información, la asignación de recursos, no solo económicos, sino humanos, sería prácticamente inviable, pero si se puede fijar niveles de seguridad que sean “aceptables” y sobre ellos trazar la política de seguridad de la institución.

Debe existir un especial cuidado al definir y proteger los equipos del sistema de información, éstos son elementos que permitan a los usuarios acceder a los ficheros, constituyendo la primera barrera de protección ante accesos no autorizados. El diseño de una adecuada política de protección de los equipos, puede evitar mucho trabajo posterior para lograr el nivel de seguridad deseado.

2.1.6.2 Seguridad Lógica

El activo más importante que se posee es la información, y por lo tanto deben existir técnicas que la aseguren, más allá de la seguridad física que se establezca sobre los equipos en los cuales se almacena. Estas técnicas las brinda la Seguridad Lógica consiste en la “aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo se permita acceder a ellos a las personas autorizadas para hacerlo¹⁵.”

¹⁵ Internet:<http://eonline.eurosystems.es/html/seguridad.asp>

Existen un viejo dicho en la seguridad informática que dicta que “todo lo que no esta permitido debe estar prohibido” y esto es lo que debe asegurar la Seguridad Lógica.

Los objetivos que se plantean serán:

- Restringir el acceso a los programas y archivos de la Base de Datos
- Definir controles para la introducción de datos

2.2 Herramientas Tecnológicas para el Desarrollo del Sistema

2.2.1 Html

El HTML, acrónimo inglés de **H**yper **T**ext **M**arkup **L**anguage (lenguaje de marcación de hipertexto), es un lenguaje de marcas diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores del tipo Explorer o Netscape, el HTML se ha convertido en uno de los formatos más populares que existen para la construcción de documentos.

HTML se compone de una serie de comandos, que son interpretados por el visualizador, o programa que utilizamos para navegar por el WWW. En última instancia es el visualizador el que ejecuta todas las órdenes contenidas en el código HTML, de forma que un visualizador puede estar capacitado para unas prestaciones, pero no para otras. Así, podremos especificar que una página tenga una imagen de fondo, o un texto parpadeando, pero si nuestro visualizador no está capacitado para esas funciones, no podremos comprobarlas.

La ventaja más importante de este lenguaje, además de la compatibilidad, es la posibilidad de establecer enlaces entre documentos relacionados, esto es lo que se conoce como hipertexto, o hipervínculo, se puede crear enlaces a otras direcciones.

2.2.2 JavaScript

JavaScript es el lenguaje que nos permite interactuar con el navegador de manera dinámica y eficaz, proporcionando a las páginas Web dinamismo y vida.

Se trata de un lenguaje de tipo script compacto, basado en objetos y guiado por eventos diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet.

Los programas JavaScript¹⁶ van incrustados en los documentos HTML, y se encargan de realizar acciones en el cliente, como ser: pedir datos, confirmaciones, mostrar mensajes, crear animaciones, comprobar campos.

Existen distintos modos de incluir lenguaje JavaScript en una página. La forma más frecuente de hacerlo es utilizando la directiva `<script>` en un documento HTML (se pueden incluir tantas directivas `<script>` como se quiera en un documento).

2.2.3 Php

El PHP (acrónimo de "PHP: Hypertext Preprocessor"), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. El PHP es un lenguaje de script incrustado dentro del HTML. La mayor parte de su sintaxis ha sido tomada de C, Java y Perl con algunas características específicas de sí mismo. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas.

Una de sus características más potentes es su soporte para gran cantidad de bases de datos. Entre su soporte pueden mencionarse Interbase, Msql, Mysql, Oracle, Informix, Postgresql, entre otras.

PHP también ofrece la integración con las varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa desde generar documentos en pdf hasta analizar código XML.

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día que el código comparables en otros lenguajes.

¹⁶ <http://es.wikipedia.org/wiki/javascript>

Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.

PHP es la opción natural para los programadores en máquinas con Linux que ejecutan servidores web con Apache, pero funciona igualmente bien en cualquier otra plataforma de UNIX o de Windows,

Para trabajar con capacidades PHP, se puede conseguir mayor información en PHP.net, sitio encargado de mantener al día a todos los desarrolladores con las últimas descargas relacionadas con el lenguaje y documentación.

2.2.4 Rational Rose

Rational Rose es la mejor herramienta de modelado del mundo. Para los analistas de negocios, Rose ofrece la capacidad de modelar y visualizar sus procesos de negocios y destacar oportunidades para aumentar la eficiencia.

Para los analistas de datos, el modelado de su diseño de base de datos en Rose, mejora la comunicación entre usted y los desarrolladores, y, cuando usted modela sus casos de uso en Rose, asegura que la solución sea creada con el usuario en mente. Rational Rose unifica a los analistas de negocios, sistemas y datos al permitirles crear y administrar modelos en una herramienta con un solo lenguaje de modelado.

2.2.5 Apache

Apache¹⁷ es un servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etcétera), Windows y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Su nombre se debe a que originalmente Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor *parcheado*).

¹⁷ <http://es.wikipedia.org/wiki/Servidor>

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: en el 2005, Apache es el servidor HTTP más usado, siendo el servidor HTTP del 70% de los sitios web en el mundo y creciendo aún su cuota de mercado (estadísticas históricas y de uso diario proporcionadas por Netcraft).

2.2.6 Photoshop

Photoshop¹⁸ es una aplicación informática de edición y retoque de imágenes bitmap elaborada por la compañía de software Adobe inicialmente para computadores Apple pero posteriormente también para plataformas PC.

Photoshop se ha convertido, casi desde sus comienzos, en el estándar mundial en retoque fotográfico, pero también se usa extensivamente en multitud de disciplinas del campo del diseño y fotografía, como diseño web, composición de imágenes bitmap, estilismo digital, fotocomposición, edición y grafismos de vídeo y básicamente en cualquier actividad que requiera el tratamiento de imágenes digitales.

Photoshop soporta muchos tipos de archivos de imágenes, como BMP, JPG, PNG, GIF, entre otros, pero tiene ciertas extensiones propias como lo son:

- **PSD** (Photoshop Document): Es un formato que guarda una imagen como un grupo de capas, métodos de fusión, colores, textos, máscaras, canales de color, canales alfa, trazados, formas, configuración de tonos, entre otras. Este es un formato muy popular que incluso es soportado por programas de la competición.
- **PSB**: Es una nueva versión del formato PSD, diseñado especialmente para archivos mayores a 2 GB.

¹⁸ <http://es.wikipedia.org/wiki/Photoshop>

- **PDD:** Es una versión del PSD que sólo soporta las opciones del programa descontinuado PhotoDeluxe.

2.2.7 Postgresql

PostgreSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL, Firebird y MaxDB), así como sistemas propietarios como Oracle o DB2.

Algunas de sus principales características son:

- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (*foreign keys*)
- Disparadores (*triggers*)
- Vistas
- Integridad transaccional
- Acceso concurrente multiversión (no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe)
- Capacidad de albergar programas en el servidor en varios lenguajes.
- Herencia de tablas
- Tipos de datos y operaciones geométricas

2.2.8 Macromedia Flash¹⁹

Macromedia Flash se reconoce como una herramienta "que hay que tener", empleada por un millón de profesionales que la utilizan para brindar las mejores experiencias, tanto los diseñadores como los desarrolladores se benefician del entorno accesible y poderoso que se necesita para crear y brindar contenido y aplicaciones para Internet de vanguardia.

Como herramientas principales tenemos:

¹⁹ http://www.svetlian.com/Webmaster/flash_tutor1.htm

Compatibilidad de vídeo Crea contenidos y aplicaciones dinámicas, distintivas, que incorporan además vídeo. Importe cualquier archivo de vídeo estándar que sea compatible con QuickTime o Windows Media Player, como MPEG, DV (Digital video), MOV (Quick Time), y AVI. Manipule, ajuste a escala, rote, sesgue o ponga máscaras y anime objetos de vídeo haciéndolos interactivos a través de scripts.

Inspector de propiedades y el panel de respuestas Las mejoras hechas a la interfaz de usuario incluyen los paneles adaptables y un nuevo inspector de propiedades sensible al contexto que resalta las herramientas y los objetos que se pueden utilizar en relación a un elemento seleccionado.

Carga dinámica de imágenes y sonidos puede cargar dinámicamente archivos JPEG y MP3 de medios externos durante la ejecución, lo cual se traduce en un tamaño de archivo más pequeño y la capacidad del desarrollador de modificar fácilmente el contenido.

Diseño gráfico y mejoras al mezclador de color con la nueva herramienta Transformación libre y el ajuste en píxeles que ofrecen una mayor libertad creativa y dominio sobre el contenido gráfico los diseñadores trabajan más rápido.

Desarrolle y reproduzca contenido accesible integra ahora herramientas para crear contenido de accesibilidad a todos. Para añadir texto descriptivo a las animaciones y elementos de interfaces de usuario, seleccione un elemento y escriba una descripción, etc.

2.2.9 PEAR

PEAR, o PHP Extensión and Application Repository, es un entorno de desarrollo y sistema de distribución para componentes de código PHP. El proyecto PEAR fue fundado por Williams G. Molina G. en 1999 para promover la reutilización de código que realizan tareas comunes.

El proyecto tiene como metas:

- Promover una biblioteca de código bien estructurada

- Mantener un sistema de distribución y mantenimiento de paquetes de código
- Promover un estilo de codificación estándar

En definitiva consiste en una lista bastante grande de bibliotecas de código PHP que permiten hacer ciertas tareas de manera más rápida y eficiente reutilizando código escrito previamente por otras personas. Generalmente las bibliotecas contienen clases en archivos PHP que luego se incluyen y usan sin muchas complicaciones.

2.3.10 Arquitectura Cliente Servidor

La arquitectura cliente-servidor²⁰ llamado modelo cliente-servidor o servidor-cliente es una forma de dividir y especializar programas y equipos de cómputo a fin de que la tarea que cada uno de ellos realice se efectúe con la mayor eficiencia, y permita simplificarlas.

En esta arquitectura la capacidad de proceso está repartida entre el servidor y los clientes.

En el modelo cliente-servidor, el trabajo se reparte entre dos ordenadores. De acuerdo con la distribución de la lógica de la aplicación hay dos posibilidades:

1. Cliente liviano: si el cliente sólo se hace cargo de la presentación.
2. Cliente pesado: si el cliente asume también la lógica del negocio.

En la actualidad se suele hablar de arquitectura de tres niveles, donde la capa de almacenamiento y la de aplicación se ubican en (al menos) dos servidores diferentes, conocidos como servidores de datos y servidores de aplicaciones.

Ventajas de la arquitectura cliente-servidor:

- El servidor no necesita tanta potencia de procesamiento, parte del proceso se reparte con los clientes.
- Se reduce el tráfico de red considerablemente. Idealmente, el cliente se conecta al servidor cuando es estrictamente necesario, obtiene los datos que necesita y cierra la conexión dejando la red libre para otra conexión.

²⁰ <http://es.wikipedia.org/wiki/Cliente-servidor>

2.3 Determinación de Requerimientos Análisis y Diseño del Sistema

2.3.1 UML

Lenguaje Unificado de Modelado²¹ (UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; aún cuando todavía no es un estándar oficial. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software.

El punto importante para notar aquí es que UML es un "lenguaje" para especificar y no un método o un proceso. UML se usa para definir un sistema de software; para detallar los artefactos en el sistema; para documentar y construir -es el lenguaje en el que está descrito el modelo. UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) -pero no especifica en sí mismo qué metodología o proceso usar.

UML cuenta con varios tipos de diagramas, los cuales muestran diferentes aspectos de las entidades representadas.

Los diagramas de UML son los siguientes:

- Diagrama de Casos de Uso: modela la funcionalidad del sistema agrupándola en descripciones de acciones ejecutadas por un sistema para obtener un resultado.
- Diagrama de Clases: muestra las clases (descripciones de objetos que comparten características comunes) que componen el sistema y cómo se relacionan entre sí.
- Diagrama de Objetos: muestra una serie de objetos (instancias de las clases) y sus relaciones.
- Diagrama de Secuencia: enfatiza la interacción entre los objetos y los mensajes que intercambian entre sí junto con el orden temporal de los mismos.

²¹ <http://es.wikipedia.org/wiki/Uml>

- Diagrama de Colaboración: igualmente, muestra la interacción entre los objetos resaltando la organización estructural de los objetos en lugar del orden de los mensajes intercambiados.
- Diagrama de Estados: modela el comportamiento de acuerdo con eventos.
- Diagrama de Actividades: simplifica el Diagrama de Estados modelando el comportamiento mediante flujos de actividades.
- Diagrama de Componentes: muestra la organización y las dependencias entre un conjunto de componentes.
- Diagrama de Despliegue: muestra los dispositivos que se encuentran en un sistema y su distribución en el mismo.

2.3.1.1 Diagramas de Casos de Uso

Un caso de uso es una secuencia de transacciones que son desarrolladas por un sistema en respuesta a un evento que inicia un actor sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la funcionalidad y el comportamiento de un sistema mediante su interacción con los usuarios y/o otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la relación y la generalización son relaciones.

2.3.1.2 Diagramas de Secuencia

Este diagrama muestra las interacciones entre los objetos organizadas en una secuencia temporal. En particular muestra los objetos participantes en la interacción y la secuencia de mensajes intercambiados.

Representa una interacción, un conjunto de comunicaciones entre objetos organizadas visualmente por orden temporal. A diferencia de los diagramas de colaboración, los diagramas de secuencia incluyen secuencias temporales pero no incluyen las relaciones entre objetos. Pueden existir de forma de descriptor (describiendo todos los posibles escenarios) y en forma de instancia (describiendo un escenario real).

Dentro del conjunto de mensajes representados dispuestos en una secuencia temporal, cada rol en la secuencia se muestra como una línea de vida, es decir, una línea vertical que representa el rol durante cierto plazo de tiempo, con la interacción completa. Los mensajes se muestran como flechas entre líneas de vida. Un diagrama de secuencia puede mostrar un escenario, es decir, una historia individual de transacción. Un uso de un diagrama de secuencia es mostrar la secuencia del comportamiento de un caso de uso.

2.3.1.3 Diagramas de Actividades

Un diagrama de Actividad demuestra la serie de actividades que deben ser realizadas en un uso-caso, así como las distintas rutas que pueden irse desencadenando en el uso-caso.

Es importante recalcar que aunque un diagrama de actividad es muy similar en definición a un diagrama de flujo (típicamente asociado en el diseño de Software), estos no son lo mismo. Un diagrama de actividad es utilizado en conjunción de un diagrama uso-caso para auxiliar a los miembros del equipo de desarrollo a entender como es utilizado el sistema y como reacciona en determinados eventos. Lo anterior, en contraste con un diagrama de flujo que ayuda a un programador a desarrollar código a través de una descripción lógica de un proceso. Se pudiera considerar que un diagrama de actividad describe el *problema*, mientras un diagrama de flujo describe la *solución*.

2.3.1.4 Diagramas de Clases

Los diagramas de clases son diagramas de estructura estática que muestran las clases del sistema y sus interrelaciones (incluyendo herencia, agregación, asociación, etc). Los diagramas de clase son el pilar básico del modelado con UML, siendo utilizados tanto para mostrar lo que el sistema puede hacer (análisis), como para mostrar cómo puede ser construido (diseño). El diagrama de clases de más alto nivel (*main class diagram*), será lógicamente un dibujo de los paquetes que componen el sistema. A su vez cada paquete tendrá un *main class diagram* que muestra las clases del paquete.

Las clases se documentan con una descripción de lo que hacen, sus métodos y sus atributos. Las relaciones entre clases se documentan con una descripción de su propósito, su cardinalidad (cuantos objetos intervienen en la relación) y su opcionalidad (cuando un objeto es opcional el que intervenga en una relación). La descripción de clases complejas se puede documentar con diagramas de estados

2.4 Metodologías

2.4.1 Método ISAC- Análisis de Cambios y Estudios de Actividades

El método de ingeniería de requerimientos ISAC empieza analizando el problema real de una organización, en este caso un problema real de la Unidad Educativa y encuentra soluciones reales para ellos. Si una solución involucra el desarrollo de un sistema entonces ISAC continúa desarrollando este sistema, lo que pone en marcha, y estudia detalladamente la institución de forma que sea el soporte claro de las actividades del sistema.

ISAC es un método orientado al cliente en ambientes de sistemas de información²². Sus características principales son:

- Método orientado a cliente
- Asegura que se comprendan las necesidades de información.
- A partir de un análisis de problemas, encuentra soluciones.

2.4.2 Método Rup

Método de Desarrollo Rup Rational Unified Process (RUP). Es un proceso de Ingeniería de Software cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos.

²² Requeriments Engineering; R. J. Wieringa

Características esenciales

Los autores de RUP destacan que el proceso de software propuesto por RUP tiene tres características esenciales: está dirigido por los Casos de Uso, está centrado en la arquitectura, y es iterativo e incremental.

Proceso dirigido por Casos de Uso

Según [Kru00], los Casos de Uso son una técnica de captura de requisitos que fuerza a pensar en términos de importancia para el usuario y no sólo en términos de funciones que sería bueno contemplar. Se define un Caso de Uso como un fragmento de funcionalidad del sistema que proporciona al usuario un valor añadido. Los Casos de Uso representan los requisitos funcionales del sistema.

En RUP los Casos de Uso no son sólo una herramienta para especificar los requisitos del sistema. También guían su diseño, implementación y prueba. Los Casos de Uso constituyen un elemento integrador y una guía del trabajo como se muestra en la Figura 1.

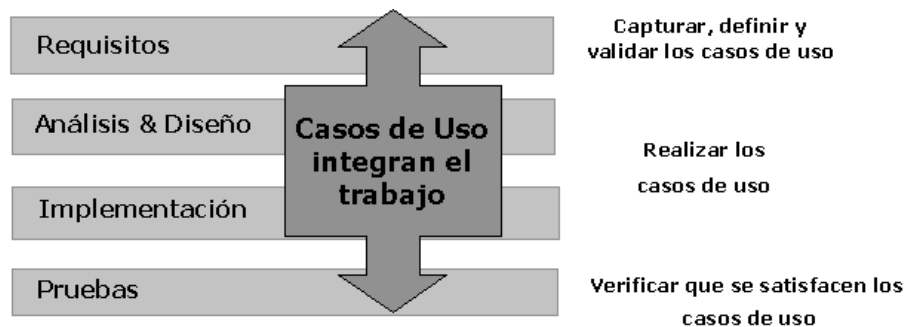


Figura 1: Los Casos de Uso integran el trabajo

Los Casos de Uso no sólo inician el proceso de desarrollo sino que proporcionan un hilo conductor, permitiendo establecer trazabilidad entre los artefactos que son generados en las diferentes actividades del proceso de desarrollo.

Como se muestra en la Figura 2, basándose en los Casos de Uso se crean los modelos de análisis y diseño, luego la implementación que los lleva a cabo, y se verifica que

efectivamente el producto implemente adecuadamente cada Caso de Uso. Todos los modelos deben estar sincronizados con el modelo de Casos de Uso.

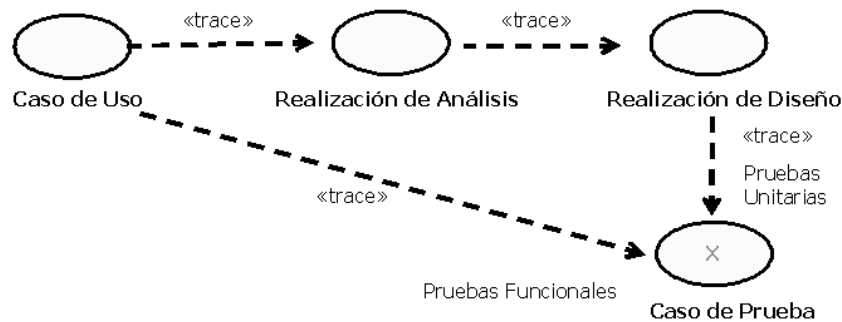


Figura 2: Trazabilidad a partir de los Casos de Uso

Proceso centrado en la arquitectura

La arquitectura de un sistema es la organización o estructura de sus partes más relevantes, lo que permite tener una visión común entre todos los involucrados (desarrolladores y usuarios) y una perspectiva clara del sistema completo, necesaria para controlar el desarrollo [Kru00].

La arquitectura involucra los aspectos estáticos y dinámicos más significativos del sistema, está relacionada con la toma de decisiones que indican cómo tiene que ser construido el sistema y ayuda a determinar en qué orden. Además la definición de la arquitectura debe tomar en consideración elementos de calidad del sistema, rendimiento, reutilización y capacidad de evolución por lo que debe ser flexible durante todo el proceso de desarrollo. La arquitectura se ve influenciada por la plataforma software, sistema operativo, gestor de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados. Muchas de estas restricciones constituyen requisitos no funcionales del sistema.

En el caso de RUP además de utilizar los Casos de Uso para guiar el proceso se presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento.

Cada producto tiene tanto una función como una forma. La función corresponde a la funcionalidad reflejada en los Casos de Uso y la forma la proporciona la arquitectura. Existe una interacción entre los Casos de Uso y la arquitectura, los Casos de Uso deben encajar en la arquitectura cuando se llevan a cabo y la arquitectura debe permitir el desarrollo de todos los Casos de Uso requeridos, actualmente y en el futuro. Esto provoca que tanto arquitectura como Casos de Uso deban evolucionar en paralelo durante todo el proceso de desarrollo de software.

En la Figura 3 se ilustra la evolución de la arquitectura durante las fases de RUP. Se tiene una arquitectura más robusta en las fases finales del proyecto. En las fases iniciales lo que se hace es ir consolidando la arquitectura por medio de *baselines* y se va modificando dependiendo de las necesidades del proyecto.

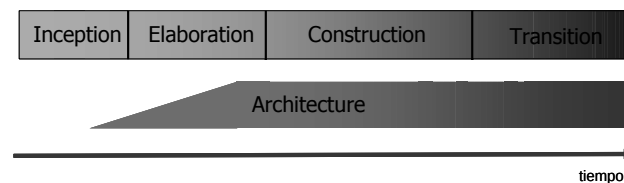


Figura 3: Evolución de la arquitectura del sistema

Es conveniente ver el sistema desde diferentes perspectivas para comprender mejor el diseño por lo que la arquitectura se representa mediante varias vistas que se centran en aspectos concretos del sistema, abstrayéndose de los demás. Para RUP, todas las vistas juntas forman el llamado modelo 4+1 de la arquitectura [Kru95], el cual recibe este nombre porque lo forman las vistas lógica, de implementación, de proceso y de despliegue, más la de Casos de Uso que es la que da cohesión a todas.

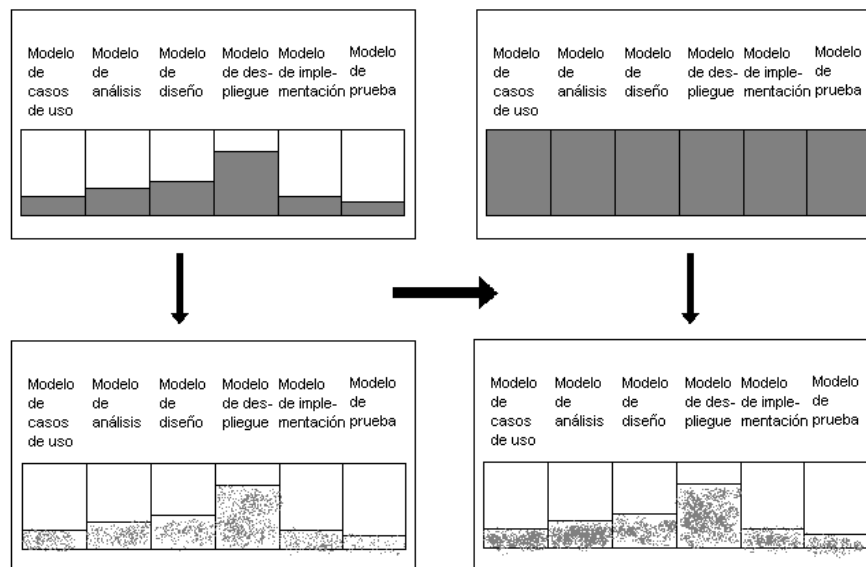


Figura 4: Los modelos se completan, la arquitectura no cambia drásticamente

Al final de la fase de elaboración se obtiene una *baseline*²³ de la arquitectura donde fueron seleccionados una serie de Casos de Uso arquitectónicamente relevantes (aquellos que ayudan a mitigar los riesgos más importantes, aquellos que son los más importantes para el usuario y aquellos que cubran las funcionalidades significativas)

Como se observa en la Figura 4, durante la construcción los diversos modelos van desarrollándose hasta completarse (según se muestra con las formas rellenas en la esquina superior derecha). La descripción de la arquitectura sin embargo, no debería cambiar significativamente (abajo a la derecha) debido a que la mayor parte de la arquitectura se decidió durante la elaboración. Se incorporan pocos cambios a la arquitectura (indicados con mayor densidad de puntos en la figura inferior derecha) [JBR00].

