



Pontificia Universidad Católica de Valparaíso

Facultad de Ingeniería

Escuela de Ingeniería Informática

**SAPP -
SISTEMA ADMINISTRADOR DE PROGRAMAS DE
POSTITULO**

Autor:

Eric Michael Aguilera Arce - Jaime Andrés Jeria Fernández

Informe final del Proyecto para optar al Título profesional de

Ingeniero de Ejecución en Informática

Profesor guía:

Silvana Roncagliolo de la Horra

Profesor Co-referente:

Alexandru Cristian Rusu

Julio 2007

Resumen

El Sistema Administrador de Programas de Postítulo (SAPP) fue desarrollado para la Oficina Técnica de Estudios Avanzados de la Pontificia Universidad Católica de Valparaíso.

Este sistema permite manejar de manera rápida y eficiente los programas de postítulo dictados en la Universidad y los datos de los alumnos pertenecientes a dichos programas.

Para realizar el software, se tomó una serie de requerimientos por parte del cliente, se decidió llevar a cabo una solución, la cual requirió un análisis y un diseño para lograr la creación de un sistema que pudiera cubrir de la mejor manera los problemas encontrados.

Todos los pasos requeridos para la creación del sistema, fueron basados en el paradigma Proceso Unificado de Desarrollo, el cual brindó reglas y tareas en cada una de las distintas etapas de la creación de la aplicación.

Se utilizó UML para la creación de diagramas, lo que ayudó a que en todo el proceso de desarrollo del sistema el cliente pudiera comprender de manera gráfica la solución que sería implantada.

SAPP fue programado utilizando lenguaje Java, y SQL Server como motor de base de datos.

Abstract

The System Manager of Programs of Postítulo (SAPP) was developed for the Technical Office of Advanced Studies of the Pontifical Catholic University of Valparaiso.

This system allows to handle fast and in an efficient way the programs of postítulo dictated in the University and the data of the students that belongs to these programs.

To carry out the software, a series of requirements were taken from the client, a solution was decided to carry out, which needed an analysis and a design to achieve the creation of a system that could satisfy of the best way the found problems.

All the steps required for the creation of the system, were based on the paradigm Unified Process of Development, which offered rules and tasks in each one of the different stages of the creation of the application.

UML for the creation of diagrams was used, which helped in all the process of development of the system, the client could understand on a graphical way, the solution that would be implanted.

SAPP was programmed on Java language, and SQL Server as database engine.

AGRADECIMIENTOS

A mis padres por su apoyo incondicional y consejos que siempre escuche, y me ayudaron a lograr lo que estoy alcanzando.

A mi familia por ser la piedra angular de lo que he llegado a ser tanto como persona y profesional. A mi novia por ser esa luz que ilumina los momentos más oscuros y mantiene la claridad en esos momentos felices de mi vida.

A mis amigos de Universidad, por hacer que el tiempo pasara inadvertido. A la Pontificia Universidad Católica de Valparaíso y a sus docentes por entregarme y enseñarme a usar las herramientas necesarias para ser un profesional íntegro.

Eric Aguilera Arce

AGRADECIMIENTOS

Agradezco a Dios y a mi Familia por estar siempre a mi lado brindándome su fuerza, energía y apoyo incondicional.

Doy gracias también a la Pontificia Universidad Católica de Valparaíso por darme la oportunidad de convertirme en un profesional íntegro, con los conocimientos adecuados para desempeñarme con excelencia en mi futuro laboral.

A mi esposa y mi hija por ser, mi apoyo y mi motivación.

A todos los que han estado conmigo y me han ayudado a ser quien soy hoy, gracias de corazón.

Jaime Jeria Fernández

INTRODUCCIÓN

El siguiente informe tiene como finalidad dar a conocer las distintas etapas que se llevaron a cabo para la construcción del Sistema Administrador de Programas de Postítulo, SAPP, el cual fue construido para la Oficina Técnica de Estudios Avanzados de la Pontificia Universidad Católica de Valparaíso.

Para dar una explicación más amplia del proyecto, es que se describe la organización para la que se desarrolló el sistema, dando a conocer su funcionamiento actual, la realidad tecnológica en la que se encuentra y los problemas que surgen ante dichas situaciones y que impulsaron la creación de SAPP.

Se describen también los requerimientos que el usuario del sistema especificó para la solución a desarrollar. Estos requerimientos son los que dieron paso a los diagramas de casos de uso y a otros diagramas, los cuales fueron formando progresivamente un proyecto que determinó la creación de una aplicación destinada a minimizar los problemas existentes en la organización.

El proyecto incluye también dentro de sus etapas, distintos estudios que deben ser realizados. Dichos estudios son contemplados en este informe, donde se puede verificar si la solución tiene la factibilidad de ser desarrollada e implementada, así como también se identifican los principales riesgos que se pueden correr y se detallan las mitigaciones y las contingencias sobre dichos riesgos en caso de que alguno de ellos ocurra.

Con el fin de mostrar la aplicación desarrollada, es que se anexan las imágenes de las principales interfaces del sistema, mostrando con esto que después de una serie de pasos, se logró construir un sistema a medida, que permite satisfacer las necesidades del cliente y que ayuda a solucionar la mayoría de los problemas existentes en la organización.

CAPÍTULO 1: ESTUDIO PRELIMINAR

1.1-INTRODUCCIÓN

Actualmente es muy fácil darse cuenta de que el mundo laboral es altamente competitivo y es necesario estar en una constante actualización de los conocimientos profesionales con el fin de que estos no queden obsoletos y la persona se pueda mantener vigente y poder prestar sus mejores aptitudes al lugar donde se desempeña.

Cabe destacar que no sólo se puede continuar con estudios exclusivamente por una actividad netamente profesional sino que también puede esto estar dado por un gusto personal de actualización o mejoramiento de los conocimientos profesionales que se poseen.

Esta realidad plantea la posibilidad de actualizar o refinar los conocimientos profesionales mediante programas de estudios que permiten aumentar el grado de conocimiento a través programas destinados a dar una mayor especificación a distintas áreas de conocimiento.

La Pontificia Universidad Católica de Valparaíso imparte programas de magíster, doctorado y postítulos, con lo que se le da a las personas, la posibilidad de mejorar profesionalmente, brindando grados de conocimientos específicos a modo de obtener profesionales con mayor capacidad que puedan desempeñarse de mejor manera en sus respectivas áreas de especialidad, con un alto grado de conocimiento reconocido a nivel nacional como internacional (en ciertos casos).

1.2.-DESCRIPCIÓN DEL TEMA

El presente proyecto está dirigido a la Oficina Técnica de Estudios Avanzados de la Pontificia Universidad Católica de Valparaíso (PUCV), área dependiente de la Vice-Rectoría de Investigación y Estudios Avanzados de la misma Universidad, específicamente a los programas de postítulo de los cuales se encarga esta Oficina.

Esta dependencia se ubica en la casa central de la PUCV y es la encargada de administrar y gestionar el proceso académico y técnico-pedagógico de los alumnos de los programas de postgrado y postítulo y evaluar su desarrollo con el propósito de cumplir los objetivos de la Vice-Rectoría.

La Oficina Técnica de Estudios Avanzados, debe planificar, organizar, dirigir, ejecutar, supervisar y evaluar el desarrollo de las actividades curriculares -pedagógicas con el fin de cumplir las metas exigidas por la Vice-Rectoría. Con este motivo, la Oficina debe entregar reportes estadísticos para toma de decisiones. Estos reportes, son enviados a la Vice-Rectoría, Rectoría y al Ministerio de Educación.

A pesar de la importancia de los reportes entregados, la Oficina Técnica de Estudios Avanzados, no posee una herramienta que permita administrar los datos de los alumnos pertenecientes a los programas de postítulo, por lo que esta tarea debe hacerse manualmente generando los problemas típicos que aparecen ante dicha situación.

Se puede considerar que no son pocos los problemas que se presentan al realizar reportes estadísticos contando registros manualmente, por otro lado, se puede apreciar que el tiempo empleado para dicha tarea es excesivo. Estos problemas y todos los demás serán descritos posteriormente.

La idea es crear un sistema capaz de administrar los datos de los alumnos pertenecientes a los programas de postítulo, que pueda manejar estos mismos programas y generar los reportes exigidos, minimizando el riesgo de error, disminuyendo el tiempo en la realización de las tareas y siendo una herramienta que permita apoyar todas las gestiones administrativas que la Oficina Técnica de Estudios Avanzados debe hacer ante los programas de postítulo y los alumnos pertenecientes a estos.

1.3.-FUNCIÓN PRINCIPAL DE LA ORGANIZACIÓN

La Oficina Oficina Técnica de Estudios Avanzados de la Pontificia Universidad Católica de Valparaíso (PUCV), tiene como función principal gestionar en forma eficiente el proceso académico y técnico -pedagógico de los alumnos y estudiantes de postgrado y postítulo, a

través de la planificación del quehacer técnico; coordinar y asesorar la ejecución, evaluar su desarrollo con el propósito de facilitar el logro de los objetivos propuestos por la Vice-Rectoría de Investigación y Estudios Avanzados. Debe además tener la capacidad de desarrollar criterios de diagnóstico y formulación de las estrategias adecuadas a la solución de conflictos en pos de los objetivos propuesto.

Otra función es el tener la capacidad estratégica para vincular el ambiente interno y externo, en pos de una constante actualización e intercambio multidisciplinario.

Para el cumplimiento de las funciones descritas, es que se deben cumplir ciertas tareas, como por ejemplo:

Hacer efectivas las normas emanadas por la autoridad para el desarrollo de las actividades de

índole técnico -pedagógico

Aplicar las normas y técnicas que permiten el desarrollo académico -curricular

Apoyar la gestión de los distintos estamentos

Recopilar información actualizada para retroalimentar el proceso técnico -curricular

En síntesis, en la Oficina Técnica de Estudios Avanzados se debe planificar, organizar, dirigir, ejecutar, supervisar y evaluar el desarrollo de las actividades curriculares - pedagógicas, realizando acciones de coordinación académica, orientadas a integrar, armonizar y facilitar el logro de las metas, en el ámbito de la eficiencia, eficacia y calidad de los procesos de formación académica, propuestos por la Vice-Rectoría de Investigación y Estudios Avanzados, con el fin de fortalecer la imagen institucional en los ámbitos de la investigación y la formación pluridisciplinaria.

El jefe de la Oficina Técnica de Estudios Avanzados debe entonces gestionar en forma eficiente y eficaz el proceso de formación académica y curricular de los alumnos y estudiantes de postgrado y postítulo, a través de la supervisión técnica permanente, con el fin de proveer calidad a este proceso, a la luz de los objetivos planteados para el logro de las

metas propuestas, que emanan del Plan de Desarrollo de la Vice-Rectoría de Investigación y Estudios Avanzados de la PUCV.

1.3.1.-Organigrama

En la Figura 1 se muestra el organigrama de la Pontificia Universidad Católica de Valparaíso, la Oficina Técnica de Estudios Avanzados pertenece a dicha institución, por lo que pasa a ser parte de dicho organigrama.