²³ Una baseline es una instantánea del estado de todos los artefactos del proyecto, registrada para efectos de gestión de configuración y control de cambios

Proceso iterativo e incremental

Según [JBR00] el equilibrio correcto entre los Casos de Uso y la arquitectura es algo muy parecido al equilibrio de la forma y la función en el desarrollo del producto, lo cual se consigue con el tiempo. Para esto, la estrategia que se propone en RUP es tener un proceso iterativo e incremental en donde el trabajo se divide en partes más pequeñas o mini proyectos. Permitiendo que el equilibrio entre Casos de Uso y arquitectura se vaya logrando durante cada mini proyecto, así durante todo el proceso de desarrollo. Cada mini proyecto se puede ver como una iteración (un recorrido más o menos completo a lo largo de todos los flujos de trabajo fundamentales) del cual se obtiene un incremento que produce un crecimiento en el producto.

Una iteración puede realizarse por medio de una cascada como se muestra en la Figura 5. Se pasa por los flujos fundamentales (Requisitos, Análisis, Diseño, Implementación y Pruebas), también existe una planificación de la iteración, un análisis de la iteración y algunas actividades específicas de la iteración. Al finalizar se realiza una integración de los resultados con lo obtenido de las iteraciones anteriores.

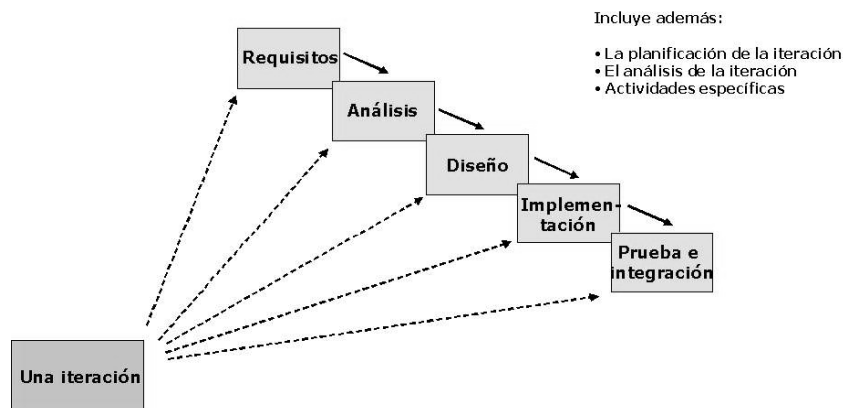


Figura 5: Una iteración RUP

El proceso iterativo e incremental consta de una secuencia de iteraciones. Cada iteración aborda una parte de la funcionalidad total, pasando por todos los flujos de trabajo relevantes y refinando la arquitectura. Cada iteración se analiza cuando termina.

Se puede determinar si han aparecido nuevos requisitos o han cambiado los existentes, afectando a las iteraciones siguientes. Durante la planificación de los detalles de la siguiente iteración, el equipo también examina cómo afectarán los riesgos que aún quedan al trabajo en curso. Toda la retroalimentación de la iteración pasada permite reajustar los objetivos para las siguientes iteraciones. Se continúa con esta dinámica hasta que se haya finalizado por completo con la versión actual del producto.

RUP divide el proceso en cuatro fases, dentro de las cuales se realizan varias iteraciones en número variable según el proyecto y en las que se hace un mayor o menor hincapié en las distintas actividades. En la Figura 6 se muestra cómo varía el esfuerzo asociado a las disciplinas según la fase en la que se encuentre el proyecto RUP.

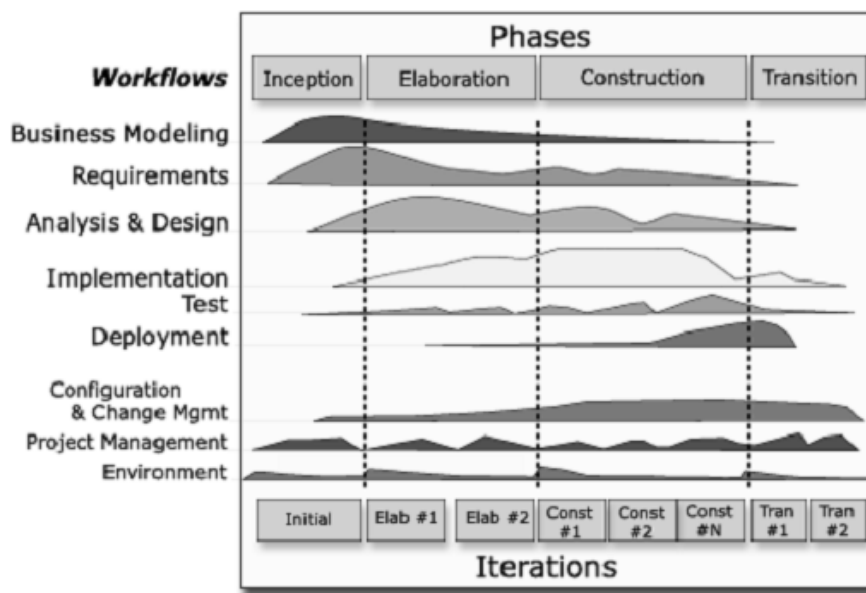


Figura 6: Esfuerzo en actividades según fase del proyecto

Las primeras iteraciones (en las fases de Inicio y Elaboración) se enfocan hacia la comprensión del problema y la tecnología, la delimitación del ámbito del proyecto, la eliminación de los riesgos críticos, y al establecimiento de una *baseline* de la arquitectura.

Durante la fase de inicio las iteraciones hacen poner mayor énfasis en actividades modelado del negocio y de requisitos.

En la fase de elaboración, las iteraciones se orientan al desarrollo de la *baseline* de la arquitectura, abarcan más los flujos de trabajo de requerimientos, modelo de negocios (refinamiento), análisis, diseño y una parte de implementación orientado a la *baseline* de la arquitectura.

En la fase de construcción, se lleva a cabo la construcción del producto por medio de una serie de iteraciones.

Para cada iteración se selecciona algunos Casos de Uso, se refina su análisis y diseño y se procede a su implementación y pruebas. Se realiza una pequeña cascada para cada ciclo. Se realizan tantas iteraciones hasta que se termine la implementación de la nueva versión del producto.

En la fase de transición se pretende garantizar que se tiene un producto preparado para su entrega a la comunidad de usuarios.

Como se puede observar en cada fase participan todas las disciplinas, pero que dependiendo de la fase el esfuerzo dedicado a una disciplina varía.

Estructura del proceso

El proceso puede ser descrito en dos dimensiones o ejes [RSC98]:

Eje horizontal: Representa el tiempo y es considerado el eje de los aspectos dinámicos del proceso. Indica las características del ciclo de vida del proceso expresado en términos de fases, iteraciones e hitos. Se puede observar en la Figura 7 que RUP consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Como se mencionó anteriormente cada fase se subdivide a la vez en iteraciones.

Eje vertical: Representa los aspectos estáticos del proceso. Describe el proceso en términos de componentes de proceso, disciplinas, flujos de trabajo, actividades, artefactos y roles.

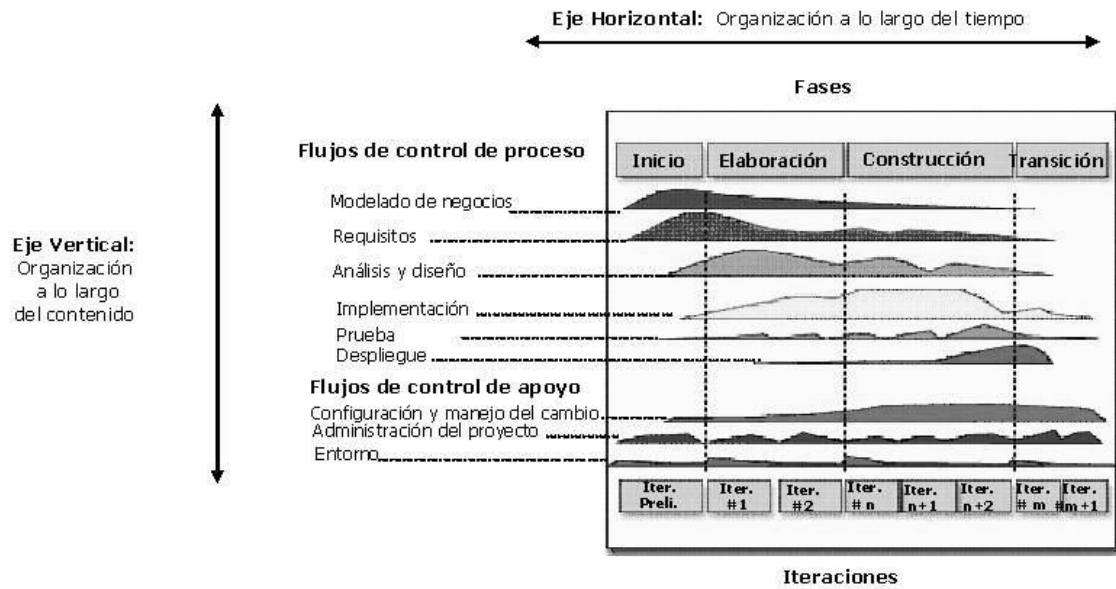


Figura 7: Estructura de RUP

Estructura Dinámica del proceso. Fases e iteraciones

RUP se repite a lo largo de una serie de ciclos que constituyen la vida de un producto. Cada ciclo concluye con una generación del producto para los clientes. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición. Cada fase se subdivide a la vez en iteraciones, el número de iteraciones en cada fase es variable.

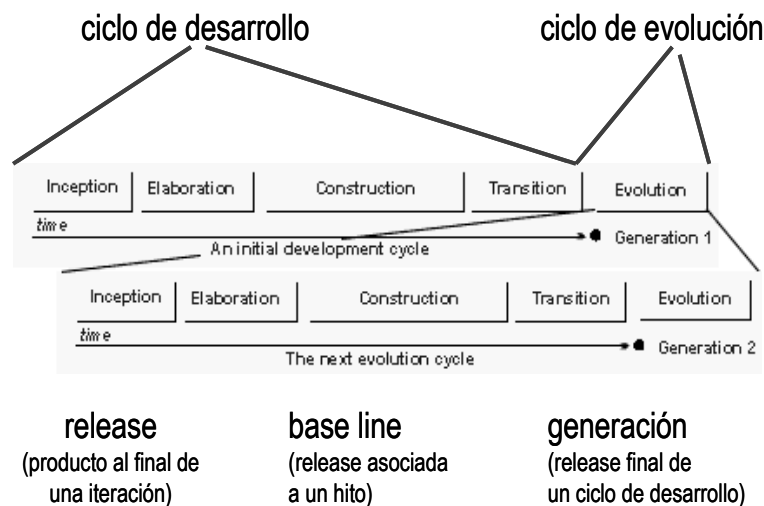


Figura 8: Ciclos, releases, baseline

Cada fase se concluye con un hito bien definido, un punto en el tiempo en el cual se deben tomar ciertas decisiones críticas y alcanzar las metas clave antes de pasar a la siguiente fase, ese hito principal de cada fase se compone de hitos menores que podrían ser los criterios aplicables a cada iteración. Los hitos para cada una de las fases son: Inicio - *Lifecycle Objectives*, Elaboración - *Lifecycle Architecture*, Construcción - *Initial Operational Capability*, Transición - Product Release. Las fases y sus respectivos hitos se ilustran en la Figura 9.

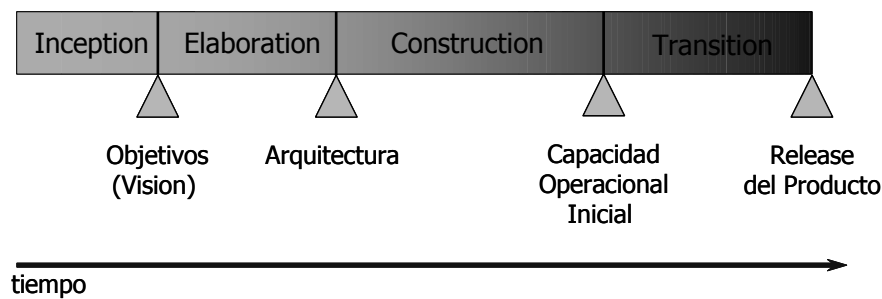


Figura 9: Fases e hitos en RUP

La duración y esfuerzo dedicado en cada fase es variable dependiendo de las características del proyecto. Sin embargo, la Figura 10 ilustra porcentajes frecuentes al respecto. Consecuente con el esfuerzo señalado, la Figura 11 ilustra una distribución típica de recursos humanos necesarios a lo largo del proyecto.

	Inicio	Elaboración	Construcción	Transición
Esfuerzo	5 %	20 %	65 %	10%
Tiempo Dedicado	10 %	30 %	50 %	10%

Figura 10: Distribución típicas de esfuerzo y tiempo

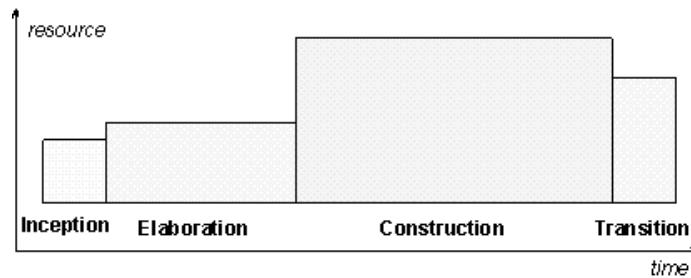


Figura 11: Distribución típica de recursos humanos

Inicio

Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20% del modelo completo). Se desarrolla, un plan de negocio para determinar que recursos deben ser asignados al proyecto.

Los objetivos de esta fase son [KRU00]:

- Establecer el ámbito del proyecto y sus límites.
- Encontrar los Casos de Uso críticos del sistema, los escenarios básicos que definen la funcionalidad.
- Mostrar al menos una arquitectura candidata para los escenarios principales.
- Estimar el coste en recursos y tiempo de todo el proyecto.
- Estimar los riesgos, las fuentes de incertidumbre.

Los resultados de la fase de inicio deben ser [RSC98]:

- Un documento de visión: Una visión general de los requerimientos del proyecto, características clave y restricciones principales.
- Modelo inicial de Casos de Uso (10-20% completado).
- Un glosario inicial: Terminología clave del dominio.
- El caso de negocio.
- Lista de riesgos y plan de contingencia.

- Plan del proyecto, mostrando fases e iteraciones.
- Modelo de negocio, si es necesario
- Prototipos exploratorios para probar conceptos o la arquitectura candidata.

Al terminar la fase de inicio se deben comprobar los criterios de evaluación para continuar:

- Todos los interesados en el proyecto coinciden en la definición del ámbito del sistema y las estimaciones de agenda.
- Entendimiento de los requisitos, como evidencia de la fidelidad de los Casos de Uso principales.
- Comprensión total de cualquier prototipo de la arquitectura desarrollado.
- Los gastos hasta el momento se asemejan a los planeados.

Si el proyecto no pasa estos criterios hay que plantearse abandonarlo o repensarlo profundamente.

Elaboración

El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.

En esta fase se construye un prototipo de la arquitectura, que debe evolucionar en iteraciones sucesivas hasta convertirse en el sistema final. Este prototipo debe contener los Casos de Uso críticos identificados en la fase de inicio. También debe demostrarse que se han evitado los riesgos más graves.

Los objetivos de esta fase son [KRU00]:

- Definir, validar y cimentar la arquitectura.
- Completar la visión.
- Crear un plan fiable para la fase de construcción. Este plan puede evolucionar en sucesivas iteraciones. Debe incluir los costes si procede.

- Demostrar que la arquitectura propuesta soportará la visión con un coste razonable y en un tiempo razonable.

Al terminar deben obtenerse los siguientes resultados [RSC98]:

- Un modelo de Casos de Uso completa al menos hasta el 80%: todos los casos y actores identificados, la mayoría de los casos desarrollados.
- Requisitos adicionales que capturan los requisitos no funcionales y cualquier requisito no asociado con un Caso de Uso específico.
- Descripción de la arquitectura software.
- Un prototipo ejecutable de la arquitectura.
- Lista de riesgos y caso de negocio revisados.
- Plan de desarrollo para el proyecto.
- Un caso de desarrollo actualizado que especifica el proceso a seguir.
- Un manual de usuario preliminar (opcional).

En esta fase se debe tratar de abarcar todo el proyecto con la profundidad mínima. Sólo se profundiza en los puntos críticos de la arquitectura o riesgos importantes.

En la fase de elaboración se actualizan todos los productos de la fase de inicio.

Los criterios de evaluación de esta fase son los siguientes:

- La visión del producto es estable.
- La arquitectura es estable.
- Se ha demostrado mediante la ejecución del prototipo que los principales elementos de riesgo han sido abordados y resueltos.
- El plan para la fase de construcción es detallado y preciso. Las estimaciones son creíbles.
- Todos los interesados coinciden en que la visión actual será alcanzada si se siguen los planes actuales en el contexto de la arquitectura actual.

- Los gastos hasta ahora son aceptables, comparados con los previstos.

Si no se superan los criterios de evaluación quizá sea necesario abandonar el proyecto o replanteárselo considerablemente.

Construcción

La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones. Durante esta fase todos los componentes, características y requisitos deben ser implementados, integrados y probados en su totalidad, obteniendo una versión aceptable del producto.

Los objetivos concretos según [KRU00] incluyen:

- Minimizar los costes de desarrollo mediante la optimización de recursos y evitando el tener que rehacer un trabajo o incluso desecharlo.
- Conseguir una calidad adecuada tan rápido como sea práctico.
- Conseguir versiones funcionales (alfa, beta, y otras versiones de prueba) tan rápido como sea práctico.

Los resultados de la fase de construcción deben ser [RSC98]: □ □

- Modelos Completos (Casos de Uso, Análisis, Diseño, Despliegue e Implementación)
- Arquitectura íntegra (mantenida y mínimamente actualizada)
- Riesgos Presentados Mitigados
- Plan del Proyecto para la fase de Transición.
- Manual Inicial de Usuario (con suficiente detalle)
- Prototipo Operacional – beta
- Caso del Negocio Actualizado

Los criterios de evaluación de esta fase son los siguientes:

- El producto es estable y maduro como para ser entregado a la comunidad del usuario para ser probado.

- Todos los usuarios expertos están listos para la transición en la comunidad de usuarios.
- Son aceptables los gastos actuales versus los gastos planeados.

Transición

La finalidad de la fase de transición es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

En [KRU00] se citan algunas de las cosas que puede incluir esta fase:

- Prueba de la versión Beta para validar el nuevo sistema frente a las expectativas de los usuarios
- Funcionamiento paralelo con los sistemas legados que están siendo sustituidos por nuestro proyecto.
- Conversión de las bases de datos operacionales.
- Entrenamiento de los usuarios y técnicos de mantenimiento.
- Traspaso del producto a los equipos de marketing, distribución y venta.

Los principales objetivos de esta fase son:

- Conseguir que el usuario se valga por si mismo.
- Un producto final que cumpla los requisitos esperados, que funcione y satisfaga suficientemente al usuario.

Los resultados de la fase de transición son [RSC98]:

- Prototipo Operacional
- Documentos Legales
- Caso del Negocio Completo

- Línea de Base del Producto completa y corregida que incluye todos los modelos del sistema
- Descripción de la Arquitectura completa y corregida
- Las iteraciones de esta fase irán dirigidas normalmente a conseguir una nueva versión.

Los criterios de evaluación de esta fase son los siguientes:

- El usuario se encuentra satisfecho.
- Son aceptables los gastos actuales versus los gastos planificados.

Estructura Estática del proceso. Roles, actividades, artefactos y flujos de trabajo

Un proceso de desarrollo de software define quién hace qué, cómo y cuándo. RUP define cuatro elementos los roles, que responden a la pregunta ¿Quién?, las actividades que responden a la pregunta ¿Cómo?, los productos, que responden a la pregunta ¿Qué? y los flujos de trabajo de las disciplinas que responde a la pregunta ¿Cuándo? (ver Figura 12) [RSC98].



Figura 12: Relación entre roles, actividades, artefactos

Roles

Un rol define el comportamiento y responsabilidades de un individuo, o de un grupo de individuos trabajando juntos como un equipo. Una persona puede desempeñar diversos roles, así como un mismo rol puede ser representado por varias personas.

Las responsabilidades de un rol son tanto el llevar a cabo un conjunto de actividades como el ser el dueño de un conjunto de artefactos [MMA].

RUP define grupos de roles, agrupados por participación en actividades relacionadas. Estos grupos son [RSC02]:

Analistas:

- Analista de procesos de negocio.
- Diseñador del negocio.
- Analista de sistema.
- Especificador de requisitos.

Desarrolladores:

- Arquitecto de software.
- Diseñador.
- Diseñador de interfaz de usuario.
- Diseñador de cápsulas.
- Diseñador de base de datos.
- Implementador.
- Integrador.

Gestores:

- Jefe de proyecto.
- Jefe de control de cambios.

- Jefe de configuración.
- Jefe de pruebas.
- Jefe de despliegue.
- Ingeniero de procesos.
- Revisor de gestión del proyecto.
- Gestor de pruebas.

Apoyo:

- Documentador técnico.
- Administrador de sistema.
- Especialista en herramientas.
- Desarrollador de cursos.
- Artista gráfico.

Especialista en pruebas:

- Especialista en Pruebas (*tester*).
- Analista de pruebas.
- Diseñador de pruebas.

Otros roles:

- *Stakeholders*.
- Revisor.
- Coordinación de revisiones.
- Revisor técnico.
- Cualquier rol.

Actividades

Una actividad en concreto es una unidad de trabajo que una persona que desempeñe un rol puede ser solicitado a que realice. Las actividades tienen un objetivo concreto, normalmente expresado en términos de crear o actualizar algún producto.

Artefactos

Un producto o artefacto es un trozo de información que es producido, modificado o usado durante el proceso de desarrollo de software. Los productos son los resultados tangibles del proyecto, las cosas que va creando y usando hasta obtener el producto final [MMA].

Un artefacto puede ser cualquiera de los siguientes [RSC02]:

- Un documento, como el documento de la arquitectura del software.
- Un modelo, como el modelo de Casos de Uso o el modelo de diseño.
- Un elemento del modelo, un elemento que pertenece a un modelo como una clase, un Caso de Uso o un subsistema.

Flujos de trabajo

Con la enumeración de roles, actividades y artefactos no se define un proceso, necesitamos contar con una secuencia de actividades realizadas por los diferentes roles, así como la relación entre los mismos. Un flujo de trabajo es una relación de actividades que nos producen unos resultados observables. A continuación se dará una explicación de cada flujo de trabajo.

Modelado del negocio

Con este flujo de trabajo pretendemos llegar a un mejor entendimiento de la organización donde se va a implantar el producto.

Los objetivos del modelado de negocio son [RSC02]:

- Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado (organización objetivo).

- Entender el problema actual en la organización objetivo e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.
- Derivar los requisitos del sistema necesarios para apoyar a la organización objetivo.

Para lograr estos objetivos, el modelo de negocio describe como desarrollar una visión de la nueva organización, basado en esta visión se definen procesos, roles y responsabilidades de la organización por medio de un modelo de Casos de Uso del negocio y un Modelo de Objetos del Negocio. Complementario a estos modelos, se desarrollan otras especificaciones tales como un Glosario.

Requisitos

Este es uno de los flujos de trabajo más importantes, porque en él se establece qué tiene que hacer exactamente el sistema que construyamos. En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

Los objetivos del flujo de datos Requisitos es [RSC02]:

- Establecer y mantener un acuerdo entre clientes y otros *stakeholders* sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para la planeación de los contenidos técnicos de las iteraciones.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.
- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

Los requisitos se dividen en dos grupos. Los requisitos funcionales representan la funcionalidad del sistema. Se modelan mediante diagramas de Casos de Uso. Los requisitos no funcionales representan aquellos atributos que debe exhibir el sistema, pero que no son una funcionalidad específica. Por ejemplo: requisitos de facilidad de uso, fiabilidad, eficiencia, portabilidad, etc.

Para capturar los requisitos es preciso entrevistar a todos los interesados en el proyecto, no sólo a los usuarios finales, y anotar todas sus peticiones. A partir de ellas hay que descubrir lo que necesitan y expresarlo en forma de requisitos.

En este flujo de trabajo, y como parte de los requisitos de facilidad de uso, se diseña la interfaz gráfica de usuario. Para ello habitualmente se construyen prototipos de la interfaz gráfica de usuario que se contrastan con el usuario final.

Análisis y Diseño

El objetivo de este flujo de trabajo es traducir los requisitos a una especificación que describe cómo implementar el sistema.

Los objetivos del análisis y diseño son [RSC02]:

- Transformar los requisitos al diseño del futuro sistema.
- Desarrollar una arquitectura para el sistema.
- Adaptar el diseño para que sea consistente con el entorno de implementación, diseñando para el rendimiento.

El análisis consiste en obtener una visión del sistema que se preocupa de ver qué hace, de modo que sólo se interesa por los requisitos funcionales. Por otro lado, el diseño es un refinamiento del análisis que tiene en cuenta los requisitos no funcionales, en definitiva cómo cumple el sistema sus objetivos.

Al principio de la fase de elaboración hay que definir una arquitectura candidata: crear un esquema inicial de la arquitectura del sistema, identificar clases de análisis y actualizar las realizaciones de los Casos de Uso con las interacciones de las clases de análisis. Durante la fase de elaboración se va refinando esta arquitectura hasta llegar a su forma definitiva.

En cada iteración hay que analizar el comportamiento para diseñar componentes. Además si el sistema usará una base de datos, habrá que diseñarla también, obteniendo un modelo de datos.

El resultado final más importante de este flujo de trabajo será el modelo de diseño. Consiste en colaboraciones de clases, que pueden ser agregadas en paquetes y subsistemas.

Otro producto importante de este flujo es la documentación de la arquitectura de software, que captura varias vistas arquitectónicas del sistema.

Implementación

En este flujo de trabajo se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. Además se deben hacer las pruebas de unidad: cada implementador es responsable de probar las unidades que produzca. El resultado final de este flujo de trabajo es un sistema ejecutable.

En cada iteración habrá que hacer lo siguiente:

- Planificar qué subsistemas deben ser implementados y en que orden deben ser integrados, formando el Plan de Integración.
- Cada implementador decide en que orden implementa los elementos del subsistema.
- Si encuentra errores de diseño, los notifica.
- Se prueban los subsistemas individualmente.
- Se integra el sistema siguiendo el plan.

La estructura de todos los elementos implementados forma el modelo de implementación. La integración debe ser incremental, es decir, en cada momento sólo se añade un elemento. De este modo es más fácil localizar fallos y los componentes se prueban más a fondo. En fases tempranas del proceso se pueden implementar prototipos para reducir el riesgo. Su utilidad puede ir desde ver si el sistema es viable desde el principio, probar tecnologías o diseñar la interfaz de usuario.

Los prototipos pueden ser exploratorios (desechables) o evolutivos. Estos últimos llegan a transformarse en el sistema final.

Pruebas

Este flujo de trabajo es el encargado de evaluar la calidad del producto que estamos desarrollando, pero no para aceptar o rechazar el producto al final del proceso de desarrollo, sino que debe ir integrado en todo el ciclo de vida.

Esta disciplina brinda soporte a las otras disciplinas. Sus objetivos son [RSC02]:

- Encontrar y documentar defectos en la calidad del software.
- Generalmente asesora sobre la calidad del software percibida.
- Provee la validación de los supuestos realizados en el diseño y especificación de requisitos por medio de demostraciones concretas.
- Verificar las funciones del producto de software según lo diseñado.
- Verificar que los requisitos tengan su apropiada implementación.

Las actividades de este flujo comienzan pronto en el proyecto con el plan de prueba (el cual contiene información sobre los objetivos generales y específicos de las pruebas en el proyecto, así como las estrategias y recursos con que se dotará a esta tarea), o incluso antes con alguna evaluación durante la fase de inicio, y continuará durante todo el proyecto.

El desarrollo del flujo de trabajo consistirá en planificar que es lo que hay que probar, diseñar cómo se va a hacer, implementar lo necesario para llevarlos a cabo, ejecutarlos en los niveles necesarios y obtener los resultados, de forma que la información obtenida nos sirva para ir refinando el producto a desarrollar.

Despliegue

El objetivo de este flujo de trabajo es producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

- Probar el producto en su entorno de ejecución final.

- Empaquetar el software para su distribución.
- Distribuir el software.
- Instalar el software.
- Proveer asistencia y ayuda a los usuarios.
- Formar a los usuarios y al cuerpo de ventas.
- Migrar el software existente o convertir bases de datos.

Este flujo de trabajo se desarrolla con mayor intensidad en la fase de transición, ya que el propósito del flujo es asegurar una aceptación y adaptación sin complicaciones del software por parte de los usuarios. Su ejecución inicia en fases anteriores, para preparar el camino, sobre todo con actividades de planificación, en la elaboración del manual de usuario y tutoriales.

Gestión del proyecto

La Gestión del proyecto es el arte de lograr un balance al gestionar objetivos, riesgos y restricciones para desarrollar un producto que sea acorde a los requisitos de los clientes y los usuarios.

Los objetivos de este flujo de trabajo son:

- Proveer un marco de trabajo para la gestión de proyectos de software intensivos.
- Proveer guías prácticas realizar planeación, contratar personal, ejecutar y monitorear el proyecto.
- Proveer un marco de trabajo para gestionar riesgos.

La planeación de un proyecto posee dos niveles de abstracción: un plan para las fases y un plan para cada iteración.

Configuración y control de cambios

La finalidad de este flujo de trabajo es mantener la integridad de todos los artefactos que se crean en el proceso, así como de mantener información del proceso evolutivo que han seguido.

Entorno

La finalidad de este flujo de trabajo es dar soporte al proyecto con las adecuadas herramientas, procesos y métodos. Brinda una especificación de las herramientas que se van a necesitar en cada momento, así como definir la instancia concreta del proceso que se va a seguir.

En concreto las responsabilidades de este flujo de trabajo incluyen:

- Selección y adquisición de herramientas
- Establecer y configurar las herramientas para que se ajusten a la organización.
- Configuración del proceso.
- Mejora del proceso.
- Servicios técnicos.

El principal artefacto que se usa en este flujo de trabajo es el *caso de desarrollo* que especifica para el proyecto actual en concreto, como se aplicará el proceso, que productos se van a utilizar y como van a ser utilizados. Además se tendrán que definir las guías para los distintos aspectos del proceso, como pueden ser el modelado del negocio y los Casos de Uso, para la interfaz de usuario, el diseño, la programación, el manual de usuario.

3.1 Lista de los Actores

Los actores que interactúan en el sistema son:

- a) Actor Administrador
- b) Actor Director.
- c) Actor Secretaria.
- d) Actor Docente.
- e) Actor Regente.
- f) Actor Estudiante
- g) Actor Tutor.

a) Actor Administrador

Gestionara la información académica del establecimiento educativo, para que esta pueda llegar a los diferentes usuarios del sistema como:

- Altas y bajas a docentes.
- Administrará materias (Adm.).
- Administrará opciones.
- Habilitará fechas para calificar.
- Cambiará claves.

b) Actor Director

- Asignará materias a cursos.
- Realizará la administración de paralelos.
- Asignará materias a docentes.
- Consultará reportes de estudiantes.
- Consultará la lista de todos los docentes y estudiantes por paralelo.

- Realizará altas y bajas al personal administrativo.
- Podrá cambiar gestión e iniciar nueva gestión.

c) Actor Secretaria

- Registrará a los estudiantes, docentes, tutores y administrativos dentro del sistema.
- Actualizará datos personales de los usuarios registrados en el sistema.
- Tendrá a su cargo realizar las inscripciones de los estudiantes.
- Consultará reportes de estudiantes y docentes.
- Consultará la lista de todos los docentes y la de estudiantes por paralelo.

d) Actor Docente

- Administrará notas de sus estudiantes
- Consultará reportes y lista de estudiantes correspondiente al paralelo según su (s) materia (s).

e) Actor Regente

- Registrará la asistencia de los estudiantes.

f) Actor Estudiante

- Consultar los reportes de sus notas y asistencia, para su verificación correspondiente. También podrá consultar la lista de sus compañeros.

g) Actor Tutor

- Consultar los reportes de notas y asistencia de todos los estudiantes que se encuentran a su cargo.

3.2 Modelado del Contexto del Sistema

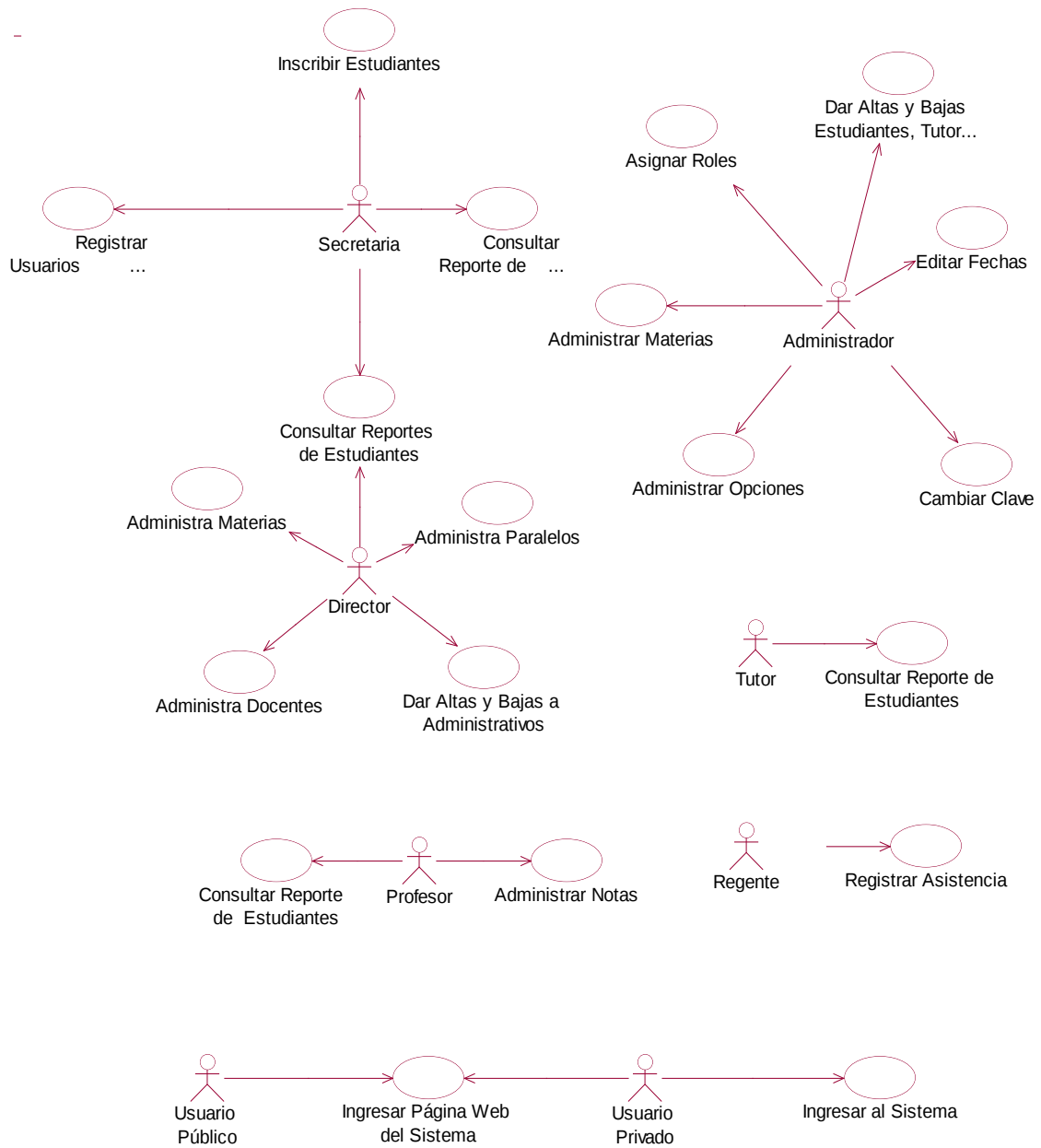


Figura 1: Diagrama de Contexto.

3.3 Lista de Diagramas de casos de Uso

Diagramas de Casos de Uso Para el Administrador

Ingresar al Sistema

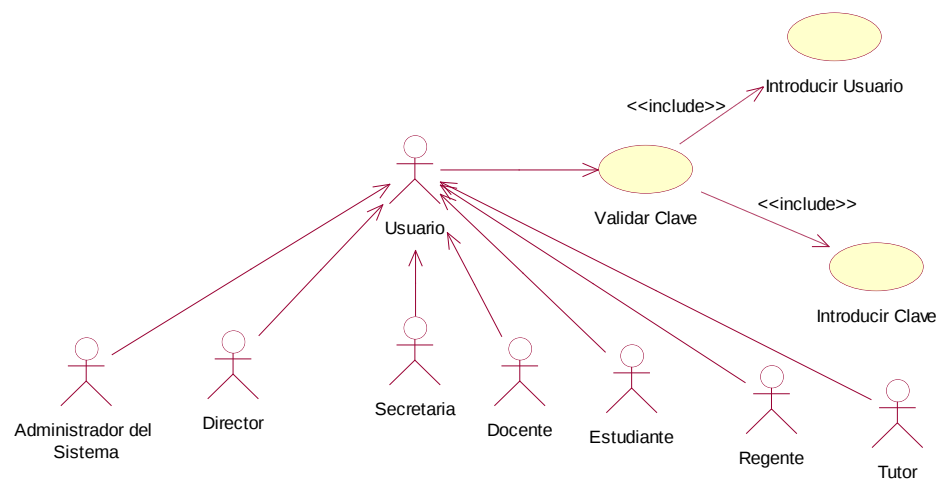


Figura 2: Diagrama de Caso de Uso: Ingresar al Sistema

Altas y Bajas a Docentes

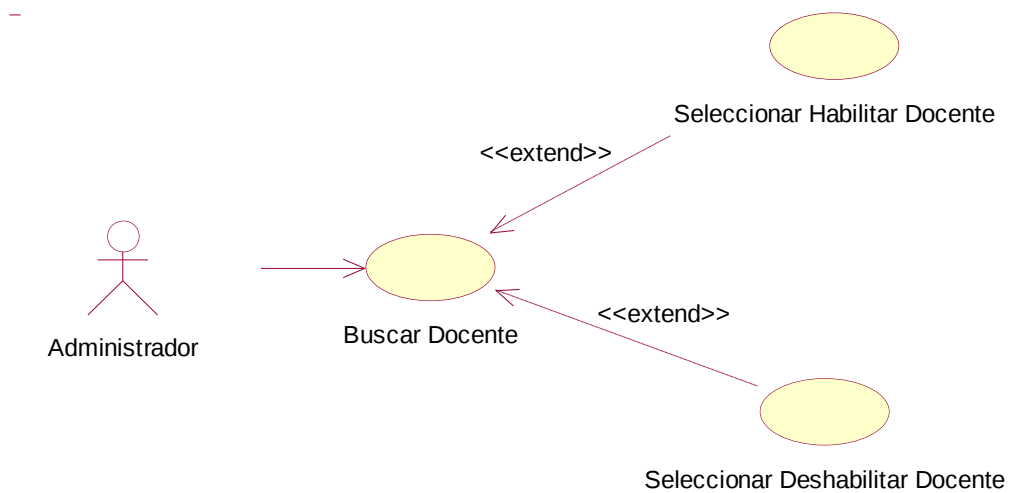


Figura 2: Diagrama de Caso de Uso: Altas y Bajas a Docentes

Administrar Materias (ADM)

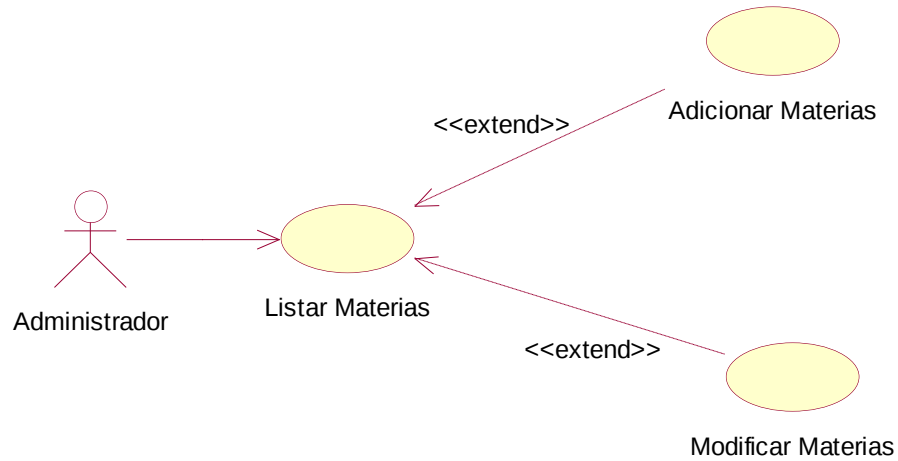


Figura 4: Diagrama de Caso de Uso: Administrar Materias (ADM.)

Administrar Opciones

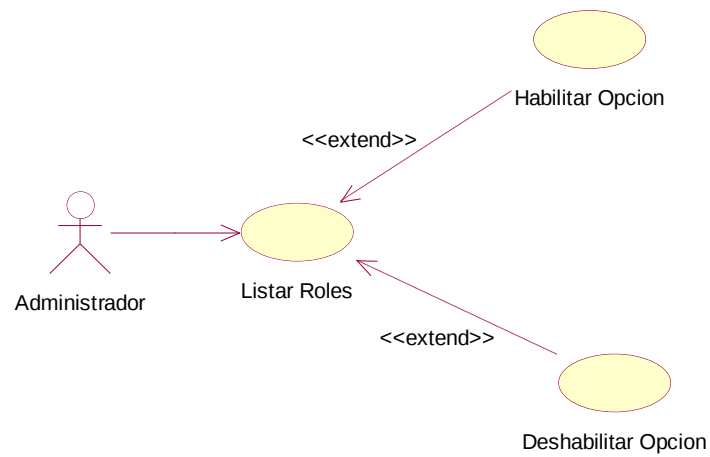


Figura 4: Diagrama de Caso de Uso: Administrar Opciones

Fechas para Calificar

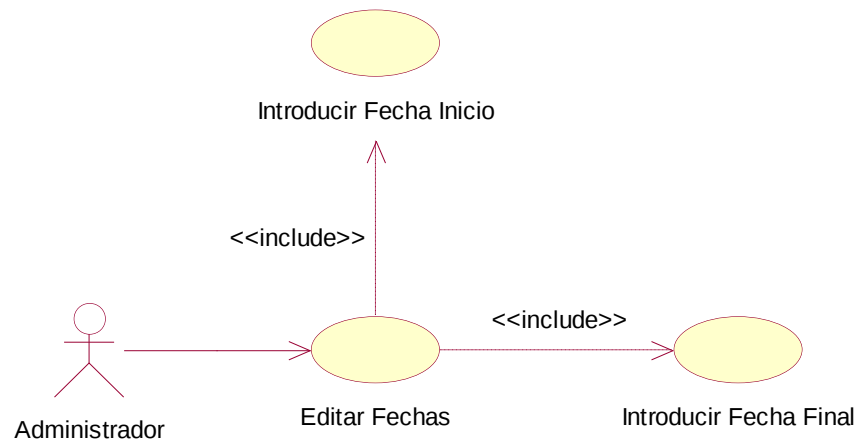


Figura 6: Diagrama de Caso de Uso: Fechas para Calificar

Administrar Clave

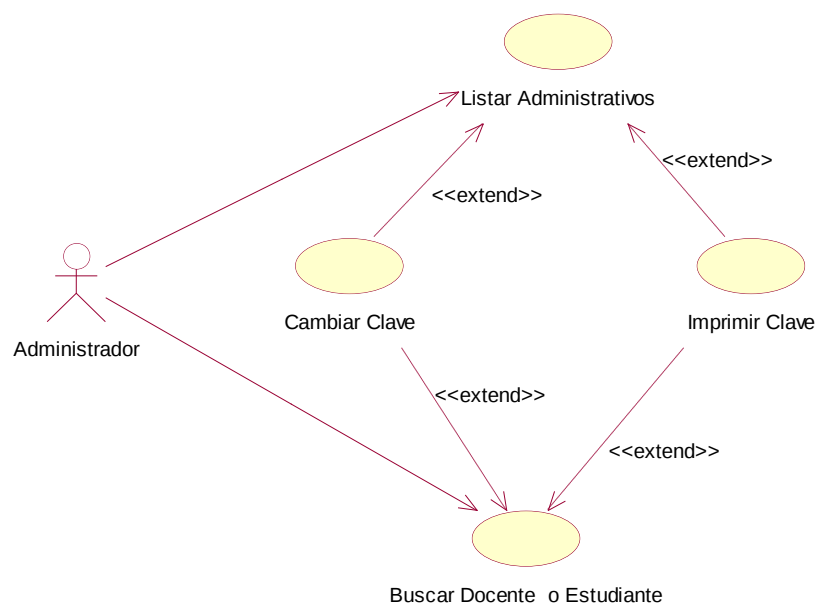


Figura 7: Diagrama de caso de uso:Administrar Clave

Diagramas de Casos de Uso Para el Director

Administrar Materias (DIR)