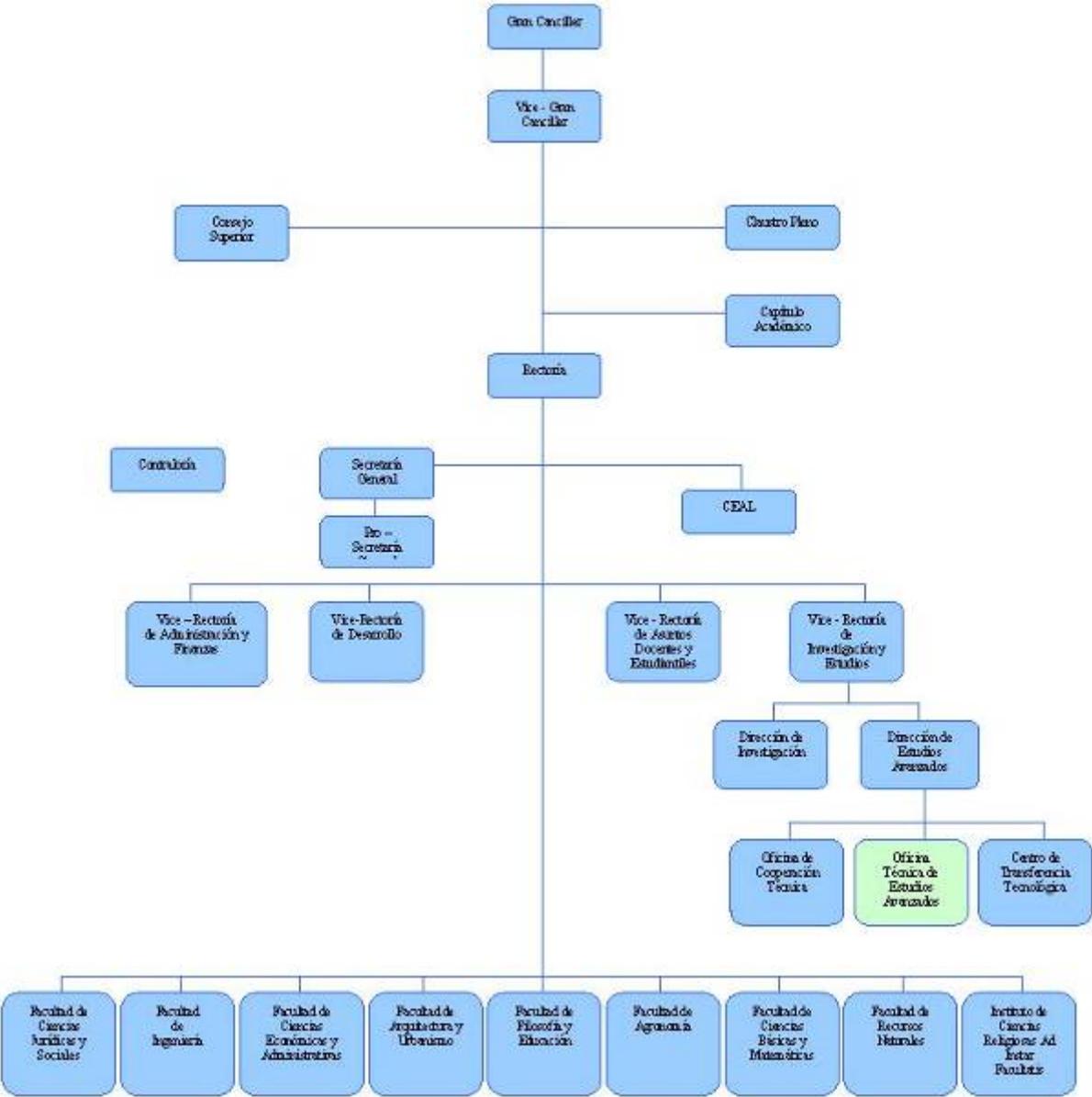


Figura 1. Organigrama

1.3.3.-Realidad Tecnológica

Actualmente, la Oficina Técnica de Estudios Avanzados no posee ningún sistema que permita tener el control de los datos de los alumnos de postítulo o de los programas que son impartidos.

La Oficina Técnica de Estudios Avanzados cuenta con un sistema computacional que permite el manejo de los datos de los alumnos de los programas de magíster y de doctorado, pero no posee un sistema de dichas características para los programas de postítulo. El sistema en cuestión se denomina Universis, y es un sistema que está implantado en la PUCV, y que maneja los datos de todos los alumnos de pregrado y postgrado, pero no así los programas y alumnos de postítulo.

El no tener una herramienta computacional que permita el desarrollo de dichas tareas produce que toda actividad realizada en esta dependencia, en cuanto a los programas de postítulo, sea realizada manualmente. Considerando que el tener control de estos datos lleva consigo un buen manejo de estos, se está ante la situación en que el manejo manual de los datos de un grupo de alumnos, de un conjunto de programas y de las actividades que se realizan con dichos datos, sea una tarea tediosa y muy sensible al error humano.

Las tareas que se realizan en la Oficina Técnica, van desde el manejo de los datos de los alumnos matriculados, hasta la generación de reportes y estadísticas las cuales son realizadas mediante un conteo manual de los datos para su posterior manejo, validación y ayuda a la toma de decisiones.

En la Oficina Técnica de Estudios Avanzados hay computadores de escritorio con acceso a Internet, pero dichos computadores no prestan ninguna utilidad al momento de realizar tareas con los programas de postítulo, dado que estas, como se mencionó anteriormente, son realizadas a mano, y llevadas al computador, sólo para llevar este conteo a planillas Excel con las cuales se desarrollan los informes deseados.

1.4.-PROBLEMAS

Falta de control sobre los datos de los alumnos

Todos los datos de los alumnos son registrados en papeles y guardados en archivadores. Estos archivadores están ordenados por año, y a su vez, internamente están organizados por programa de Postítulo. Es por esto que en caso de necesitar los datos de un alumno específico, es necesario saber el año de ingreso al programa, el programa al que ingresó, además de algún dato personal para su búsqueda. Un caso práctico de este problema se da al querer conocer el año de egreso de un alumno específico. Para saber el año de egreso, es necesario conocer el año de ingreso al postítulo para buscar en el archivador correspondiente a ese año. Si se desconoce el año de ingreso, habría que buscar en todos los archivadores hasta encontrar al alumno y ver el dato requerido.

Administración manual de los datos

Esto ocurre cada vez que ingresa un alumno a un programa de postítulo, congela, egresa o por motivos académicos no puede seguir cursando el programa. Esta información es entregada por externos (por ejemplo, docentes) a la Oficina Técnica de Estudios Avanzados con el fin de que aquí se administren los datos de los alumnos. En la Oficina estos datos son llevados a los archivadores correspondientes y las distintas gestiones que se deben hacer con los datos se realizan en forma manual.