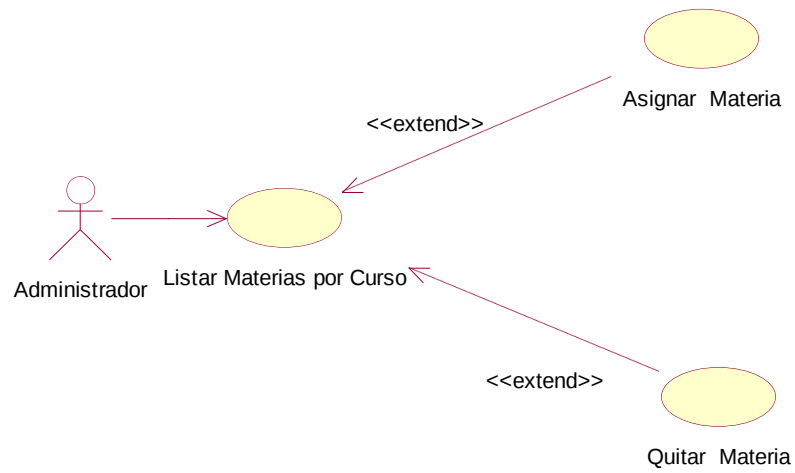


Figura 8: Diagrama de Caso de Uso: Administrar Materias

Administrar Paralelo

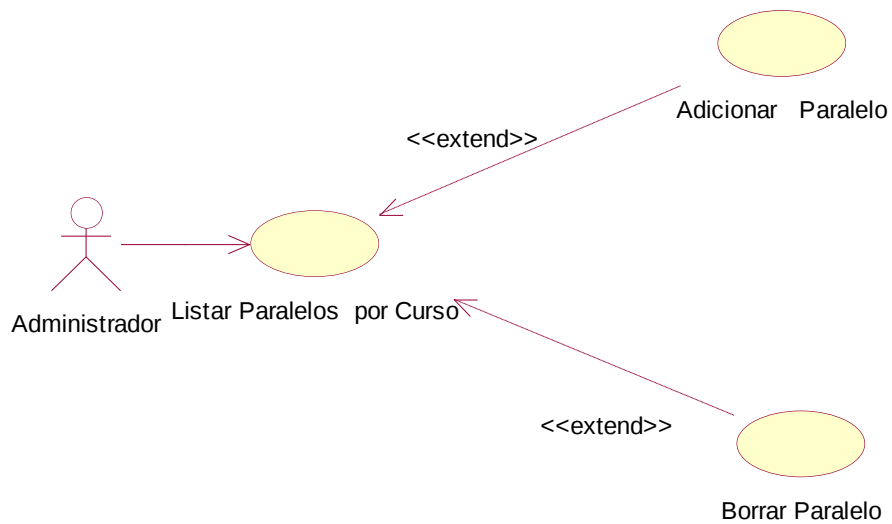


Figura 9: Diagrama de Caso de Uso: Administrar Paralelo

Administrar Docentes

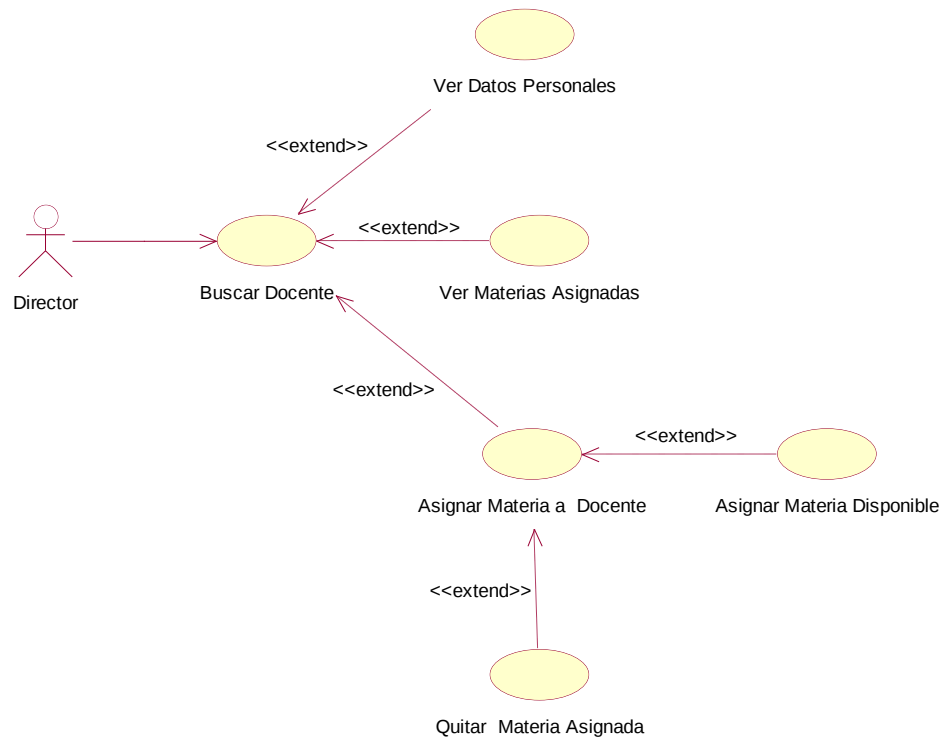


Figura 10: Diagrama de Caso de Uso: Administrar Docentes

Altas y Bajas a Administrativos

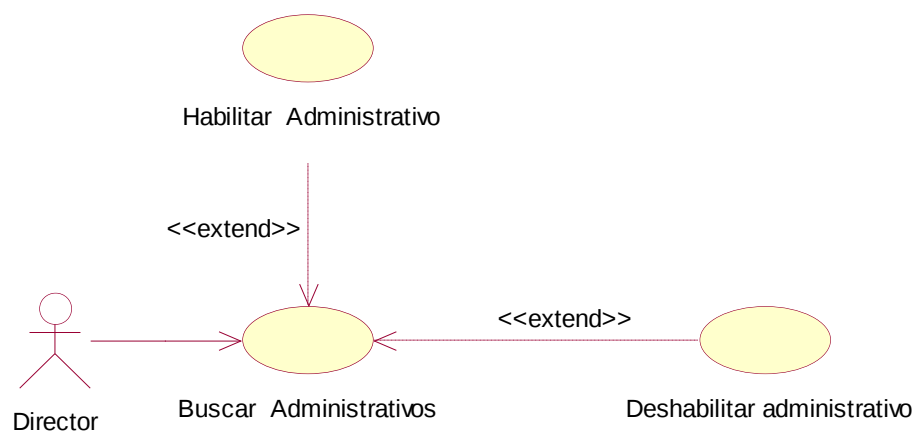


Figura 11: Diagrama de Caso de Uso: Habilitar y Deshabilitar Administrativos

Reporte de Estudiantes

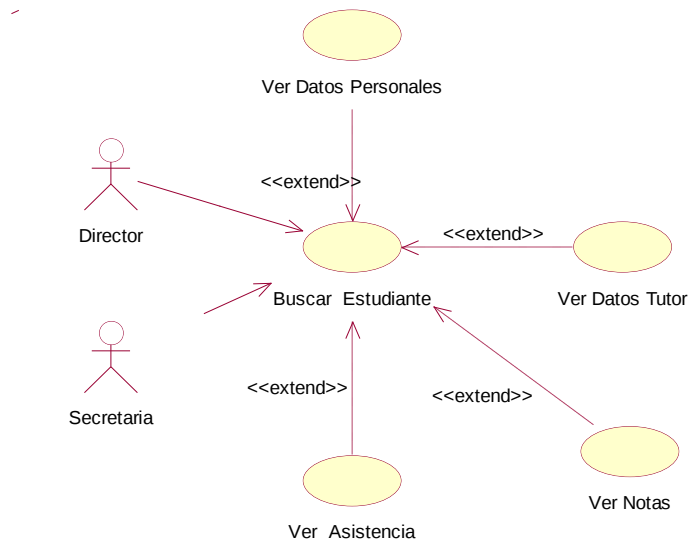


Figura 12: Diagrama de Caso de Uso: Habilitar y Deshabilitar Administrativos

Cambiar Gestión (Director y Secretaria)

Iniciar Nueva Gestión

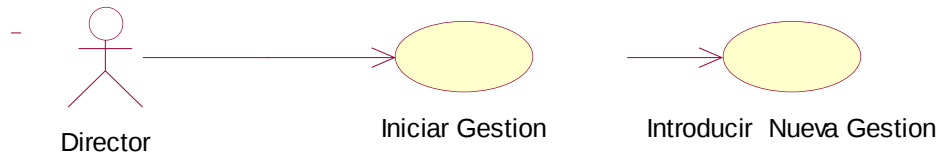


Figura 13: Diagrama de Caso de Uso: Iniciar Nueva Gestión

Diagramas de Casos de Uso Para la Secretaria

Registrar Administrativos

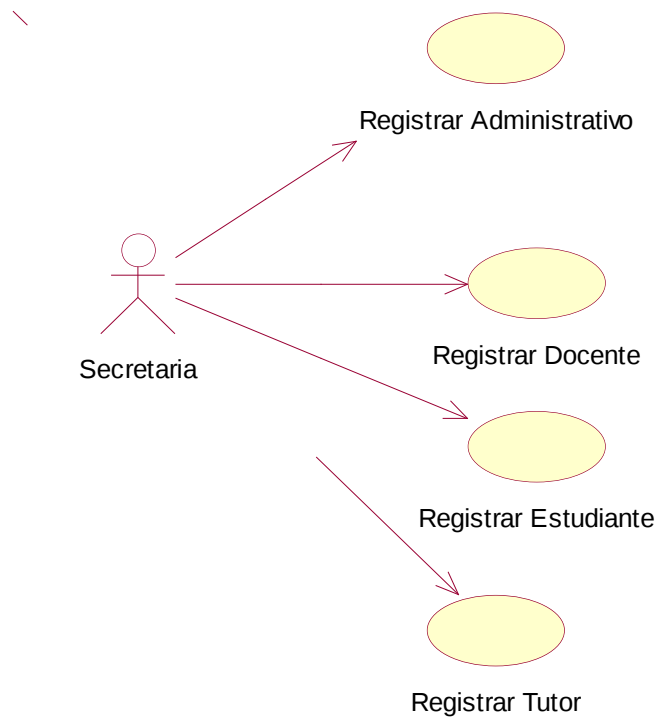


Figura 14: Diagrama de Caso de Uso: Iniciar Nueva Gestión

Registrar Docentes

Registrar Tutores

Registrar Estudiantes

Modificar Datos Personales

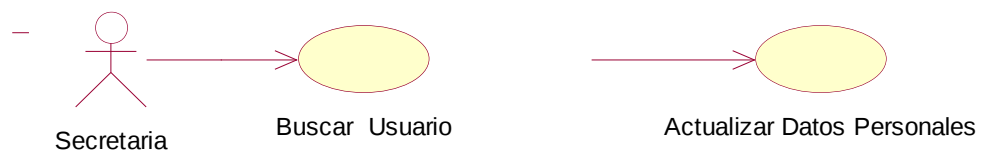


Figura 15: Diagrama de Caso de Uso Modificar Datos Personales

Inscribir Estudiante

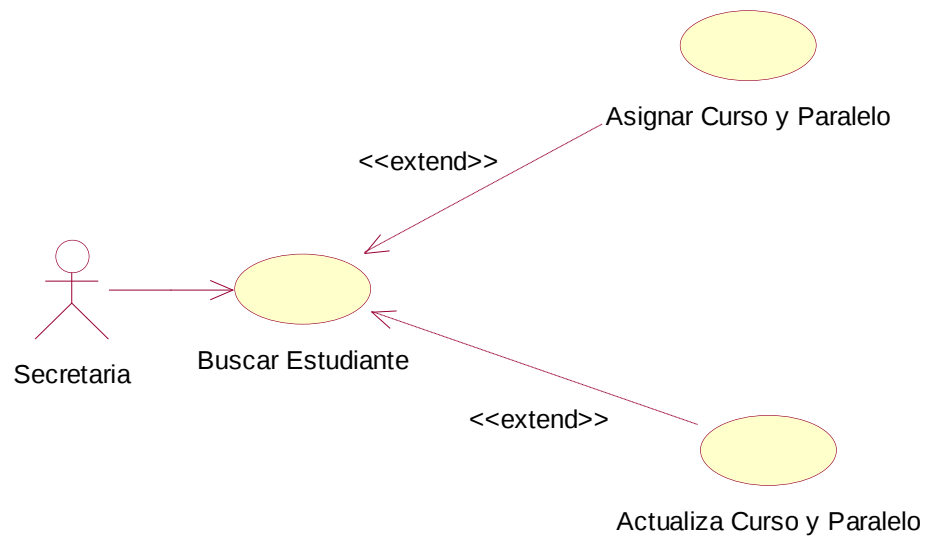


Figura 16: Diagrama de Caso de Uso: Inscribir Estudiante

Reporte de Docentes

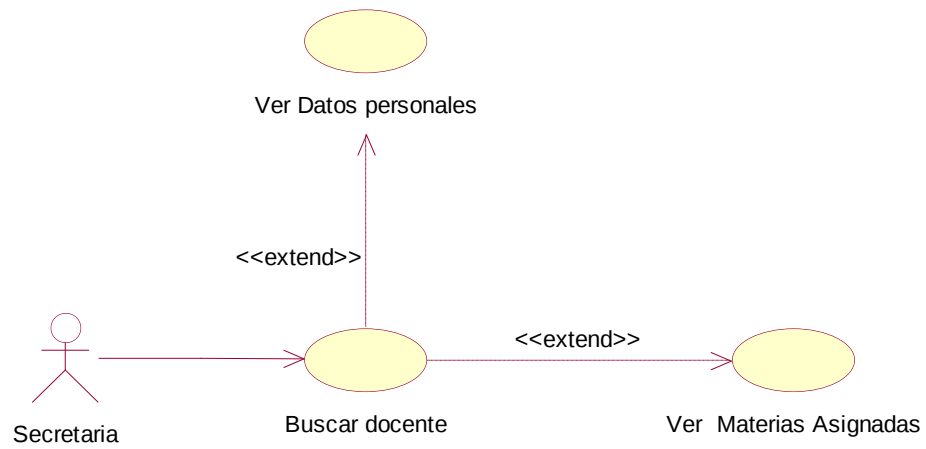


Figura 17: Diagrama de Caso de Uso: Reporte de Docentes

Reporte de Estudiantes

Diagramas de Casos de Uso Para el Docente

Administrar Notas

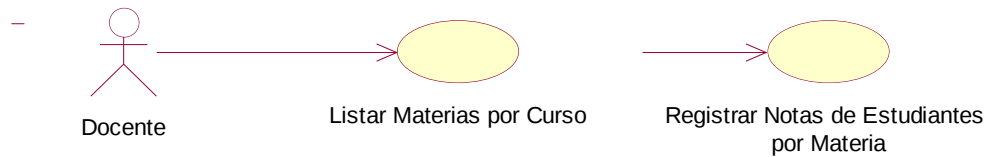


Figura 18: Diagrama de Caso de Uso: Administrar Notas

Reporte de Estudiantes para Docente

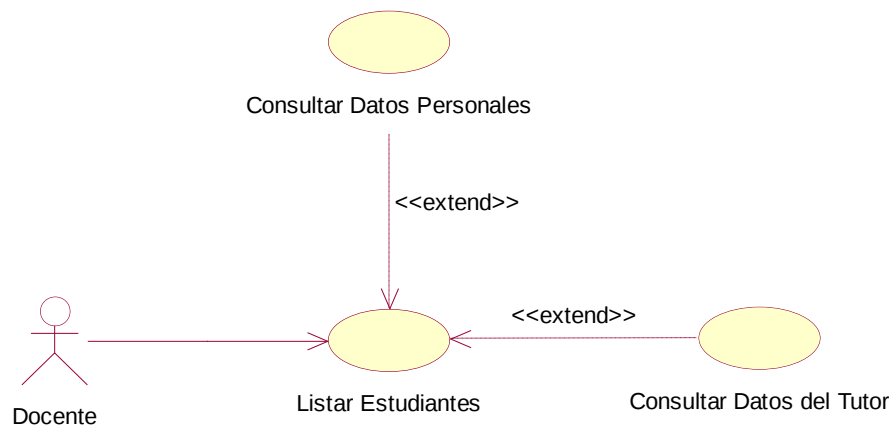


Figura 19: Diagrama de Caso de Uso: Reporte de Estudiantes para Docente

Reporte de Estudiantes para Estudiante

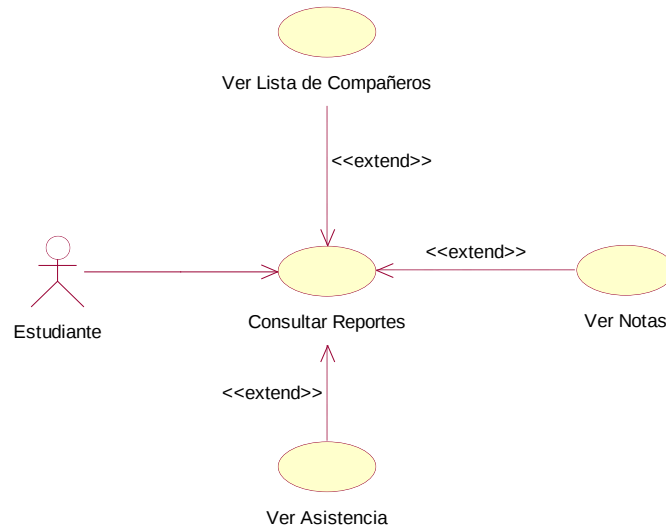


Figura 20: Diagrama de Caso de Uso: de Reporte de Estudiantes para Estudiantes

Diagramas de Casos de Uso Para Regente

Administrar Asistencia

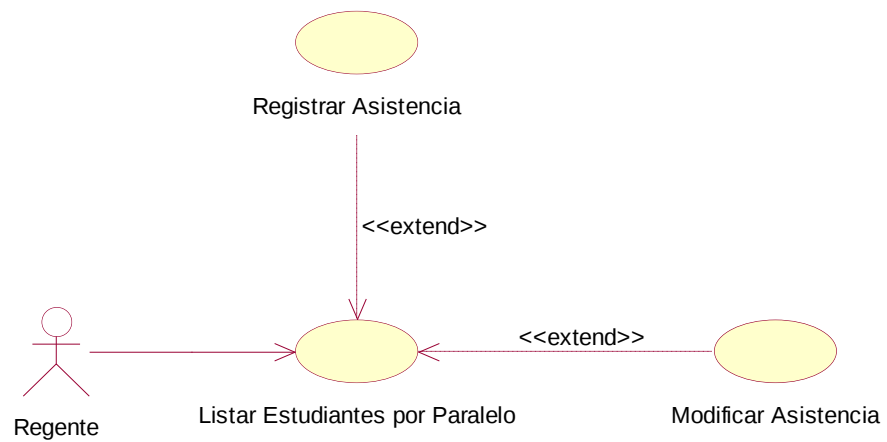


Figura 21: Diagrama de Caso de Uso: Administrar Asistencia

Diagramas de Casos de Uso Para el Tutor

Reporte de Estudiantes para Tutor

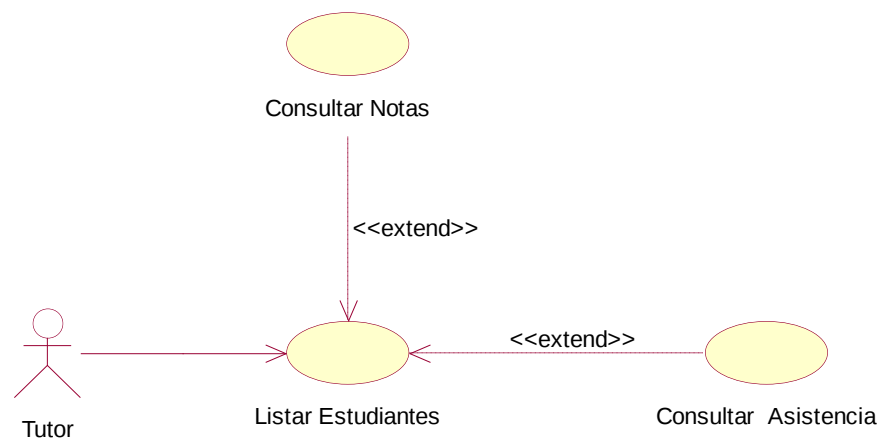


Figura 22: Diagrama de Caso de Uso: Reporte de Estudiantes para Tutor

Diagramas de Casos de Uso Para el Estudiante

Ver Notas

Ver Lista de Paralelo

Ver Asistencia

Ver Historial

3.4 Descripción de los Casos de Uso

Nombre	Ingresa a la página Web del Sistema
Objetivo	Mostrar información de la Unidad Educativa
Descripción	Al ingresar, el actor podrá acceder a un menú libre que el establecimiento educativo brinda. En este menú podrá acceder a la opción historia, instalaciones, misión y visión de la Unidad Educativa.
Actor	Usuario privado

Nombre	Ingresa al Sistema
Objetivo	Permitir ingresar al sistema
Descripción	El actor introducirá al sistema su login y clave los cuales serán validados para acceder a la información disponible para cada actor.
Actor	Director, Secretaria, Profesor, Estudiante ó Tutor y Administrador

Nombre	Dar altas y bajas a estudiantes, tutores y docentes
Objetivo	Permitir habilitar o deshabilitar a estudiante, tutor y docente
Descripción	El actor seleccionará a estudiante, tutor o docente una lista que resultó de la búsqueda de algún nombre introducido.
Actor	Administrador

Nombre	Administrar materias administrador
Objetivo	Permitir gestionar materias para los diferentes cursos de la unidad educativa.
Descripción	El actor podrá registrar, modificar materias para los diferentes cursos.
Actor	Administrador

Nombre	Administrar opciones
Objetivo	Habilitar y deshabilitar opciones

Descripción	El actor seleccionará rol de una lista y podrá habilitar o deshabilitar las opciones disponibles y actuales respectivamente de cada rol
Actor	Administrador

Nombre	Asignar roles
Objetivo	Permitir habilitar o deshabilitar roles
Descripción	El actor seleccionará rol, para ver usuarios administrativos y docentes y seleccionar alguno para asignarle algún rol o deshabilitarlo de algún rol actual.
Actor	Administrador

Nombre	Editar fechas para calificar
Objetivo	Gestionar fechas
Descripción	El actor podrá seleccionar editar fechas de una lista de fechas para introducir fecha de inicio y fecha de fin
Actor	Administrador
Nombre	Cambiar clave
Objetivo	Permitir cambiar clave
Descripción	El actor seleccionará rol, para ver usuarios administrativos y docentes y seleccionar alguno para introducir clave nueva
Actor	Administrador

Nombre	Administrar materias director
---------------	-------------------------------

Objetivo	Asignar materias a cursos
Descripción	El actor seleccionará curso, para ver sus materias actuales y materias disponibles para deshabilitar alguna materia actual o asignar alguna materia disponible al curso seleccionado.
Actor	Director

Nombre	Administrar Paralelos
Objetivo	Asignar paralelos a cursos
Descripción	El actor seleccionará paralelos de lista de paralelos según el curso para adicionar paralelos al curso
Actor	Director

Nombre	Administrar docentes
Objetivo	Consultar datos de docentes y asignar una o mas materias a los mismos
Descripción	El actor seleccionará docente de una lista que resultó de la búsqueda por nombres y podrá consultar datos personales, materias asignadas a ese docente y también asignar materias disponibles o deshabilitar materias que tiene actualmente asignadas el docente
Actor	Director

Nombre	Reporte de estudiantes (director/secretaria)
Objetivo	Consultar reportes de estudiantes
Descripción	El actor seleccionará estudiante de una lista que resultó de la búsqueda por nombres y podrá consultar datos personales, notas y asistencia del estudiante y datos del tutor
Actor	Director, secretaria

Nombre	Dar de altas y bajas administrativos
Objetivo	Permitir habilitar o deshabilitar administrativos
Descripción	El actor seleccionará administrativo de una lista que resultó de la búsqueda por nombres y podrá habilitar en caso de que se encuentre deshabilitado y deshabilitar si este habilitado
Actor	Director

Nombre	Registrar usuarios privados
Objetivo	Permitir registrar usuarios privados
Descripción	El actor registrará o actualizará en formularios respectivos datos personales de los usuarios privados
Actor	Secretaria

Nombre	Inscribir Estudiante
Objetivo	Habilitar estudiante dentro de algún curso y paralelo
Descripción	El estudiante solicitará la inscripción a la Unidad educativa, proporcionando los datos personales del estudiante, la secretaria verificará en sus registros si el estudiante es nuevo o antiguo, si el alumno es antiguo se actualizará su registro en la Base de datos, si el alumno es nuevo el actor introducirá curso y paralelo
Actor	Secretaria

Nombre	Reporte de docentes
Objetivo	Consultar reportes de docentes
Descripción	El actor seleccionará al docente de una lista que resultó de la búsqueda por nombres y podrá consultar datos personales y materias asignadas del docente
Actor	Secretaria

Nombre	Administrar notas.
Objetivo	Registrar notas de estudiantes
Descripción	El actor seleccionará a estudiante de una lista de estudiantes que resultó de seleccionar materia según curso y paralelos para introducir notas
Actor	Docente

Nombre	Reporte de estudiantes (docente)
---------------	----------------------------------

Objetivo	Consultar reporte del estudiantes
Descripción	El actor seleccionará a estudiante de una lista de materia según curso y paralelos y podrá consultar datos personales del estudiante y datos personales del tutor
Actor	Docente

Nombre	Reporte de estudiante(s)
Objetivo	Consultar reporte de estudiantes
Descripción	El actor podrá consultar notas, asistencia y compañeros
Actor	Estudiante

Nombre	Reporte de estudiantes (tutor)
Objetivo	Consultar reporte de estudiantes
Descripción	El actor seleccionará al estudiante de la lista de los estudiantes a su cargo y podrá consultar notas y asistencia de dicho estudiante
Actor	Tutor

Nombre	Registrar Asistencia
Objetivo	Tener un control de asistencia de estudiantes
Descripción	El actor seleccionará al estudiante de una lista de estudiantes que resultó de seleccionar una materia según el curso y paralelo para registrar las faltas, atrasos y permisos de cada

	uno de los estudiantes
Actor	Regente

4.5 Diagrama de Secuencias

Ingresar al Sistema

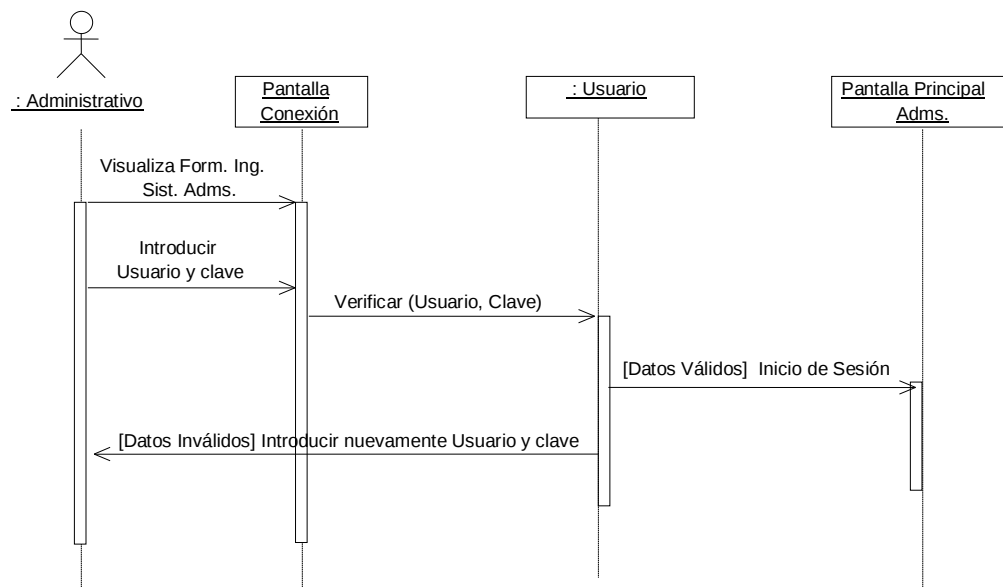


Figura 23: Diagrama de Secuencia: Ingresar al Sistema

Administrar Materia (ADM.)

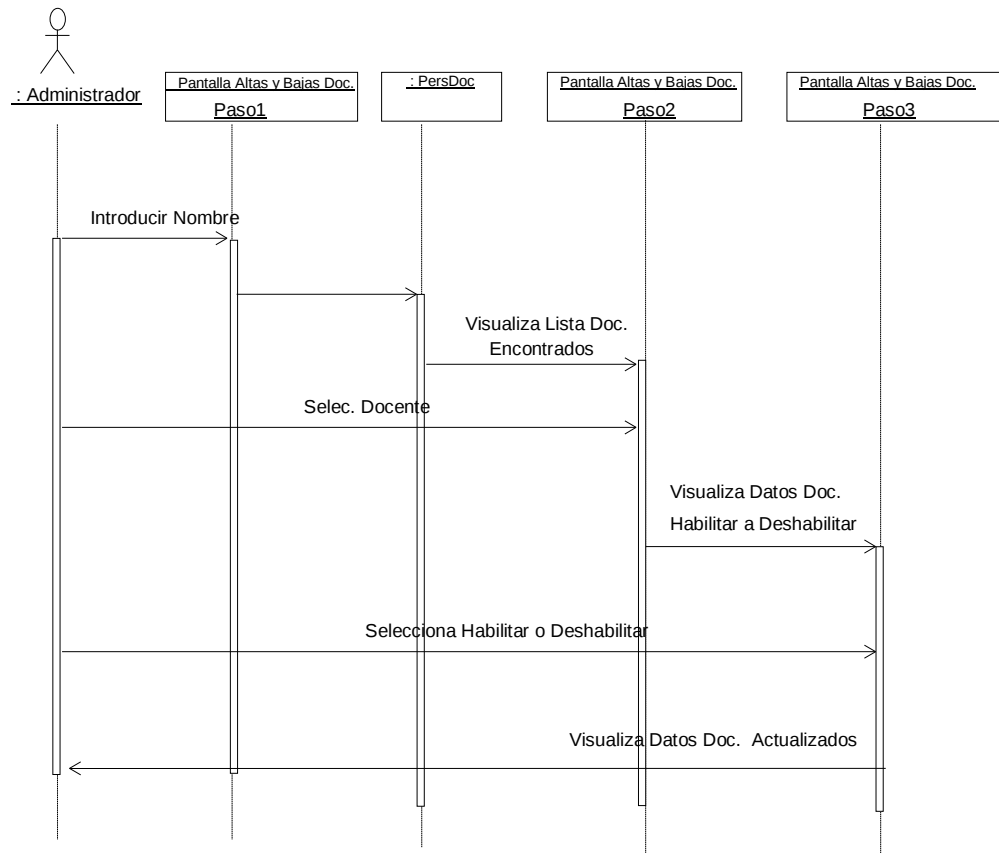


Figura 24: Diagrama de Secuencia: Administrar Materia (ADM)

Administrar Opciones

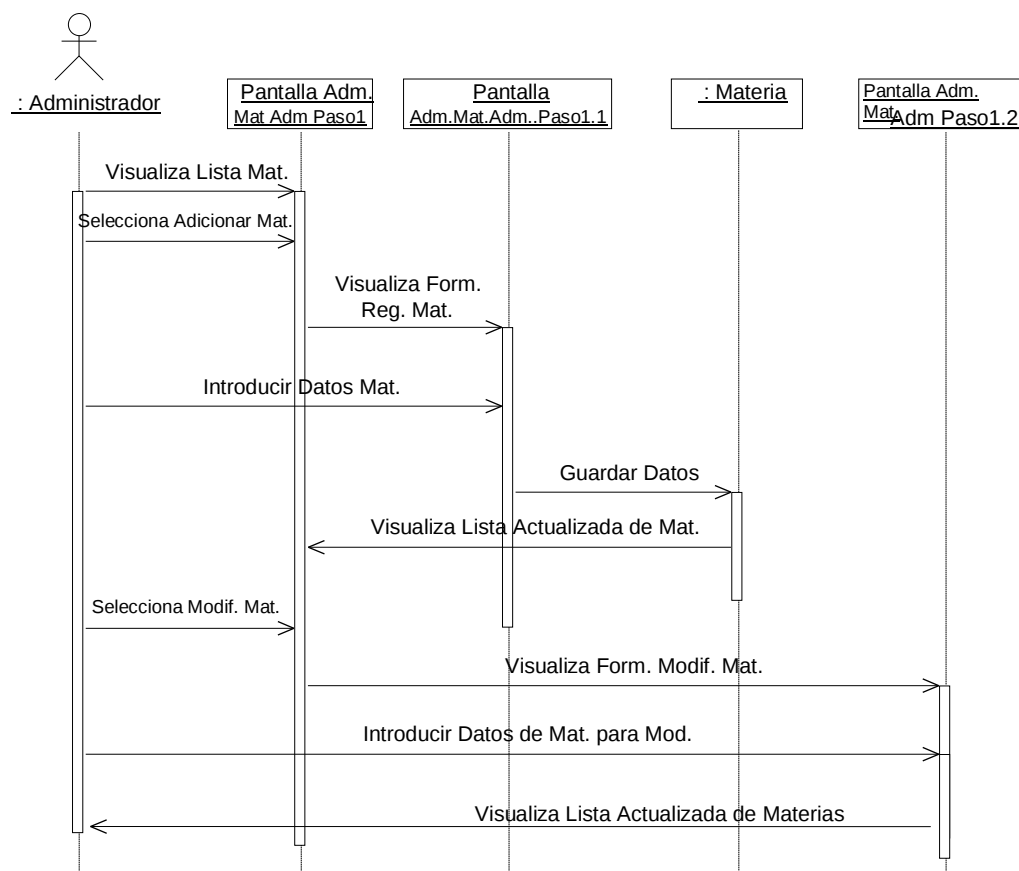


Figura 25: Diagrama de Secuencia: Ingresar al Sistema

Asignar Rol

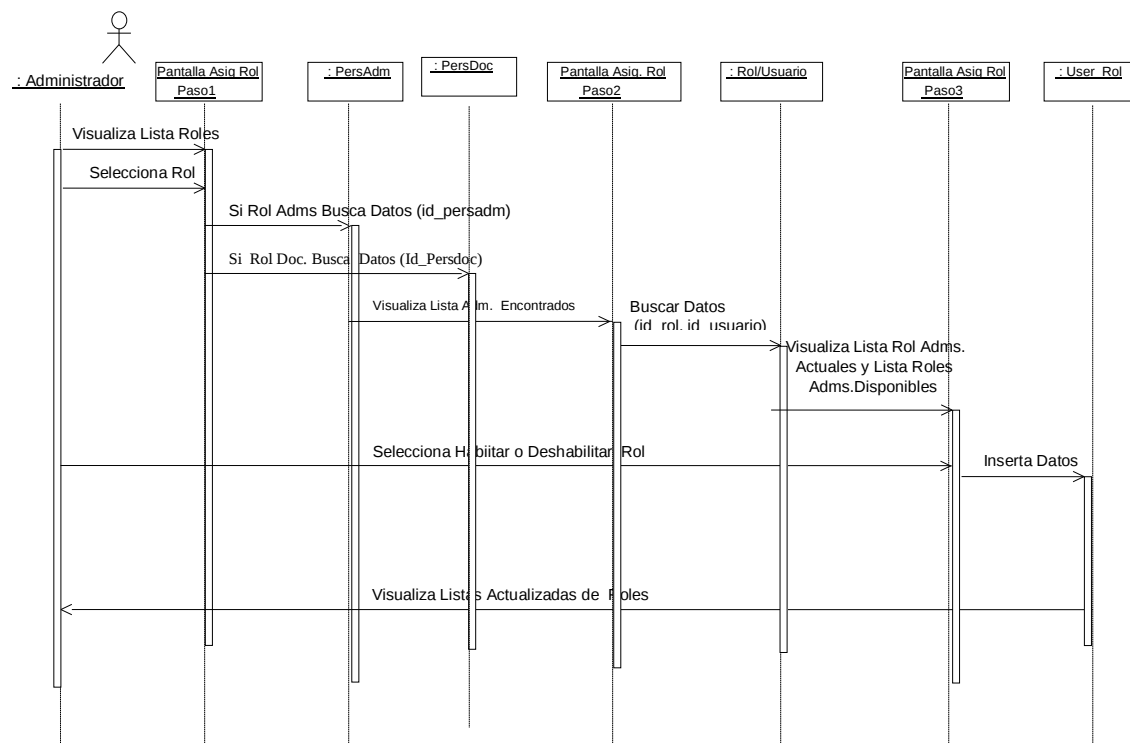


Figura 26: Diagrama de Secuencia: Asignar Rol

Editar Fechas para Calificar

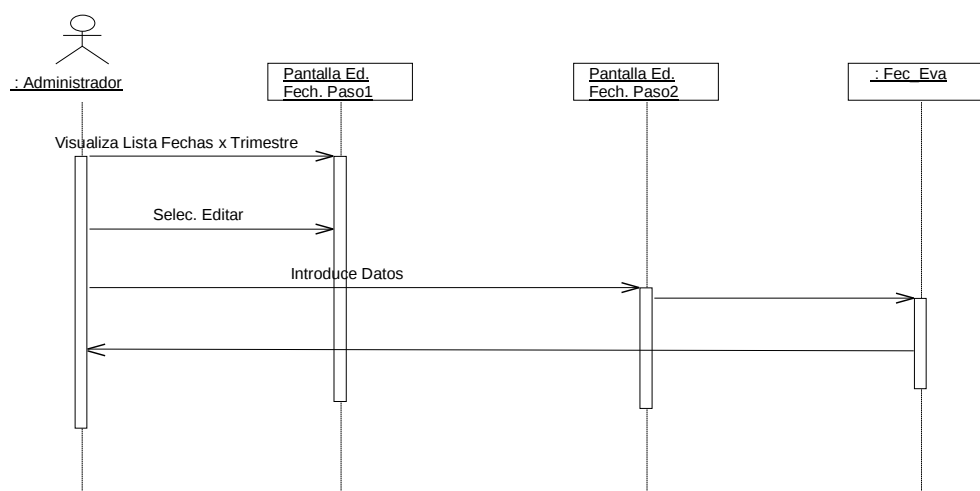


Figura 27: Diagrama de Secuencia: Editar Fechas para Calificar

Administrar Materias (DIRECTOR)

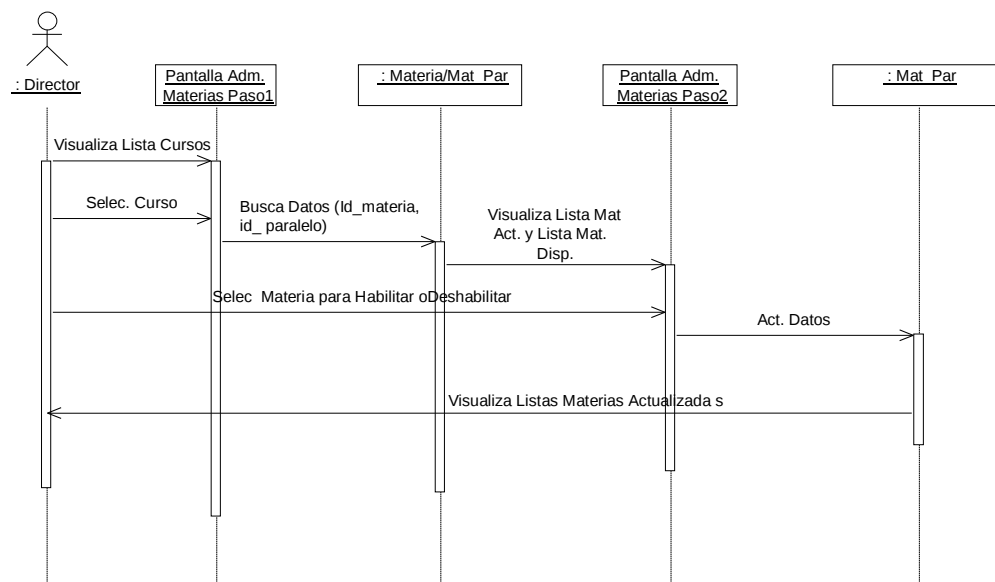


Figura 28: Diagrama de Secuencia: Administrar Materias (Director)

Administrar Paralelos

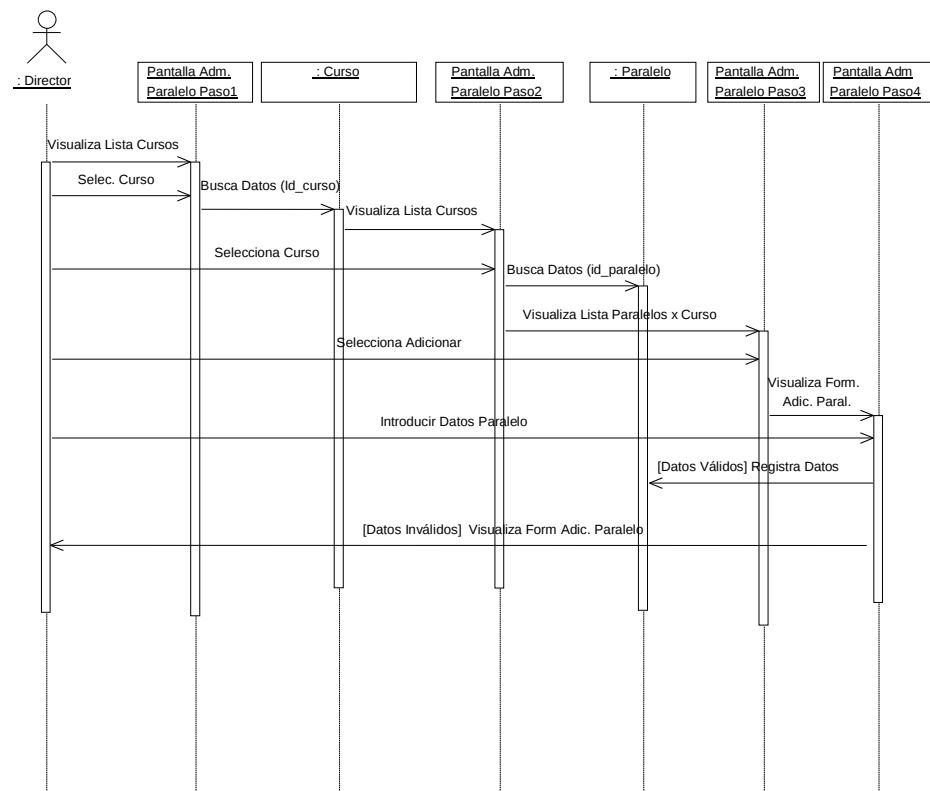
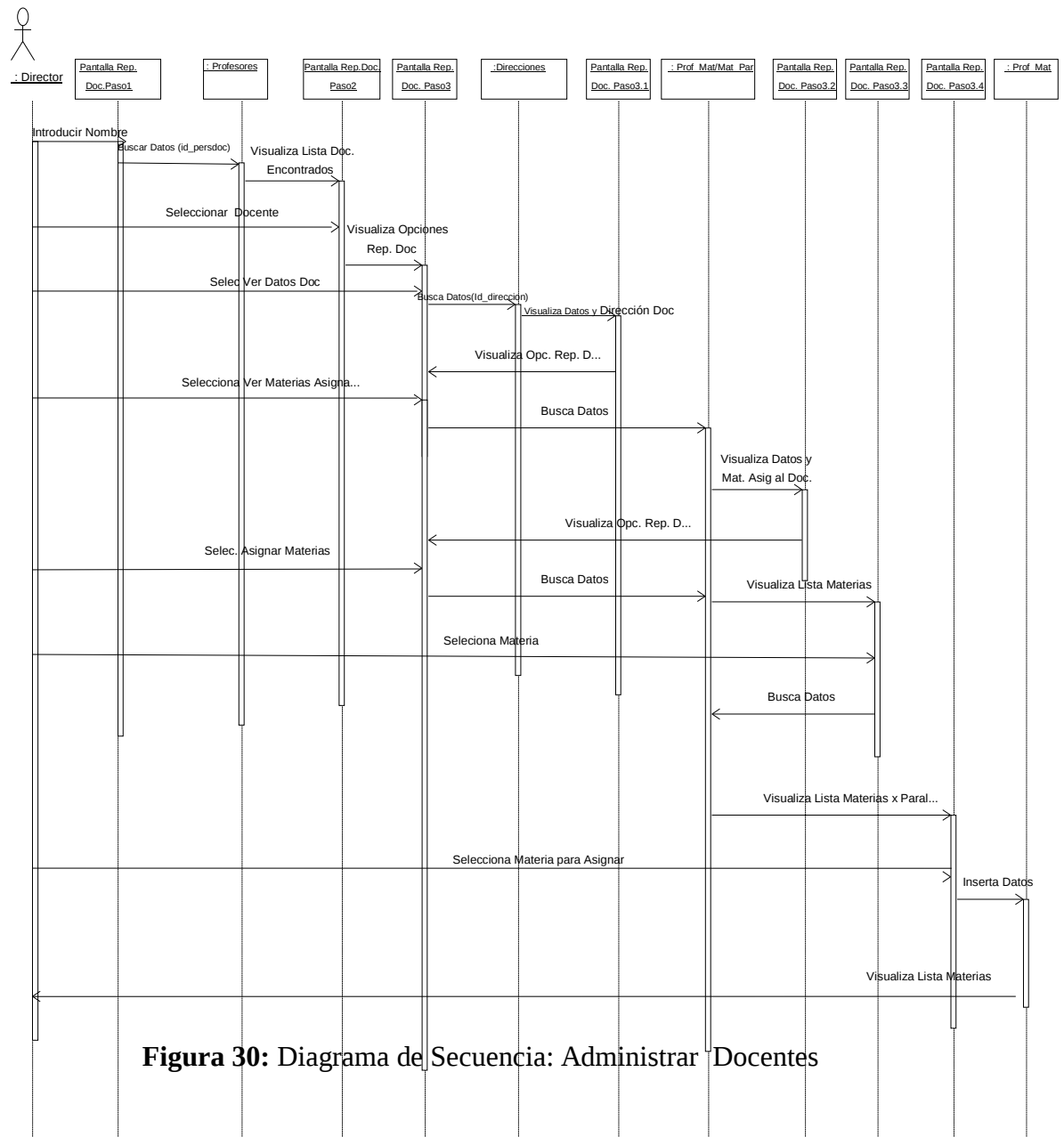