Conteo manual de los datos

Con el fin de administrar los datos de los alumnos pertenecientes a los distintos programas de postítulo, los datos de éstos son contados manualmente. Dependiendo de la información que se necesite conocer, este proceso se realiza extrayendo de los archivadores los datos de los alumnos y anotándolos en una hoja aparte para un posterior conteo. Un ejemplo de este caso, ocurre al querer conocer la cantidad de alumnos pertenecientes a un Programa de Postítulo en un año determinado. Para esto se debe tomar el archivador correspondiente al año en cuestión, ubicar el programa deseado y contar manualmente los alumnos matriculados para ese periodo y programa.

Lentitud en la gestión de Reportes.

Este problema es consecuencia del anterior. Los reportes estadísticos entregados por la Oficina Técnica de Estudios Avanzados consisten en estadísticas de distintos tipos, las cuales son entregadas a Rectoría o al Ministerio de Educación en la medida que alguna de estas entidades lo requieran. Estos reportes, pueden considerar distintos parámetros para su realización, como por ejemplo, sexo de los alumnos, programas de estudios, nacionalidad, cantidad de matriculados y egresados, etc., datos que son contados manualmente para el desarrollo de las estadísticas. Esto produce que, ante la petición de un reporte, los resultados tiendan a demorarse, dado que la generación de las estadísticas toma demasiado tiempo. Los datos son contados uno a uno de los archivadores correspondientes al rango de años solicitados. Estos datos se suman para luego ingresarlos a planillas Excel para la creación de gráficos.

Reportes poco confiables

Si bien estos reportes se realizan de manera responsable y procurando no cometer errores, hay que reconocer que muchas veces estos errores pasan de manera imperceptible, por lo que, a pesar de tener mucha cautela, pueden contarse más datos que los que realmente son, o no fijarse en algún detalle que pueda producir un cambio dentro de cierto reporte. El hecho es que al realizar un conteo manual de datos, considerando además todos los otros datos que influyen en la creación del reporte, pueden ocurrir una serie de errores los que llevarán consigo la entrega de resultados no reales e/o inconsistentes.

Por ejemplo, si se tiene una lista de alumnos egresados en un determinado año, y además se tiene otra lista clasificada en hombres y mujeres egresados en ese mismo año, y la suma de hombres y mujeres no concuerda con la información dada por la otra lista, no se sabe cual de los dos reportes tiene la información correcta, provocando inconsistencia en la información brindada.

Pérdida de recursos

Este problema nace como consecuencia del anterior. Si se llega a detectar errores en algún reporte, es absolutamente necesario volver a realizarlo. Esto lleva consigo el gasto inadecuado

de los recursos para la creación de dicho reporte. Por ejemplo, si se considera el ejemplo del problema anterior, se hace necesario llevar a cabo nuevamente todos los reportes en cuestión, con el fin de asegurar confiabilidad en los resultados. La creación de estos nuevos reportes, lleva consigo tiempo de trabajo y gasto de materiales de la Oficina Técnica de Estudios Avanzados. El volver a realizar manualmente un reporte, no asegura que la información dada sea correcta, debido a que en las distintas iteraciones se pueden ir cometiendo nuevos errores.

Acceso difícil y lento a la información

Debido a que toda la información se encuentra registrada en los archivadores, al momento de acceder a los documentos específicos de un alumno, resulta difícil e incómodo revisar y encontrar la información que se necesite, provocando desorden, poca organización y demora en las tareas que se realizan en la Oficina Técnica de Estudios Avanzados. A pesar de que los datos se pueden manejar de esa manera, esta tarea es engorrosa y poco eficiente.

Poca seguridad de los datos

Como se ha dicho anteriormente, todos los datos referentes a los programas de postítulo, como los datos respectivos a los alumnos pertenecientes a dichos programas, son guardados en archivadores, lo que provoca poca seguridad ante cualquier evento inesperado, ya sea accidentes involuntarios (como por ejemplo un incendio) o voluntarios (como un posible robo de información).

1.5.-DEFINICIÓN DE OBJETIVOS

1.5.1.-Objetivo general

Desarrollar un sistema que permita la administración segura y confiable de los datos de los programas de postítulo y de los alumnos matriculados en estos.

1.5.2.-Objetivos Específicos

Mantener información segura de los alumnos matriculados en programas de postítulo, controlando los accesos a dicha información y asignando permisos específicos a usuarios autorizados.

Permitir el ingreso de los datos de alumnos matriculados en un determinado programa de postítulo.

Generar listas y estadísticas de distintos tipos con la información que maneja el sistema.

Permitir la administración de programas de postítulo, asignaturas, profesores y alumnos, con el fin de poder alimentar el sistema con todos los datos necesarios para su funcionamiento.

Permitir la visualización de los datos de alumnos pertenecientes a programas de postítulo, basado en distintas formas de búsqueda de datos.

Creación de distintos perfiles para los usuarios del sistema.

1.6.-SOLUCIÓN TECNOLÓGICA PROPUESTA

Se propone desarrollar un sistema que permita administrar los datos de los alumnos pertenecientes a los distintos programas de postítulo que son impartidos por la Universidad, centrándose en la creación de reportes estadísticos, tomando para su generación distintos parámetros con el fin de poder generar cualquier tipo de estadísticas.

Por otro lado, el sistema permitirá tener el control de todos los datos y almacenarlos de manera eficiente para poder evitar cualquier tipo de problemas con estos. El hecho de que un sistema de computación tenga los datos almacenados eliminará la mayoría de los problemas existentes en la Oficina Técnica de Estudios Avanzados respecto a los programas de postítulo, dado que no se necesitará contar manualmente los datos. Por otra parte, el sistema asegura una mayor confiabilidad sobre los resultados entregados en cuanto al manejo estadístico de los datos.

Se hace necesario desarrollar un método de búsqueda con el fin de tener mayor conocimiento de los datos de cada alumno, las listas de alumnos de los distintos programas y la lista de los

distintos programas existentes en una fecha dada, y que la administración de dichos programas se base en las normas establecidas por el decreto que está impuesto para él. La finalidad de crear un sistema de búsqueda, es poder manejar el problema de un acceso difícil y lento a la información.

Se administrarán los distintos programas de postítulo, con el fin de poder ingresar nuevos programas, si es que estos surgen, modificar los ya existentes o cancelar algún programa en el caso que éste dejara de dictarse.

Se propuso al cliente la creación de una interfaz para el alumno, con el fin de que éste pueda ver cierta información referente a su avance curricular, y donde se puedan modificar ciertos datos. Por otra parte, se le ofreció al cliente la posibilidad de dar acceso a los profesores para hacer uso del sistema para poder ingresar las notas de los alumnos, calcular promedios y cerrar actas, esto con el fin de evitar que los datos deban ser ingresados por la Oficina Técnica de Estudios Avanzados, y así ésta se preocupe solamente de generar los reportes y administrar un sistema, que se alimenta “automáticamente” sin intervención de las personas que trabajan en dicha dependencia.

El sistema a desarrollar debe tener una interfaz amistosa que permita la facilidad de uso, con el fin de brindar eficiencia a las tareas realizadas en la Oficina Técnica de Estudios Avanzados.

Dicho de otra manera, el sistema realizará de manera automatizada las tareas que se realizan frecuentemente en la Oficina Técnica de Estudios Avanzados, en lo que respecta a los programas de postítulo, considerando que estas tareas se realizan actualmente de forma manual, por lo que es considerable el beneficio que la implantación de este sistema traerá dado que permitirá a la Oficina Técnica de Estudios Avanzados realizar sus funciones normales pero con mayor eficiencia, eficacia y certeza en los datos que maneja.

1.6.1.-Descripción de la Solución

La solución tecnológica que se llevará a cabo esencialmente consiste en que mediante el uso de ambiente Web puedan realizarse todas las tareas que ahora se ejecutan manualmente, ya sea

en cuanto a estadísticas o el simple hecho de generar la lista de un Programa de Postítulo. Para poder lograr este objetivo se utilizará el servidor disponible en la Escuela de Ingeniería Informática de la PUCV y el motor de bases de datos brindado por la misma Escuela (SQL Server), además de los espacios que dispone la Universidad en la Web, los computadores de escritorios que posee el cliente. También se consideró el manejo de herramientas de licencias gratuitas que se tienen como JAVA y su servidor JBOSS, para el manejo de paginas JSP.

Como las estadísticas que se deben generar son muy variadas, se llevará a cabo la creación dinámica de estadísticas, esto quiere decir que el usuario podrá elegir, dependiendo de la estadística que necesita, los datos sobre los cuales ésta se generará, ya sea sexo, año ingreso al Programa de Postítulo, etc.

Debido a que el software será desarrollado en ambiente Web, nace la posibilidad de que el usuario de sistema pueda ejecutar la tarea que necesite desde cualquier punto físico, ya que sólo se necesitará un PC con conexión a Internet para poder llevar a cabo cualquier interacción con el sistema.

En síntesis, lo que se va a desarrollar, será un sistema Web que permitirá a los usuarios, mantener un control y mayor conocimiento de los datos, generar reportes estadísticos en una plataforma que permita la gestión de las estadísticas que el usuario desee, además de una interfaz de búsqueda para poder manejar alumnos, programas, etc. Se integrará una interfaz para que los alumnos puedan ver sus fichas académicas y una interfaz en la que los profesores podrán ingresar el avance curricular de los alumnos a modo de alimentar el sistema. Todo esto se realizará usando la tecnología nombrada anteriormente.

1.6.1.1.-Arquitectura Física

Por el hecho de tratarse de un sistema Web, lo necesario para el desarrollo y funcionamiento del sistema, son recursos de fácil acceso. Para el usuario final, sólo se necesitará un computador de Oficina que tenga una conexión a Internet.

El servidor a ocupar debe tener soporte para SQL Server, en el caso del almacenamiento de los datos, y debe tener integrado un servidor para páginas JSP, para la interacción entre el sistema y la base de datos.

La Figura 2 muestra de forma gráfica, la arquitectura física que será implementada en el sistema.

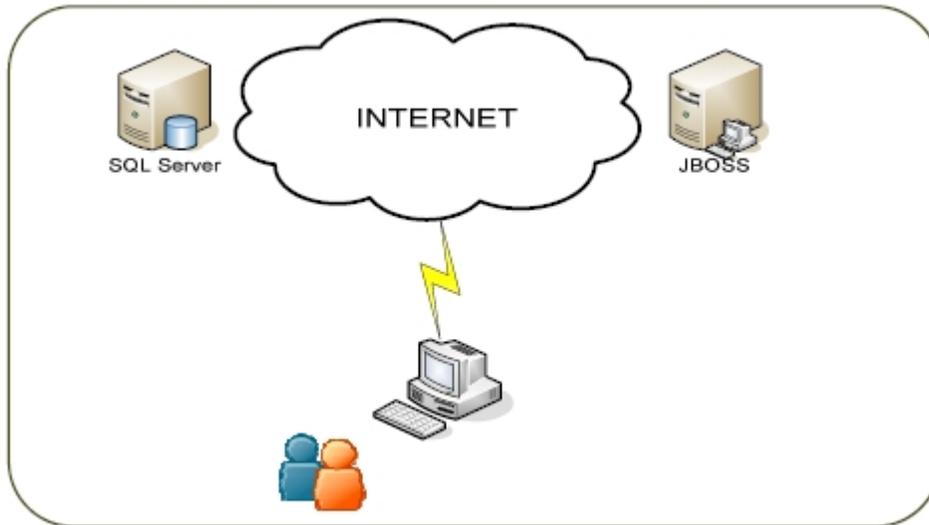


Figura 2. Arquitectura Física

1.6.1.2.-Arquitectura Lógica

Se utilizará una arquitectura lógica de 3 capas, como lo muestra la Figura 3. Cada una de las distintas capas tiene una función específica dentro del sistema. Dichas funciones son descritas posteriormente.