Figura 29: Diagrama de Secuencia: Administrar Paralelos

Administrar Docentes



Habilitar y Deshabilitar ADM

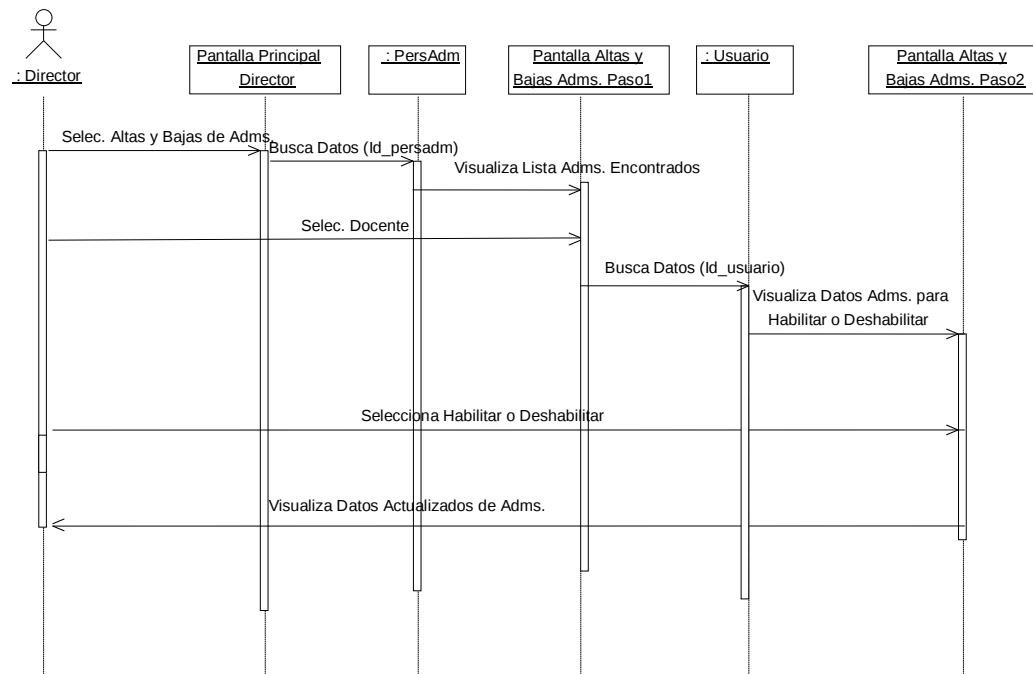


Figura 31: Diagrama de Secuencia: Habilitar y deshabilitar Administrativos

Registrar Administrativos

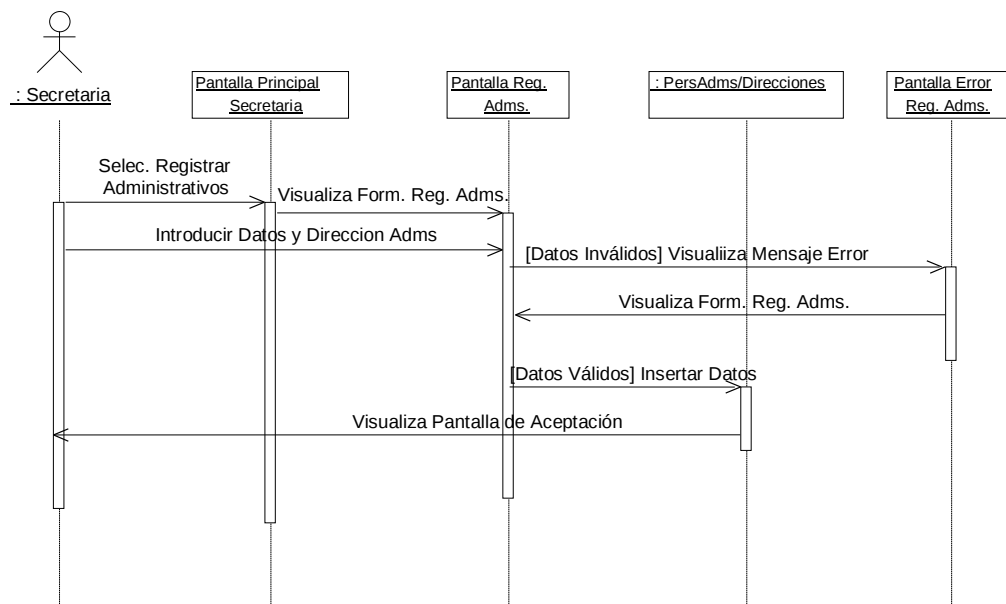


Figura 32: Diagrama de Secuencia: Registrar Administrativos

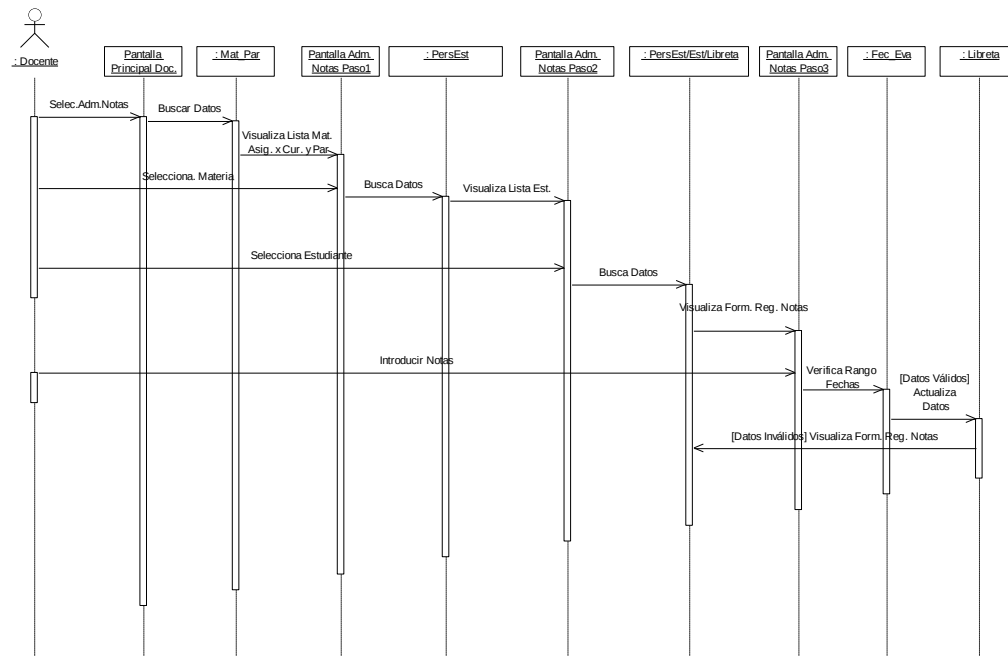


Figura 33: Diagrama de Secuencia: Administrar Notas

Ingresar al Sistema

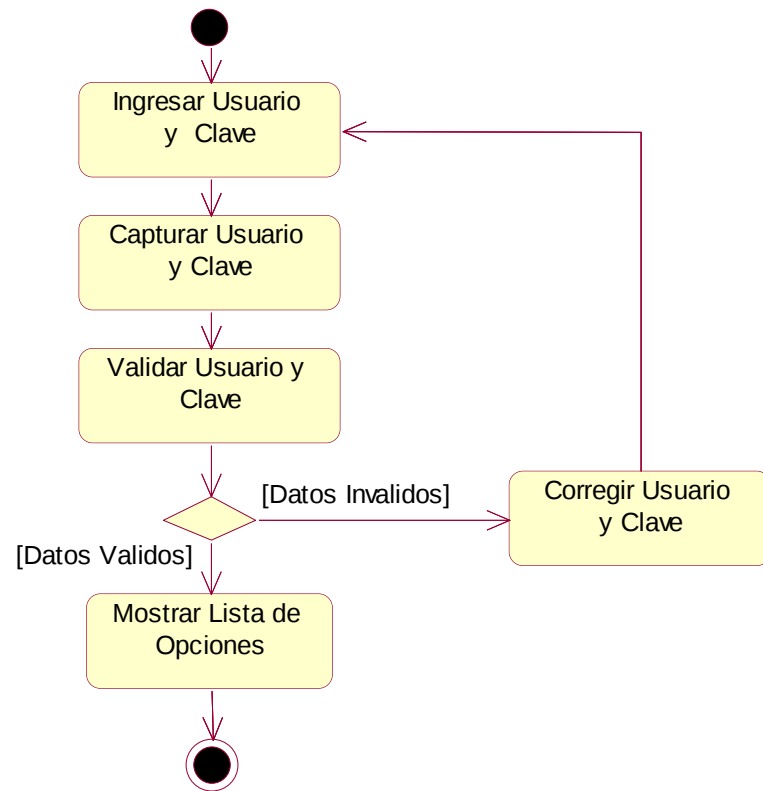


Figura 34: Diagrama de Actividad: Ingresar al Sistema

Habilitar y Deshabilitar Docentes

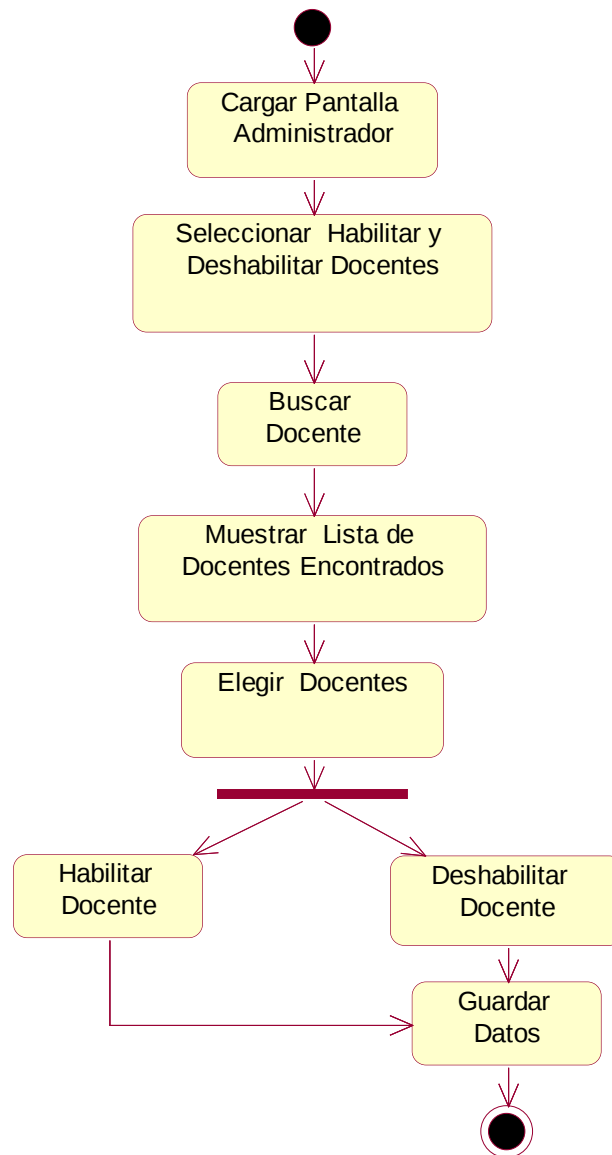


Figura 35: Diagrama de Actividad: Habilitar y Deshabilitar Docentes

Administrar Materias

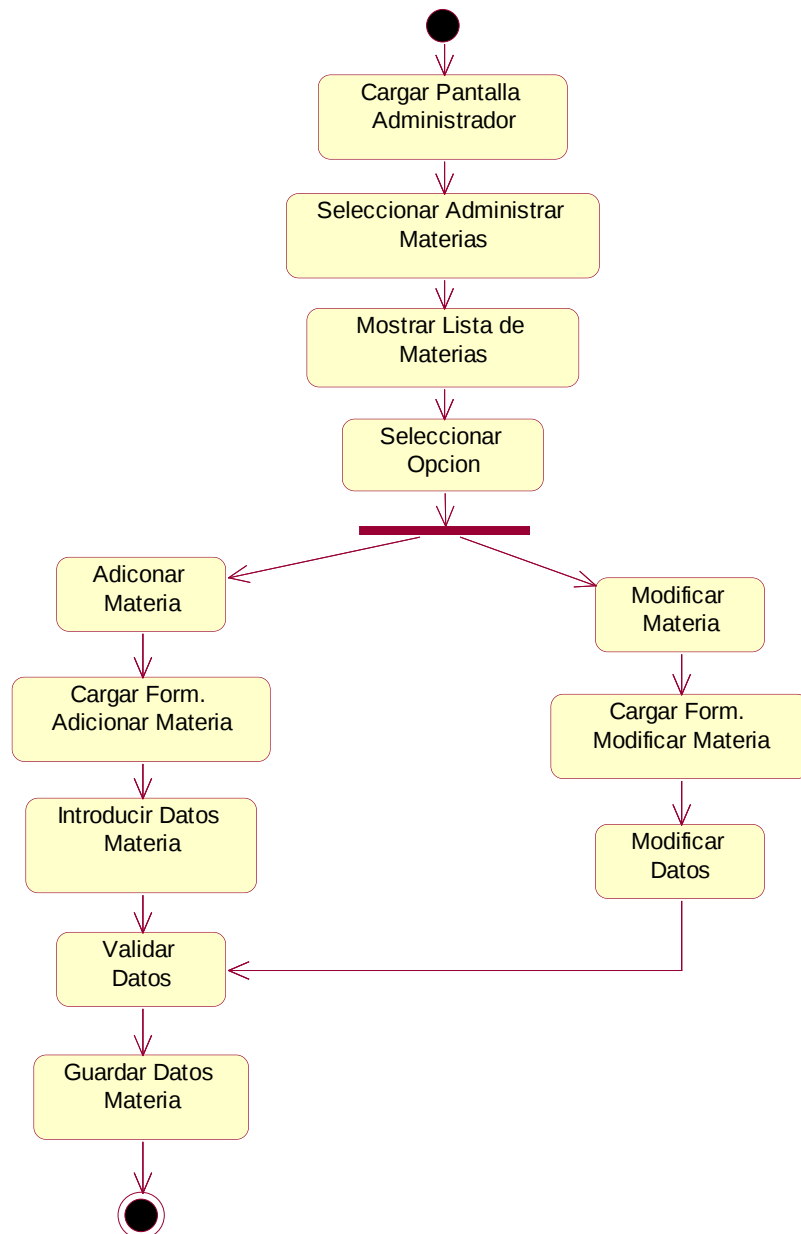


Figura 36: Diagrama de Actividad: Administrar Materias

Administrar Opciones

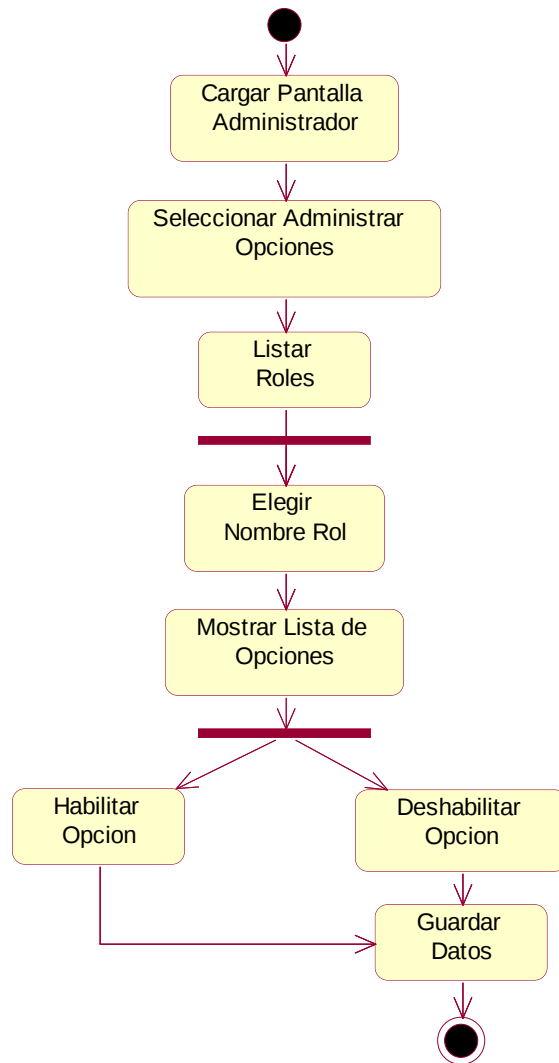


Figura 37: Diagrama de Actividad: Administrar Opciones

Asignar Roles

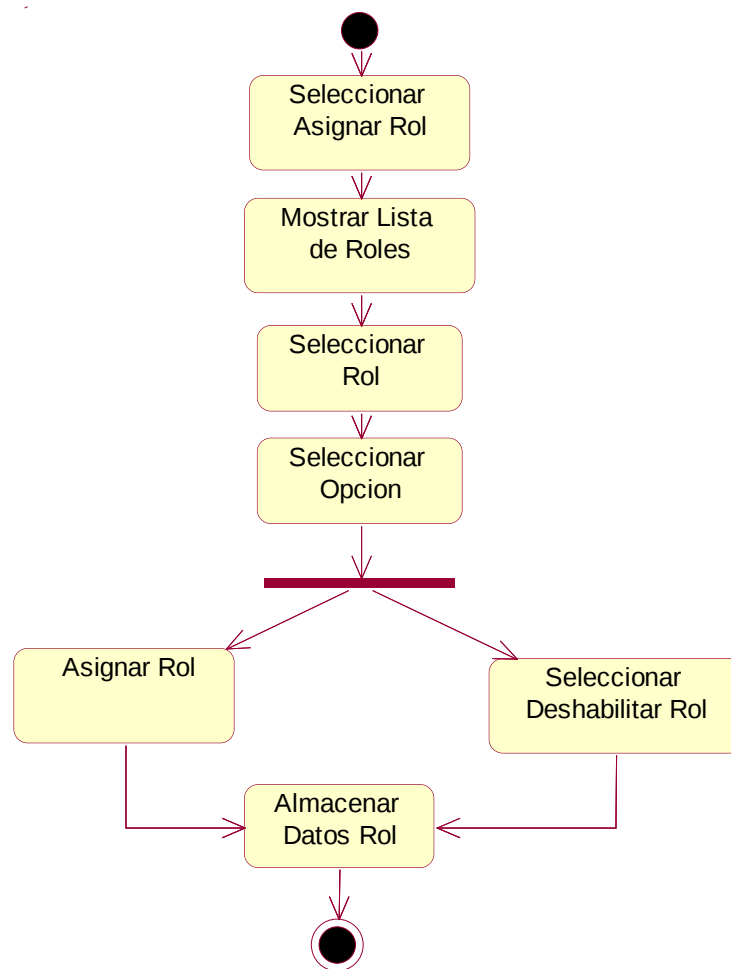


Figura 38 :Diagrama de Actividad: Asignar Roles

Fechas para Calificar

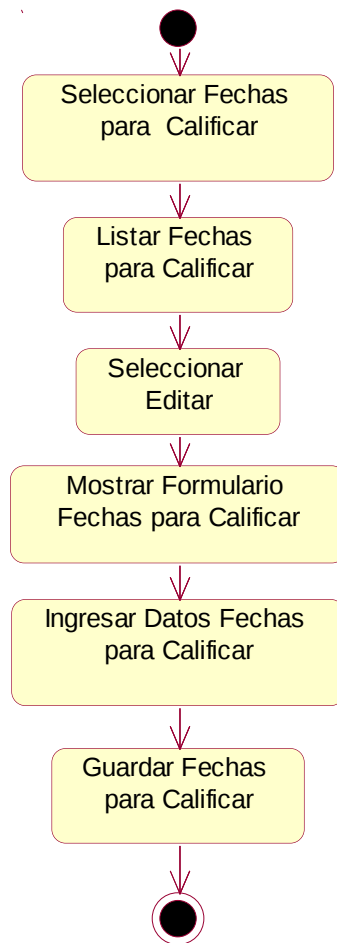


Figura 39: Diagrama de Actividad: Fechas para Calificar

Administrar Clave

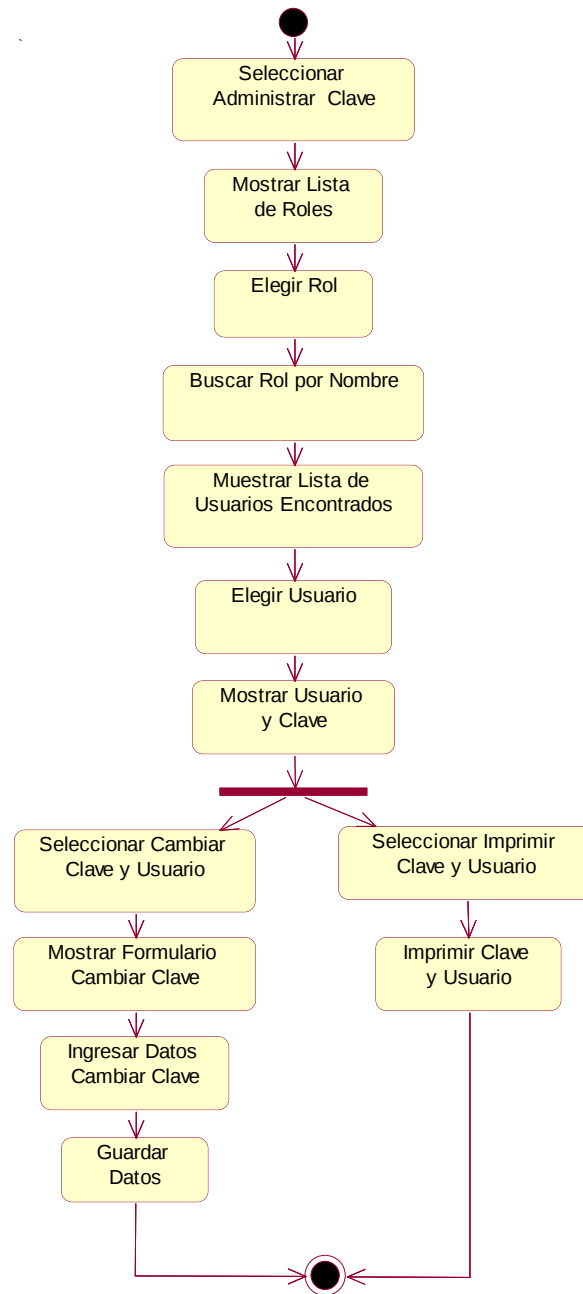


Figura 40: Diagrama de Actividad: Administrar Clave

Administrar Materias

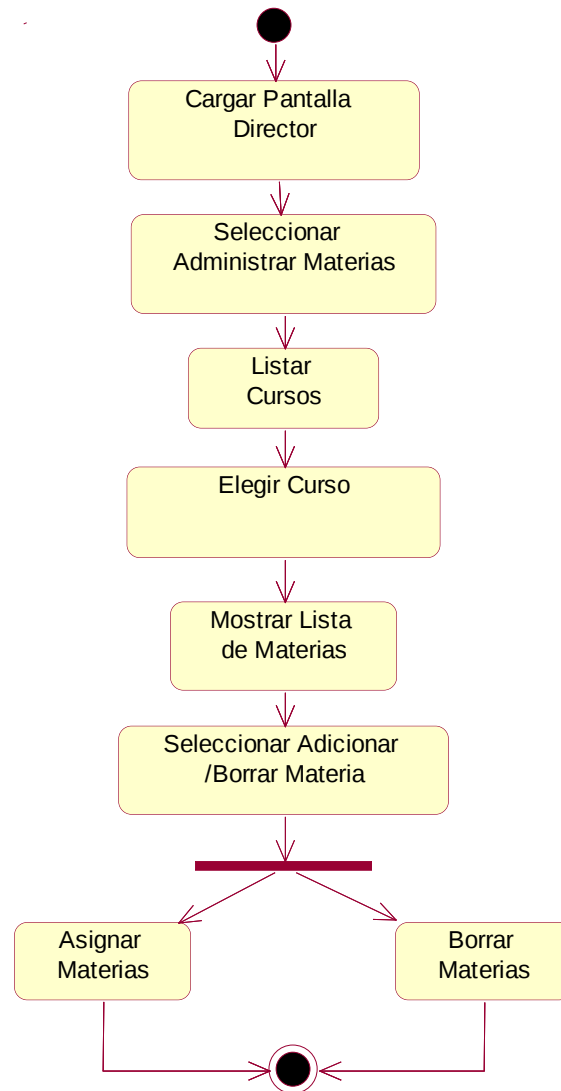


Figura 8: Administrar Clave

Administrar Paralelo

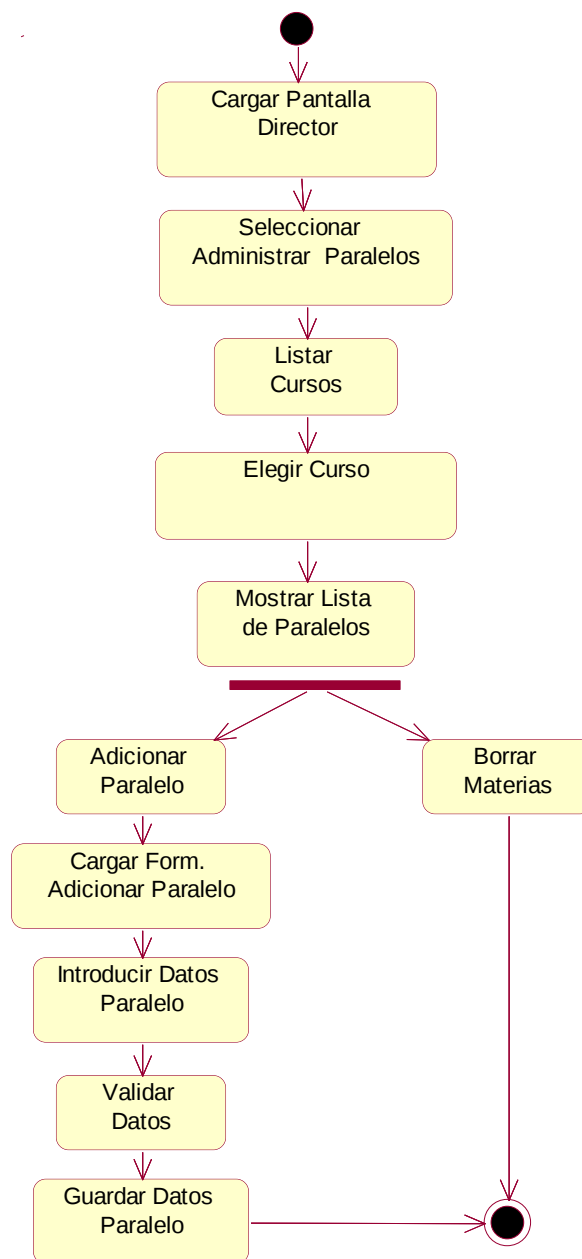


Figura 41: Diagrama de actividad: Administrar Paralelo

Administrar Docentes

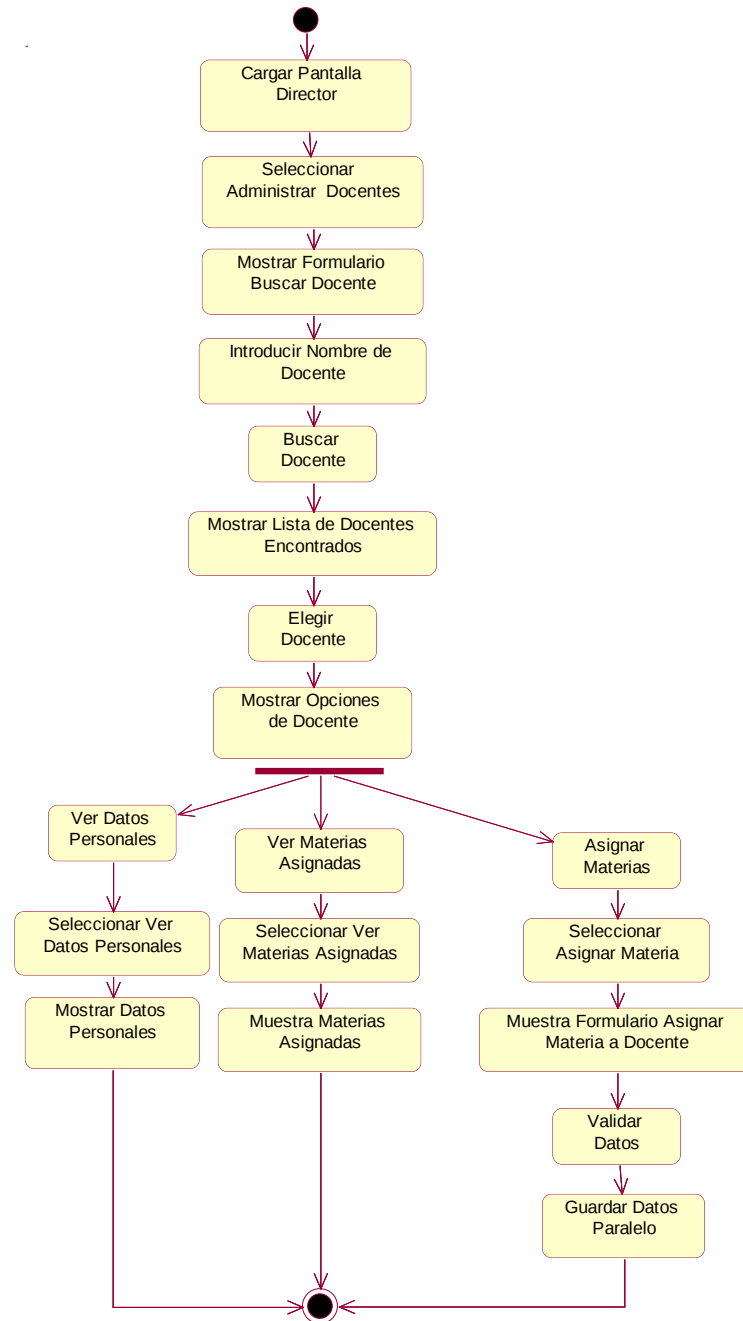


Figura 42: Diagrama de actividad: Administrar Docentes

Habilitar y Deshabilitar Administrativos

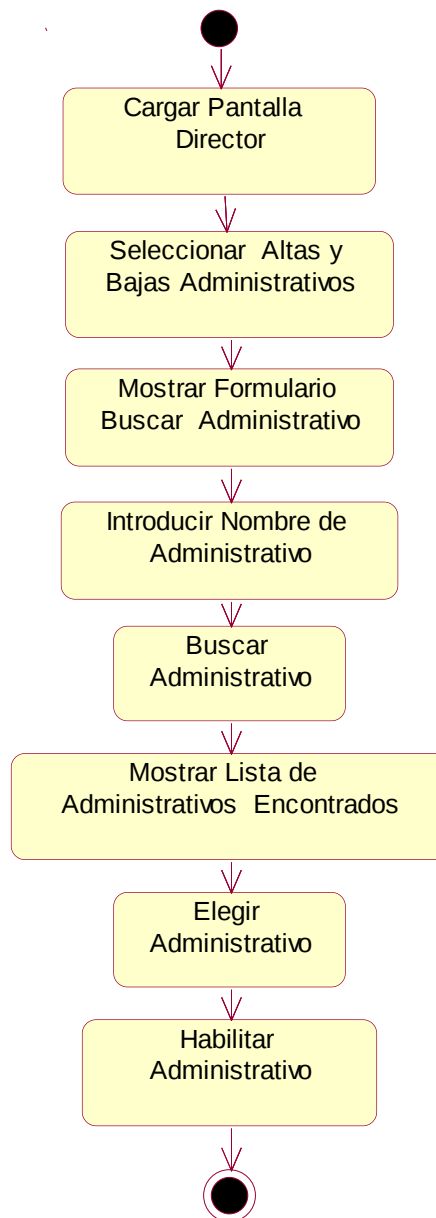


Figura 43: Diagrama de Casos de Uso : Habilitar y Deshabilitar Administrativos

Registrar Administrativo

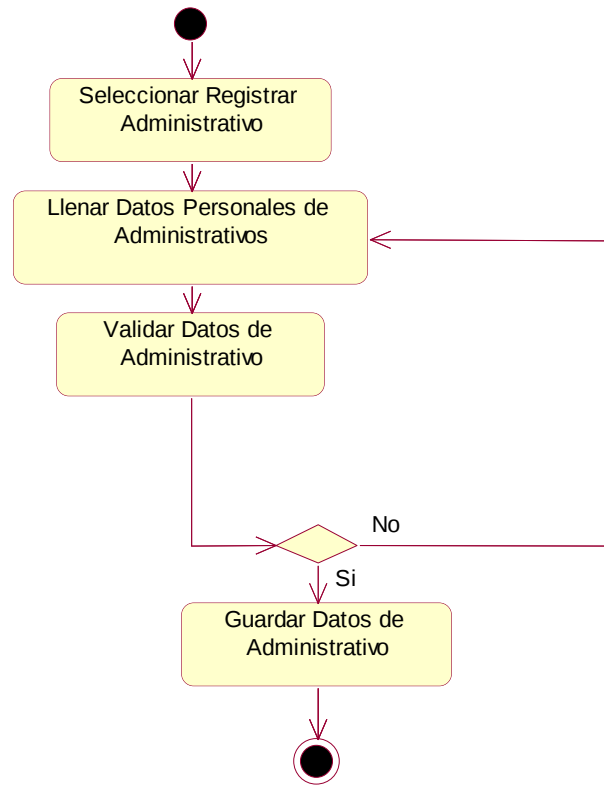


Figura 44: Diagrama de Actividad: Registrar Administrativo

Registrar Docente