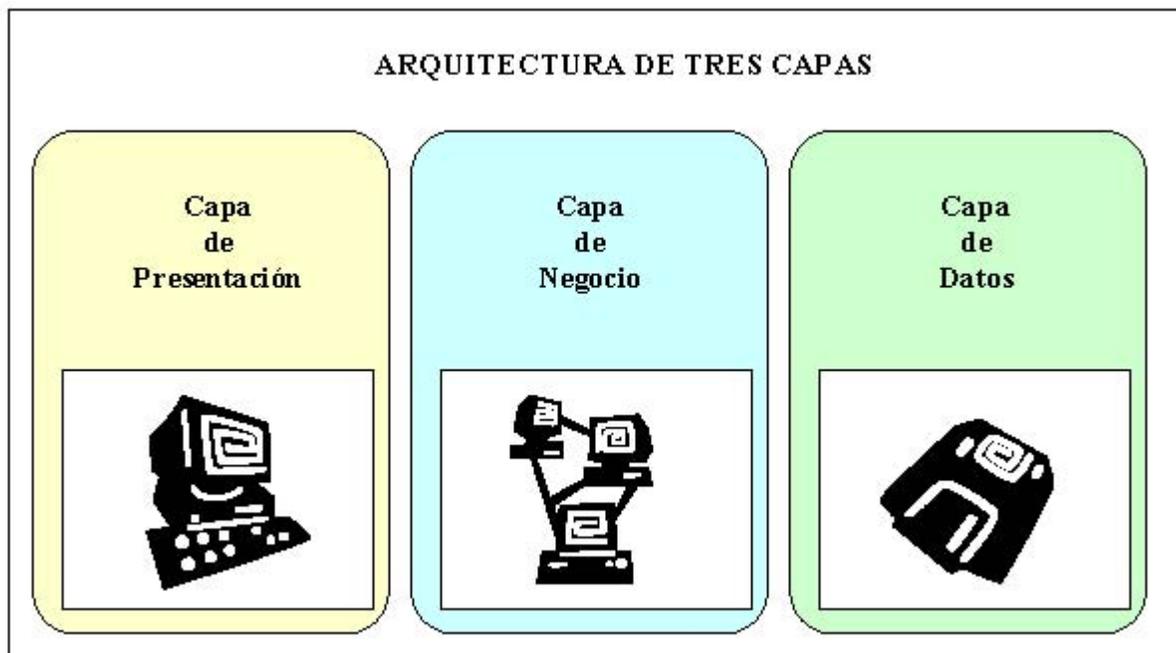


Figura 3. Arquitectura de Tres Capas

Capa de presentación: Modo en que el usuario podrá interactuar con el sistema desarrollado. Para este caso, acá se presentan las distintas interfaces con las que el usuario podrá ingresar datos de alumnos, ingresar distintos programas de postítulo, generar reportes, ver datos de alumnos, entre otros. Esta capa es la que permitirá al usuario realizar todas las funciones sobre el sistema.

Capa de negocio: Dice relación a las interacciones que existen entre la base de datos y el sistema, es decir, todo lo que tiene que ver con transacciones sistema/base de datos, ya sea procesos de inserción, modificación y/o eliminación en la base de datos, incluyendo también las consultas y el trabajo que se realiza con la información antes de ser desplegada en pantalla. Dentro de esta capa es donde cae el servidor JAVA, en este caso específico JBOSS, ya que éste es el encargado de ejecutar la comunicación sistema/base de datos.

Capa de datos: Es la capa donde se encuentra el administrador de base de datos, el cual se encarga de mantener la integridad física de los datos, o sea el almacenamiento seguro y eficaz de estos. En este caso se utilizará SQL Server.

En la Figura 4, se muestra la arquitectura lógica del sistema, identificando los módulos en la capa de presentación, la interacción de estos con la capa de negocios y a su vez el manejo de la capa de datos. Todas estas capas están acompañadas transversalmente de la comunicación y seguridad que está entre las 3 capas del sistema.

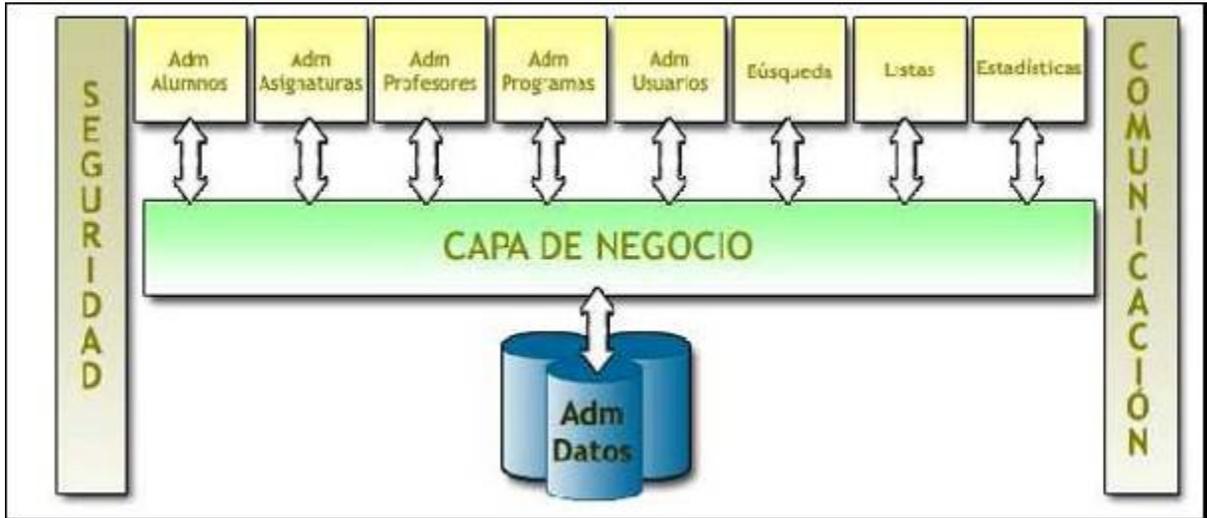


Figura 4. Arquitectura lógica de SAPP

1.7.-ESTUDIO DE FACTIBILIDAD

Dentro de cualquier proyecto a desarrollar, existen una serie de limitantes, las cuales es necesario evaluar para poder comprobar si es posible llevar a cabo el proyecto o no. Es probable que estas limitantes sean de una índole importante, lo cual impida realizar el proyecto y tener que buscar soluciones alternativas que permitan resolver el problema.