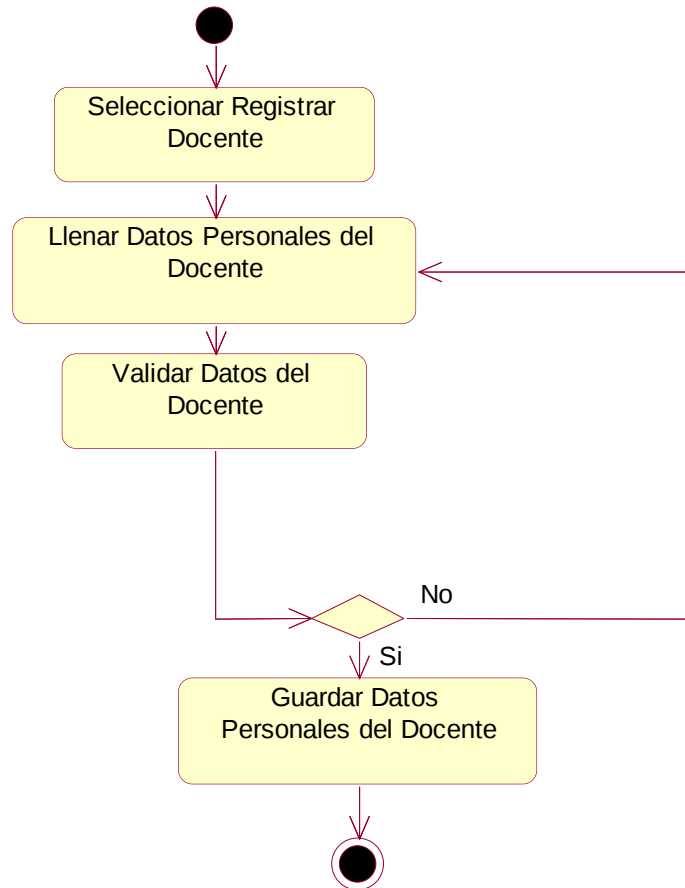
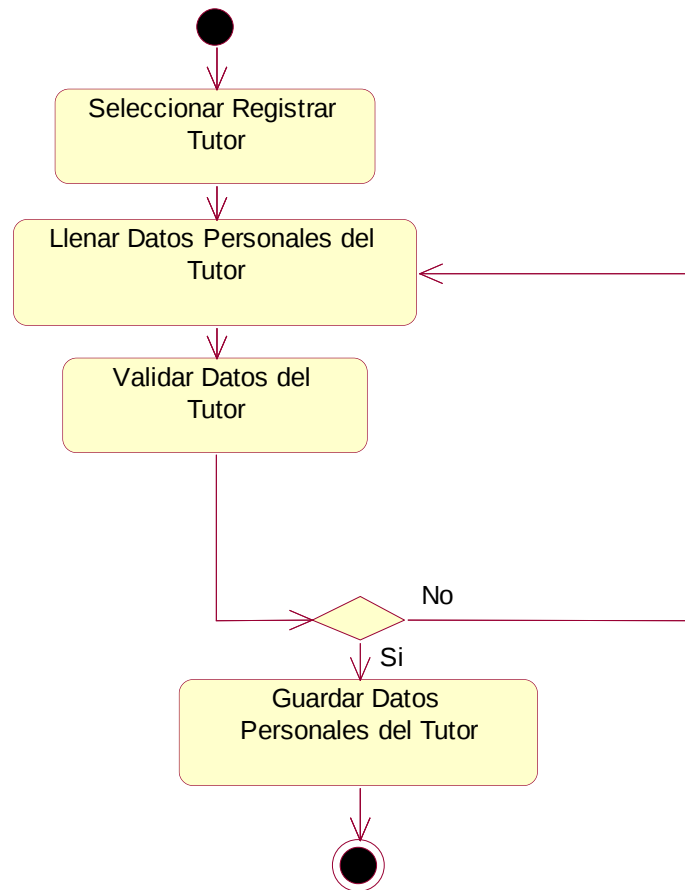


Figura 45 : Diagrama de Actividad: Registrar Administrativo

Registrar Tutor**Figura 46:** Diagrama de Actividad: Registrar Tutor

Administrar Notas

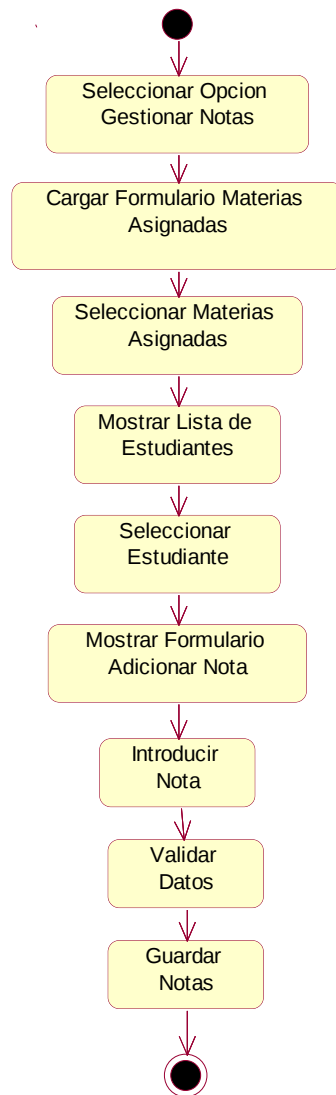
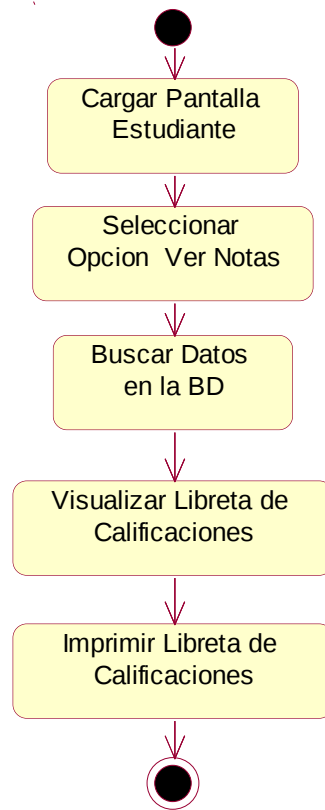


Figura 47: Diagrama de Actividad: Administrar Notas

Ver Notas**Figura 48:** Diagrama de Actividad: Ver Notas

Reporte de Estudiantes

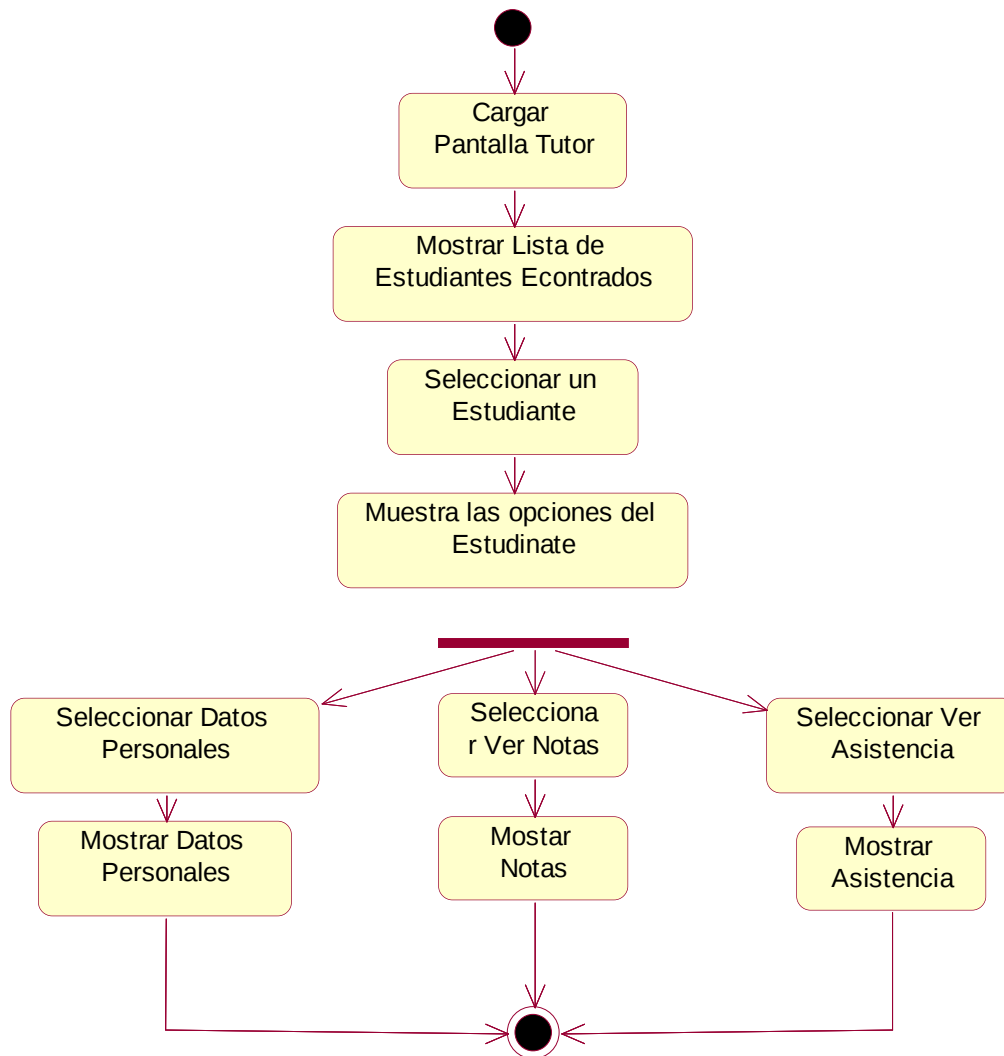
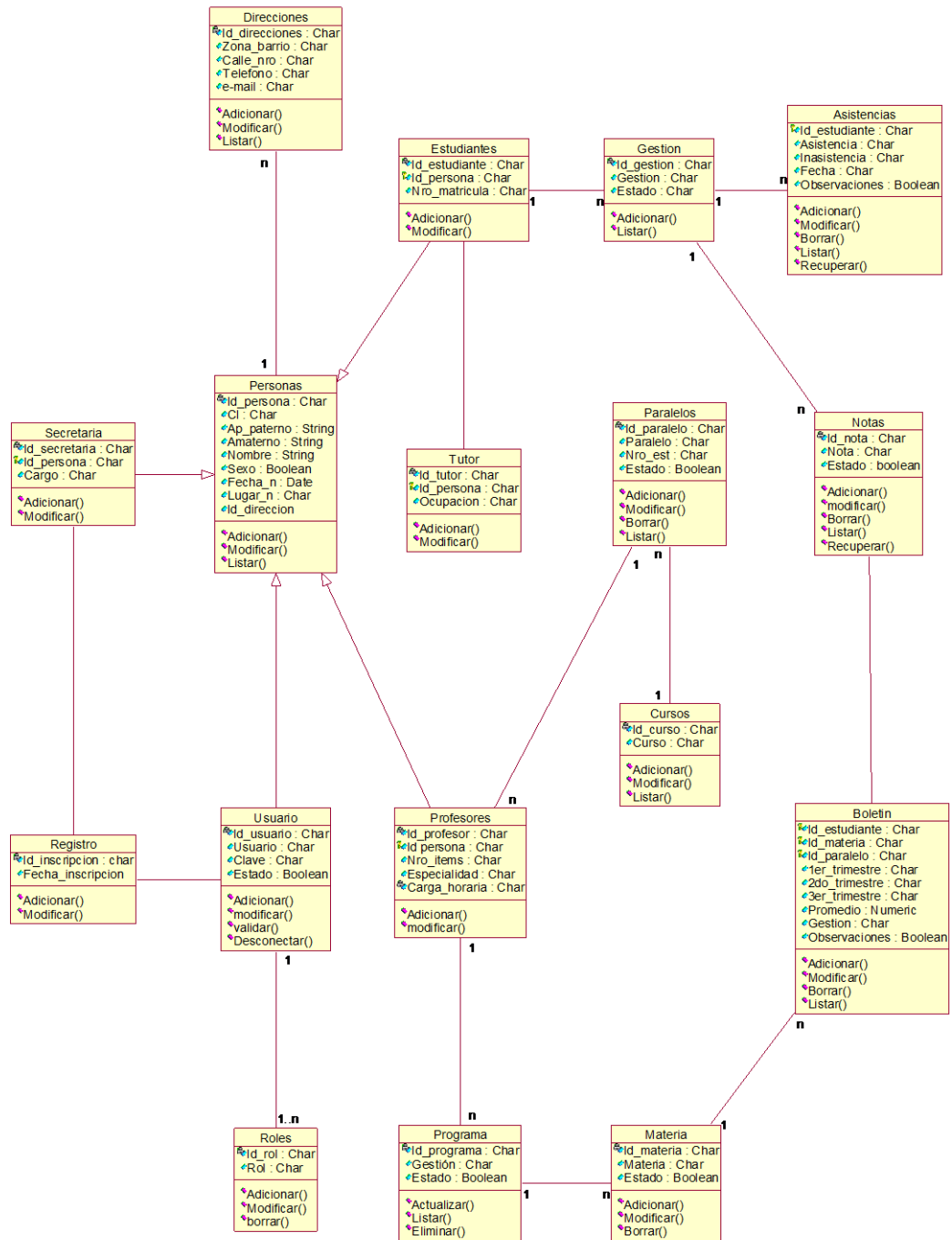
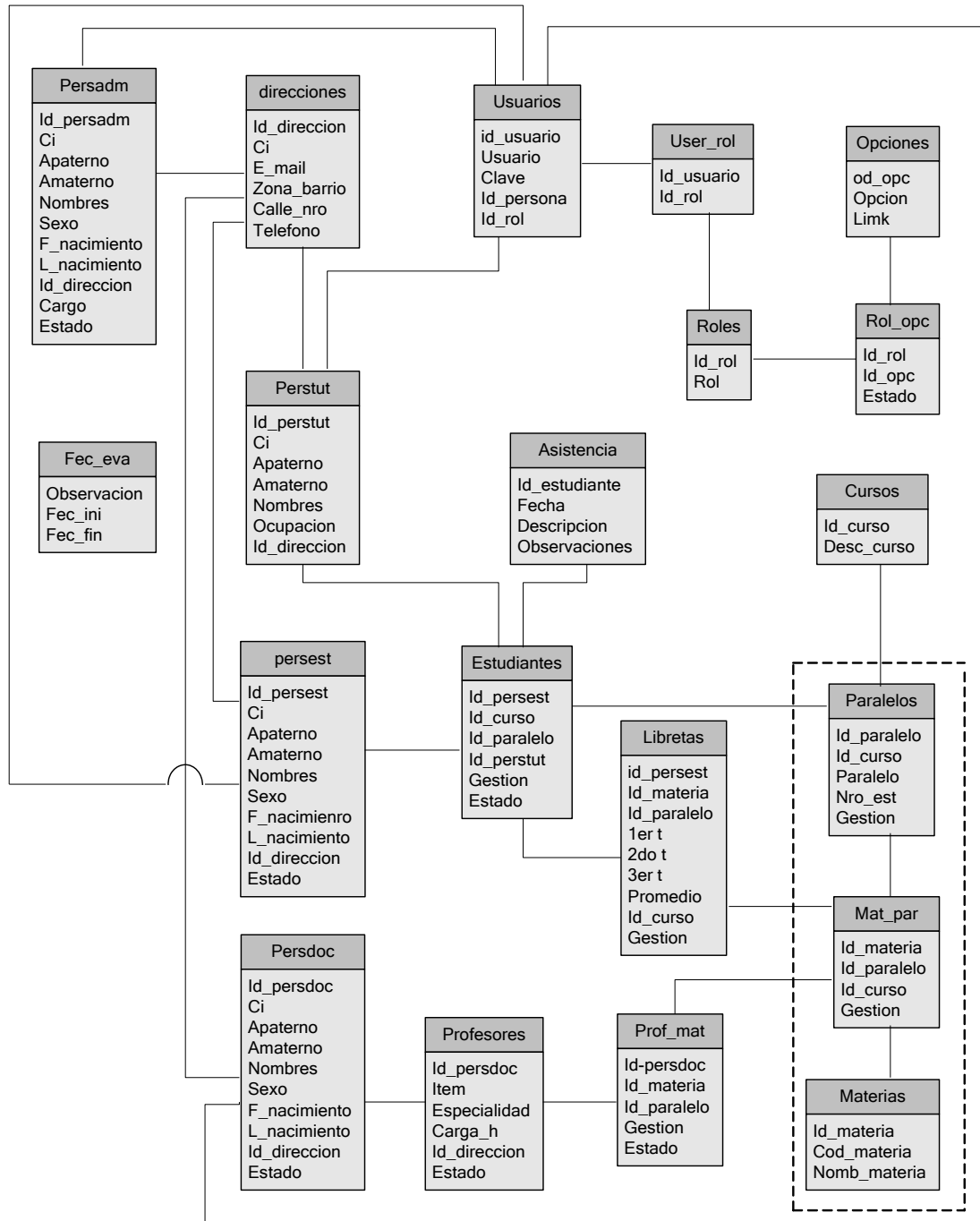


Figura 49: Diagrama de Actividad: Reporte de Estudiantes

5.1 Diagrama de Clases



5.2 Base de Datos



6.1 CONCLUSIONES

El desarrollo de este trabajo nos permite llegar a las siguientes conclusiones:

1. El Sistema Automatizado para el colegio Humberto Portocarrero, es buena solución para la toma de decisiones permitiendo así dar una mayor cobertura y organización de la información.
2. Se ha cumplido el objetivo principal y los específicos mencionados anteriormente, gracias a los conocimientos adquiridos en la Universidad y por otros medios de investigación que fueron el soporte fundamental para el desarrollo del Sistema
3. La utilización del Lenguaje UML (Lenguaje Unificado de Construcción de Modelos), permitió diseñar un sistema seguro confiable y mantenible.
4. La implementación de una Base de Datos Cliente/Servidor que garantiza la seguridad de la misma e incrementa la confiabilidad de la información.
5. Las tecnologías Web empleadas en el desarrollo del Sistema están recientemente explotadas en nuestro medio lo que nos lleva a cumplir con el principal objetivo de los grupos de investigación que es la explotación de nuevas tecnologías realizando de esta manera un aporte académico significativo.
6. La metodología RUP nos permite realizar mejoras al software, puesto que el proceso de ésta permite iterar y realizar las modificaciones necesarias para alcanzar la mayor calidad de software.