Entre estas limitantes, se puede encontrar el problema económico, dado que es necesario saber si se cuenta con el financiamiento necesario para poder desarrollar una solución tecnológica particular. Por otro lado, hay que ver si en la organización existe alguna regla o política que impida el desarrollo normal del proyecto, esto es conocido como la parte operacional, lo cual no es nada más que determinar si es posible llevar a cabo el proyecto basado en dicho ambiente organizacional. Está también presente el aspecto legal, considerando que no existe ningún resquicio legal que determine que el proyecto no sea realizable. Por último, se puede considerar como otra limitante, la parte tecnológica o técnica

del proyecto, ver si cuenta con los recursos tecnológicos necesarios para la implantación y explotación del sistema, considerando en este punto tanto software como hardware.

Para poder determinar si es posible el desarrollo de este proyecto, es que se hizo un análisis para cada una de las situaciones nombradas anteriormente, haciendo la evaluación del proyecto en distintos escenarios, con el fin de poder analizar más detalladamente la factibilidad de la realización del proyecto.

1.7.1.-Factibilidad Técnica

La idea de este punto es determinar si es posible llevar a cabo la solución tecnológica propuesta que permite solucionar el problema existente en la institución, verificando si se cuenta con las herramientas, y la tecnología necesaria para resolver exitosamente el problema planteado.

Cabe destacar, que este es un aspecto muy importante dentro de lo que es el proyecto, ya que sin las herramientas necesarias en la institución no sería posible el desarrollo y la ejecución del sistema.

Técnicamente la factibilidad de implementación es de un 100%, ya que la Escuela de Ingeniería Informática de la PUCV, facilitará la gran mayoría de las herramientas que se utilizarán en el desarrollo del proyecto, mientras que las restantes son de fácil obtención y gratuitas. Además se cuenta con el conocimiento necesario por parte del equipo, para la utilización de dichas herramientas.

1.7.2.-Factibilidad Operacional

La factibilidad operacional cae en medir si la ejecución del sistema es realizable, dependiendo de la organización de la institución.

El sistema se considera factible operacionalmente, si es que existe el personal apropiado para manejar lo que es el sistema en sí. Personal capacitado, el cual tiene acceso a un manual de usuario del sistema, además de trabajar sobre interfaces sencillas y explícitas, por ende, el nivel que se requiere para la utilización de éste sistema es de un grado básico de computación,

sumando a esto, se capacitará a cada persona que utilice el sistema, para que así éste pueda aprovecharlo de mejor maneja y obtener mayor eficiencia.

Además, se trabajará conjuntamente con el cliente, capacitando a éste de antemano para conocer de lleno como será el sistema, esto lo llevará a saber muy bien de que manera trabaja y como sacarle mayor partido a éste, con esto no tendrá problemas en el momento de su utilización. Cabe destacar, que el cliente en si será uno de los principales usuarios del sistema, por ende, se dará un gran énfasis en que este lo conozca de muy buena manera.

1.7.3.-Factibilidad Legal

Lo que abarca la factibilidad legal, es todo lo que tiene que ver con los impedimentos legales que restrinjan llevar a cabo la ejecución del sistema, ya sea leyes o decretos.

Tomando en cuenta lo que es la factibilidad legal, se observa que no existen impedimentos legales que prohíban el buen desempeño y funcionamiento del software, ya que no se pasa a llevar ninguna ley vigente, como por ejemplo:

Ley N° 17.336 la cual está relacionada con la propiedad intelectual, el artículo 41 específicamente dice relación con adaptaciones o copias del proyecto computacional, pues en este proyecto no existe ningún tipo de copia, ya sea de código fuente, interfaces, etc.

Ley N° 19.223 la cual tipifica figuras penales relativas a la informática, pues el software no daña, ni altera algún sistema de información.

1.7.4.-Factibilidad Económica

La institución a la cual se le desarrollará el sistema posee todas las tecnologías que se requieren para la implementación del software, por ende, todo lo que está relacionado a lo que es hardware se tiene, ya sea computadores de escritorios, servidores, redes, hosting, dominio, etc.

Además, se trabajará con lenguajes, motores de BD y herramientas que la institución ya tiene, por ende tiene sus licencias, como sería el sistema gestor de base de datos SQL, herramientas

CASE como Rational Rose, sumando a esto se utilizarán herramientas de desarrollo gratuitas como son Java y su servidor JBOSS (que es un servidor de páginas Java), para con esto poder trabajar con páginas JSP.

Viendo lo anterior, se puede observar que la factibilidad económica no es un impedimento para la puesta en marcha del sistema, ya que la institución no debe incurrir en ningún gasto, por consiguiente, la ejecución del sistema es factible económicamente en un 100%.

Por todos los puntos vistos anteriormente (Factibilidad Técnica, Operacional, Legal y Económica), se considera factible el desarrollo del proyecto.

CAPÍTULO 2: PLANIFICACION

2.1.-INTRODUCCIÓN

Es necesario realizar una planificación del proyecto en desarrollo, con el fin de comprender las distintas etapas y asegurar un orden en el transcurso de éstas. La idea, es poder basarse en algún paradigma de desarrollo y seguir sus pasos de una manera que se adapte al sistema a desarrollar.