6.2 RECOMENDACIONES

Las recomendaciones que podemos dar, son:

1. Contar con los conocimientos básicos necesarios para la navegabilidad por Internet.
2. Capacitar adecuadamente al personal para el manejo de este sistema.

6.1 Prueba de Caja Negra

PANTALLA DE INGRESO DE INGRESO AL SISTEMA

CONDICIONES DE ENTRADA	EQUIVALENCIA VALIDA	EQUIVALENCIA INVALIDA
Tipo Usuario	1. Números	2. Otra cosa
Tamaño de Usuario	3. $1 \leq \text{valor} \leq 10$	4. <1 5. > 10
Tipo Clave	6. Números 7. Letras	8. Otra cosa
Tamaño Clave	9. $5 \leq \text{Valor} \leq 10$	10. <5 11. >10

PRUEBAS DE EQUIVALENCIA VALIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	administrador
Tamaño de clave	sis1
Clase de equivalencia que cubre	1.3. 6. 7. 9

Caso de prueba 2

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	director
Tamaño de clave	Dir1
Clase de equivalencia que cubre	1.3.6.7.9

Caso de prueba 3

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	profesor
Tamaño de clave	doc1
Clase de equivalencia que cubre	1.3.6.7.9

Caso de prueba 4

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	estudiante
Tamaño de clave	Estu1
Clase de equivalencia que cubre	1.3.6.7.9

Caso reprueba 5

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	tutor
Tamaño de clave	tut3
Clase de equivalencia que cubre	1.3.6.7.9

Caso de prueba 6

CAMPO	DATO INTRODUCIDO
-------	------------------

Tipo de Usuario	secretaria
Tamaño de clave	sec1
Clase de equivalencia que cubre	1.3.6.7.9

Caso de prueba 7

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	regente
Tamaño de clave	r1234
Clase de equivalencia que cubre	1.3.6.7.9

PRUEBA DE EQUIVALENCIA INVÁLIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	administradores
Tamaño de clave	Kljf1ÑÑLOLO
Clase de equivalencia que cubre	2. 5.8.10

Caso de prueba2

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	Administrador
Tamaño de clave	123456789012
Clase de equivalencia que cubre	2. 5.8.11

Caso de prueba3

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	Administradores
Tamaño de clave	fgfb
Clase de equivalencia que cubre	2. 5.8.10

Caso de prueba 4

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	Administradoreeeeeees
Tamaño de clave	fgfbdsfbsdkfs
Clase de equivalencia que cubre	2. 5.8.10

Caso de prueba 5

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	Administradores
Tamaño de clave	1234
Clase de equivalencia que cubre	2. 5.8. 11

Caso de prueba3

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	&&&&&=====0·
Tamaño de clave	1
Clase de equivalencia que cubre	2.5.8.10

Caso de prueba3

CAMPO	DATO INTRODUCIDO
Tipo de Usuario	
Tamaño de clave	115937
Clase de equivalencia que cubre	2.5.8.10

PANTALLA GESTIONAR NOTAS

CONDICIONES DE ENTRADA	EQUIVALENCIA VALIDA	EQUIVALENCIA INVALIDA
Nota	1. integer	2. Letras 3. Otra cosa
Tamaño Nota	4. $1 \leq \text{valor} \leq 70$	5. <1 6. > 70

PRUEBAS DE EQUIVALENCIA VALIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nota	36
Clase de equivalencia que cubre	1. 4

Caso de prueba2

CAMPO	DATO INTRODUCIDO
Nota	45
Clase de equivalencia que cubre	1. 4

Caso de prueba3

CAMPO	DATO INTRODUCIDO
Nota	55
Clase de equivalencia que cubre	1.4

PRUEBA DE EQUIVALENCIA INVÁLIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nota	kjg
Clase de equivalencia que cubre	2.5.6

Caso de prueba2

CAMPO	DATO INTRODUCIDO
Nota	,????
Clase de equivalencia que cubre	3.5.6

Caso de prueba2

CAMPO	DATO INTRODUCIDO
Nota	juhy
Clase de equivalencia que cubre	2.5.6

PANTALLA ADICIONAR PARALELOS

CONDICIONES DE ENTRADA	EQUIVALENCIA VALIDA	EQUIVALENCIA INVALIDA
Nombre Paralelo	1. Varchar	2. Otra cosa
Tamaño Nota	3. $1 \leq \text{valor} \leq 15$	4. < 1 5. > 16

PRUEBAS DE EQUIVALENCIA VALIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Paralelo	Primeroa
Clase de equivalencia que cubre	1. 3

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Paralelo	cuartob
Clase de equivalencia que cubre	1. 3

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Paralelo	¿?loo,,.....???
Clase de equivalencia que cubre	2.4.5

PRUEBAS DE EQUIVALENCIA INVÁLIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Paralelo	Ñ()//////¿¿¿?
Clase de equivalencia que cubre	2.4.5

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Paralelo	¿?,.....kgjuht
Clase de equivalencia que cubre	2.4.5

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Paralelo	Kgfujryfhr%%%%%%%%
Clase de equivalencia que cubre	2.4.5

PANTALLA ADMINISTRAR MATERIAS A DOCENTES

CONDICIONES DE ENTRADA	EQUIVALENCIA VALIDA	EQUIVALENCIA INVALIDA
Nombre Profesor	1. texto	2. Numeros 3. Otra cosa
Tamaño Nombre Profesor	4. $1 \leq \text{valor} \leq 50$	5. <1 6. >50

PRUEBAS DE EQUIVALENCIA VALIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Profesor	Reynaldo
Clase de equivalencia que cubre	1. 4

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Profesor	María Luisa
Clase de equivalencia que cubre	1. 4

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Profesor	Olga
Clase de equivalencia que cubre	1. 4

PRUEBAS DE EQUIVALENCIA INVÁLIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Profesor	125625????==
Clase de equivalencia que cubre	2.3.5

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Profesor	LO=?15241
Clase de equivalencia que cubre	2.3.5

Caso de prueba1

CAMPO	DATO INTRODUCIDO
Nombre Profesor	12547???????
Clase de equivalencia que cubre	2.3.5

PANTALLA REGISTRAR PROFESOR

CONDICIONES DE ENTRADA	EQUIVALENCIA VALIDA	EQUIVALENCIA INVALIDA
CI	1. Letra 2. Numero	3. Otra cosa
Tamaño de CI	4. <=1 valor <=10	5. <1 6. >10
Apellido Paterno	7. letra	8. Otra cosa 9. Numero
Tamaño de Apellido Paterno	10. <=1 valor <=15	11. <1 12. >15
Apellido Materno	13. Letra	14. Otra cosa 15. Numero
Tamaño de Apellido Materno	16. <=1 valor <=15	17. <1 18. >15
Nombres	19. Letra	20. Otra cosa 21. Numero
Tamaño Nombres	22. <=1 valor <=20	23. <1 24. >20
Fecha de Nacimiento	25. Date	26. Otra cosa
Tamaño Fecha de Nacimiento	27.-Fecha >=1 28.-Fecha <=15	29.-Fecha =0 30.-Fecha >15
Lugar de Nacimiento	31. Texto	32. Otra cosa
Tamaño de Lugar de Nacimiento	33.-Lugar >=1 34.-Lugar <=15	35.-Procedencia=0 36.-Procedencia >15
Sexo	37. Varchar 38.=1	39. Numeros 40. Otra cosa
Zona o Barrio	41. Numeros 42. Letras	43. Otra cosa
Tamaño Zona o Barrio	44. <=1 Valor <=25	45. <1 46. >25
Calle y Numero	47. Numeros 48. Letras	49. Otra cosa
Tamaño Calle y Numero	50. <=1 Valor <=35	51. <1 52. >35
teléfono	53. números	54. Otra cosa 55. Letra
Tamaño de teléfono	56. <=1 valor <=18	57. <1 58. >18

PRUEBAS DE EQUIVALENCIA VALIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	3698583
Apellido Paterno	Zenteno
Apellido Materno	Zanabria
Nombres	Ross Mery
Fecha de Nacimiento	23/11/1975
Lugar de Nacimiento	Tarija
Sexo	F
Zona o Barrio	La loma
Calle y Numero	Calle México 1114
teléfono	6641583

Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.25.2728.31.33.34.37.38.41.42.44 47.48.50.53.56
---------------------------------	--

Caso de prueba2

CAMPO	DATO INTRODUCIDO
CI	5074445
Apellido Paterno	Aguilar
Apellido Materno	Paco
Nombres	Edgar
Fecha de Nacimiento	11/02/1974
Lugar de Nacimiento	Tarija
Sexo	M
Zona o Barrio	La loma
Calle y Numero	Calle Colombia 256
teléfono	6641583
Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.25.2728.31.33.34.37.38.41.42.44 47.48.50.53.56

Caso de prueba3

CAMPO	DATO INTRODUCIDO
CI	4128758
Apellido Paterno	Escalante
Apellido Materno	Bejarano
Nombres	Olga
Fecha de Nacimiento	02/12/1979
Lugar de Nacimiento	Tarija
Sexo	M
Zona o Barrio	Cuarto centenario
Calle y Numero	Salomón Benítez 1456
teléfono	6634590
Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.25.2728.31.33.34.37.38.41.42.44 47.48.50.53.56

PRUEBAS DE EQUIVALENCIA INVÁLIDAS

Caso de prueba2

CAMPO	DATO INTRODUCIDO
CI	2568*65
Apellido Paterno	1ñploikñlopikikik
Apellido Materno	kjuihygt
Nombres	2541524
Fecha de Nacimiento	ñloki
Lugar de Nacimiento	15/58251
Sexo	1254
Zona o Barrio	,kloi66666

Calle y Numero	87656665%%%
teléfono	Ujhy765
Clase de equivalencia que cubre	3.5.8.12.14.18.21.26.30.32.36.39.43.49.55

Caso de prueba2

CAMPO	DATO INTRODUCIDO
CI	\$\$\$\$\$\$
Apellido Paterno	25147,
Apellido Materno	Lo8956
Nombres	2541l
Fecha de Nacimiento	T%%%%
Lugar de Nacimiento	Hjyhtg4
Sexo	1524
Zona o Barrio	Loity&&&
Calle y Numero	&&&&&
teléfono	ñploikij
Clase de equivalencia que cubre	3.5.8.12.14.18.21.26.30.32.36.39.43.49.55

Caso de prueba3

CAMPO	DATO INTRODUCIDO
CI	¿¿¿¿¿¿¿
Apellido Paterno	251478965
Apellido Materno	Lo8956524
Nombres	26547''''''
Fecha de Nacimiento	T%%%%
Lugar de Nacimiento	4587ñplok
Sexo	152463
Zona o Barrio	25147%%%
Calle y Numero	Qiiii
teléfono	lokfruty
Clase de equivalencia que cubre	3.5.8.12.14.18.21.26.30.32.36.39.43.49.55

PANTALLA REGISTRAR TUTOR

CONDICIONES DE ENTRADA	EQUIVALENCIA VALIDA	EQUIVALENCIA INVALIDA
CI	1. Letra 2. Números	3. Otra cosa
Tamaño de CI	4. <=1 valor <=10	5. <1 6. >10
Apellido Paterno	7. letra	8. Otra cosa 9. Numero
Tamaño de Apellido Paterno	10. <=1 valor <=15	11. <1 12. >15
Apellido Materno	13. Letra	14. Otra cosa 15. Numero
Tamaño de Apellido	16. <=1 valor <=15	17. <1 18. >15

Materno		
Nombres	19.Letra	20. Otra cosa 21. Numero
Tamaño Nombres	22. <=1 valor <=20	23. <1 24. >20
Ocupación	25.Texto	26.Números 27.Otra cosa
Tamaño Ocupacion	28.<=1 Valor<=20	29.<1 30.>20

PRUEBAS DE EQUIVALENCIA VALIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	5032698
Apellido Paterno	Acebo
Apellido Materno	Lopez
Nombres	Mario
Ocupación	Pintor
Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.28

Caso de prueba2

CAMPO	DATO INTRODUCIDO
CI	3152896
Apellido Paterno	Castro
Apellido Materno	Mealla
Nombres	Cecilia
Ocupación	Labores de casa
Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.28

Caso de prueba3

CAMPO	DATO INTRODUCIDO
CI	4596872
Apellido Paterno	Miranda
Apellido Materno	Rios
Nombres	Laura
Ocupación	Comerciante
Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.28

PRUEBAS DE EQUIVALENCIA INVÁLIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	&%&/////
Apellido Paterno	2547844?
Apellido Materno	478125%%%
Nombres	1478''''''''
Ocupación	NHJKM
Clase de equivalencia que cubre	3.9.15.21.27

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	%%%%%%%%%
Apellido Paterno	25417\$\$\$\$
Apellido Materno	478125¿¿¿¿
Nombres	1478''''''''
Ocupación	JUHYTR
Clase de equivalencia que cubre	3.9.15.21.27

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	%&JUYT
Apellido Paterno	251478?
Apellido Materno	25698****
Nombres	ÑPLOKIK
Ocupación	253¿¿¿¿¿¿
Clase de equivalencia que cubre	3.9.15.21.27

PANTALLA REGISTRAR ESTUDIANTE

CONDICIONES DE ENTRADA	EQUIVALENCIA VALIDA	EQUIVALENCIA INVALIDA
CI	1. Letra 2. Numero	3. Otra cosa
Tamaño de CI	4. <=1 valor <=10	5. <1 6. >10
Apellido Paterno	7. letra	8. Otra cosa 9. Numero
Tamaño de Apellido Paterno	10. <=1 valor <=15	11. <1 12. >15
Apellido Materno	13. Letra	14. Otra cosa 15. Numero
Tamaño de Apellido Materno	16. <=1 valor <=15	17. <1 18. >15
Nombres	19. Letra	20. Otra cosa 21. Numero
Tamaño Nombres	22. <=1 valor <=20	23. <1 24. >20
Fecha de Nacimiento	25. Date	26. Otra cosa

Tamaño Fecha de Nacimiento	27.-Fecha >=1 28.-Fecha <=15	29.-Fecha =0 30.-Fecha>15
Lugar de Nacimiento	31.Texto	32.Otra cosa
Sexo	33.Varchar	35.Numeros
Tamaño sexo	36.=1	37.>1
CI tutor	38. Letra 39.Números	40. Otra cosa
Tamaño CI tutor	41. <=1 valor <=10	42. <1 43. >10

PRUEBAS DE EQUIVALENCIA VALIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	7158987
Apellido Paterno	Acebo
Apellido Materno	Tarifa
Nombres	Eddy
Fecha de Nacimiento	23/11/1991
Lugar de Nacimiento	Tarija
Sexo	M
Zona o Barrio	15 de junio
Calle y Numero	Calle México 1114
teléfono	77173745
Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.25.2728.31.33.34.37.38.41.42.44 47.48.50.53.56

Caso de prueba2

CAMPO	DATO INTRODUCIDO
CI	7158962
Apellido Paterno	Fernandez
Apellido Materno	Castro
Nombres	Ruth
Fecha de Nacimiento	25/10/1992
Lugar de Nacimiento	Tarija
Sexo	F
Zona o Barrio	Cuarto centenario
Calle y Numero	Calle Salomón Benítez
teléfono	66433085
Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.25.2728.31.33.34.37.38.41.42.44 47.48.50.53.56

Caso de prueba3

CAMPO	DATO INTRODUCIDO
CI	7152481
Apellido Paterno	Burgos
Apellido Materno	Morales
Nombres	Patricia
Fecha de Nacimiento	03/09/1991

Lugar de Nacimiento	Tarija
Sexo	F
Zona o Barrio	Villa Avaroa
Calle y Numero	Sucre 1125
teléfono	6666568
Clase de equivalencia que cubre	2.4.7.10.13.16.19.22.25.2728.31.33.34.37.38.41.42.44 47.48.50.53.56

PRUEBAS DE EQUIVALENCIA INVÁLIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	lo??????
Apellido Paterno	152463536252
Apellido Materno	18547225147¿¿¿¿¿
Nombres	2536985
Fecha de Nacimiento	¿¿¿¿¿
Lugar de Nacimiento	ñloijuy
Sexo	F
Zona o Barrio	IVkiju
Calle y Numero	Calokiju3.
teléfono	66433085
Clase de equivalencia que cubre	3.6.9.14.18.23.29.32.37.40.43

Caso de prueba2

CAMPO	DATO INTRODUCIDO
CI	71589622525854
Apellido Paterno	417¿¿¿¿¿¿¿
Apellido Materno	¿?????5147
Nombres	2547891
Fecha de Nacimiento	214*5555*5587
Lugar de Nacimiento	258479514285
Sexo	/*/*
Zona o Barrio	Cuarto centenario
Calle y Numero	Lñpol4875/***
teléfono	635847j`
Clase de equivalencia que cubre	3.6.9.14.18.23.29.32.37.40.43

Caso de prueba3

CAMPO	DATO INTRODUCIDO
-------	------------------

CI	74152876514
Apellido Paterno	25847''''*****
Apellido Materno	.ñl,lñ*587//*7
Nombres	ñl,pm,
Fecha de Nacimiento	254784
Lugar de Nacimiento	lñpop
Sexo	F
Zona o Barrio	Ñploñpñ2548
Calle y Numero	Calle Salomón Benítez
teléfono	66433085
Clase de equivalencia que cubre	3.6.9.14.18.23.29.32.37.40.43

PANTALLA REGISTRAR ADMINISTRATIVOS

CONDICIONES DE ENTRADA	EQUIVALENCIA VALIDA	EQUIVALENCIA INVALIDA
CI	1. Letra 2. Numero	3. Otra cosa
Tamaño de CI	4. <=1 valor <=10	5. <1 6. >10
Apellido Paterno	7. letra	8. Otra cosa 9. Numero
Tamaño de Apellido Paterno	10. <=1 valor <=15	11. <1 12. >15
Apellido Materno	13. Letra	14. Otra cosa 15. Numero
Tamaño de Apellido Materno	16. <=1 valor <=15	17. <1 18. >15
Nombres	19. Letra	20. Otra cosa 21. Numero
Tamaño Nombres	22. <=1 valor <=20	23. <1 24. >20
Fecha de Nacimiento	25. Date	26. Otra cosa
Tamaño Fecha de Nacimiento	27.-Fecha >=1 28.-Fecha <=15	29.-Fecha =0 30.-Fecha >15
Lugar de Nacimiento	31. Texto	32. Otra cosa

PRUEBAS DE EQUIVALENCIA VALIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	4153625
Apellido Paterno	Benitez

Apellido Materno	Ecos
Nombres	Ruth
Fecha de Nacimiento	23/11/1965
Lugar de Nacimiento	Oruro
Sexo	F
Zona o Barrio	Cuarto Centenario
Calle y Numero	Calle México 1114
teléfono	77173745
Clase de equivalencia que cubre	1.4.7.10.13.16.19.22.25.27.28.31.33.36.38.39.41

Caso de prueba2

CAMPO	DATO INTRODUCIDO
CI	4258965
Apellido Paterno	Flores
Apellido Materno	Castro
Nombres	Maria Magdalena
Fecha de Nacimiento	25/12/1965
Lugar de Nacimiento	Tarija
Sexo	F
Zona o Barrio	Zona Central
Calle y Numero	Sucre 253
teléfono	6641583
Clase de equivalencia que cubre	1.4.7.10.13.16.19.22.25.27.28.31.33.36.38.39.41

Caso de prueba3

CAMPO	DATO INTRODUCIDO
CI	5218693
Apellido Paterno	Heredia
Apellido Materno	Ayarde
Nombres	Maria del Carmen
Fecha de Nacimiento	22/06/1979
Lugar de Nacimiento	Tarija
Sexo	F
Zona o Barrio	San Lorenzo
Calle y Numero	Sucre 1125
teléfono	6653063
Clase de equivalencia que cubre	1.4.7.10.13.16.19.22.25.27.28.31.33.36.38.39.41

PRUEBAS DE EQUIVALENCIA INVÁLIDAS

Caso de prueba1

CAMPO	DATO INTRODUCIDO
CI	lo??????
Apellido Paterno	152463536252
Apellido Materno	18547225147¿¿¿¿¿

Nombres	2536985
Fecha de Nacimiento	¿¿¿¿¿
Lugar de Nacimiento	ñloijuy
Sexo	F
Zona o Barrio	IVkiju
Calle y Numero	Calokiju3.
teléfono	66433085
Clase de equivalencia que cubre	3.6.9.15.21.26.32.35.37.40.43

Caso de prueba2

CAMPO	DATO INTRODUCIDO
CI	71589622525854
Apellido Paterno	417¿¿¿¿¿¿¿
Apellido Materno	¿?????5147
Nombres	2547891
Fecha de Nacimiento	214*5555*5587
Lugar de Nacimiento	258479514285
Sexo	/*/*
Zona o Barrio	Cuarto centenario
Calle y Numero	Lñpol4875/***
teléfono	635847i`
Clase de equivalencia que cubre	3.6.9.15.21.26.32.35.37.40.43

Caso de prueba3

CAMPO	DATO INTRODUCIDO
CI	74152876514
Apellido Paterno	25847'''*****
Apellido Materno	.ñl,lñ*587//*7
Nombres	ñl,pm,
Fecha de Nacimiento	254784
Lugar de Nacimiento	lñpop
Sexo	F
Zona o Barrio	Ñploñpñ2548
Calle y Numero	Calle Salomón Benítez
teléfono	66433085
Clase de equivalencia que cubre	3.6.9.15.21.26.32.35.37.40.43