La planificación es crucial en el desarrollo de un sistema dado que se puede saber qué es lo que se va a realizar, sin embargo es necesario conocer de qué manera se va a llevar a cabo cada una de las tareas, no desde un punto de vista técnico, sino que de un punto de vista de organización.

En este punto se decide que modelo de desarrollo de software se utilizará. Existen distintos modelos de desarrollo de software. Para la elección del paradigma a seguir, se consideró el desarrollo del sistema siguiendo los distintos paradigmas, esto con el fin de poder distinguir las características del modelo que más se adapta al sistema a desarrollar. Se mostrarán entonces, las principales características de algunos paradigmas de desarrollo, para poder comprender de mejor manera cuáles son los motivos de la elección del paradigma utilizado para la realización del proyecto.

En esta etapa se planifica también el tiempo de duración de las distintas actividades, tareas y etapas (dependiendo del paradigma) que tenga el proyecto, realizando la asignación de tiempos necesarios para el buen desarrollo de las distintas actividades tanto del proyecto como de las académicas.

2.2.-PARADIGMAS DE DESARROLLO DE SOFTWARE

2.2.1.-Modelo en Cascada

Este modelo es conocido también como Ciclo de Vida Clásico del software. El enfoque que este modelo le da al desarrollo es realizar las actividades en orden consecutivo, teniendo que finalizar una actividad antes de comenzar con la siguiente. Las actividades o etapas son *análisis y definición de requerimientos*, donde se definen las metas del sistema basado en la comunicación con los clientes, con esto se especifica lo que el sistema debe hacer; *diseño*, donde se dividen los requerimientos obtenidos en la primera etapa y se diseña o establece la arquitectura del sistema; *implementación y pruebas*, en esta etapa se llevan a cabo los programas que resultaron de la fase de diseño. Estos programas deben probarse con el fin de que cada uno haga lo que debe hacer y lo realice correctamente. *Integración y prueba* es donde se integran todos los programas realizados en la etapa anterior para luego probarse como un conjunto asegurándose el correcto funcionamiento del sistema y del cumplimiento de todos los requerimientos. La última etapa corresponde a la *operación y mantenimiento*, en esta fase se instala el programa y se hace uso de éste. El mantenimiento se refiere a corregir posibles errores que puedan surgir durante el uso del sistema y que no fueron descubiertos en las etapas anteriores [PRESS].

Es probable que en la última etapa se encuentren errores o nuevas funcionalidades del sistema que no fueron cubiertas, por lo que es necesario realizar nuevamente todas las actividades. El problema de esto es que las iteraciones son costosas dado que cada etapa es documentada y aprobada nuevamente, además de volver a realizar el trabajo, por lo que los costos en tiempo y dinero son altos.

Ventajas

Si los requerimientos son bien especificados y entendidos en su totalidad, el sistema tiene muy buenos resultados.

Es fácil planificar las distintas etapas del desarrollo dado que están definidas.

Este modelo es la base para los otros paradigmas.

Desventajas

En la realidad es muy difícil seguir este modelo, dado a que es poco habitual que los proyectos sigan una secuencia estricta. Es más común encontrarse con proyectos en que las etapas se solapan.

Sólo se ve resultados al final del proyecto.

Si se hizo algo que no era lo que el cliente deseaba, esto se descubre una vez que el sistema está terminado o próximo a su finalización.

Difícil manejar los cambios de requerimientos por parte del cliente. Como los requerimientos son tomados en la etapa inicial del proyecto, es difícil tomar más requerimientos o modificar los existentes, durante las etapas posteriores.

Si se descubre un error de manera tardía, su solución es muy costosa.

2.2.2.-Desarrollo Evolutivo

La idea de este paradigma es desarrollar una implementación inicial y mostrársela al cliente (o usuario) con el fin de que éste pueda hacer sus objeciones y comentarios para ir refinando esta implementación inicial, a lo largo del tiempo. Las actividades se llevan a cabo en conjunto, es decir, la especificación, desarrollo e implementación se realizan juntas y entre ellas se retroalimentan de manera rápida [**PRESS**].

Dentro del desarrollo se pueden encontrar dos modalidades de trabajo. Una es trabajar junto al cliente en el desarrollo del sistema, haciendo evolucionar este mismo con nuevos aportes dados por el cliente hasta llegar a un resultado final, esta modalidad es conocida como desarrollo exploratorio. La otra manera es tomar los requerimientos dados, realizar una definición mejorada de los mismos y desarrollar un sistema basado en estos, experimentando en aquellos requerimientos que no son del todo entendibles. Por este motivo, una vez desarrollado esto, se muestra al cliente, el cual decide si lo que se realizó es acorde a lo que él deseaba, en caso contrario se debe desarrollar nuevamente esa parte, para ver si se logró entender los requerimientos hasta llegar a una solución deseada. Esta segunda modalidad se conoce como prototipos desechables.

Las etapas del desarrollo evolutivo, sea cual sea la modalidad que se este usando, son las siguientes:

Especificación inicial: Se toman los requerimientos del cliente para poder realizar el trabajo.
Desarrollo del producto: Se desarrolla un producto basado en las especificaciones dadas por el cliente al inicio (o durante esta etapa de desarrollo).
Implementación, uso y evaluación: Se entrega el producto al cliente, para que éste pueda utilizarlo y determinar si el sistema cumple con las especificaciones y satisface las necesidades que se tienen. A partir de este punto, pueden surgir dos etapas o hitos: la re-especificación en caso que el cliente quiera hacer modificaciones al producto entregado por no ser lo que este quería o por cualquier otro motivo que implique un nuevo desarrollo; y las distintas iteraciones del sistema, debido a que a medida que se va iterando, van surgiendo las distintas versiones del software, hasta llegar a la última versión de éste, la que cumple con todos los requerimientos del cliente.

La Figura 5 grafica las distintas etapas del desarrollo evolutivo, mostrando la manera en que avanza e itera el flujo hasta llegar a la versión del software deseada que cumple los requerimientos establecidos por el cliente.

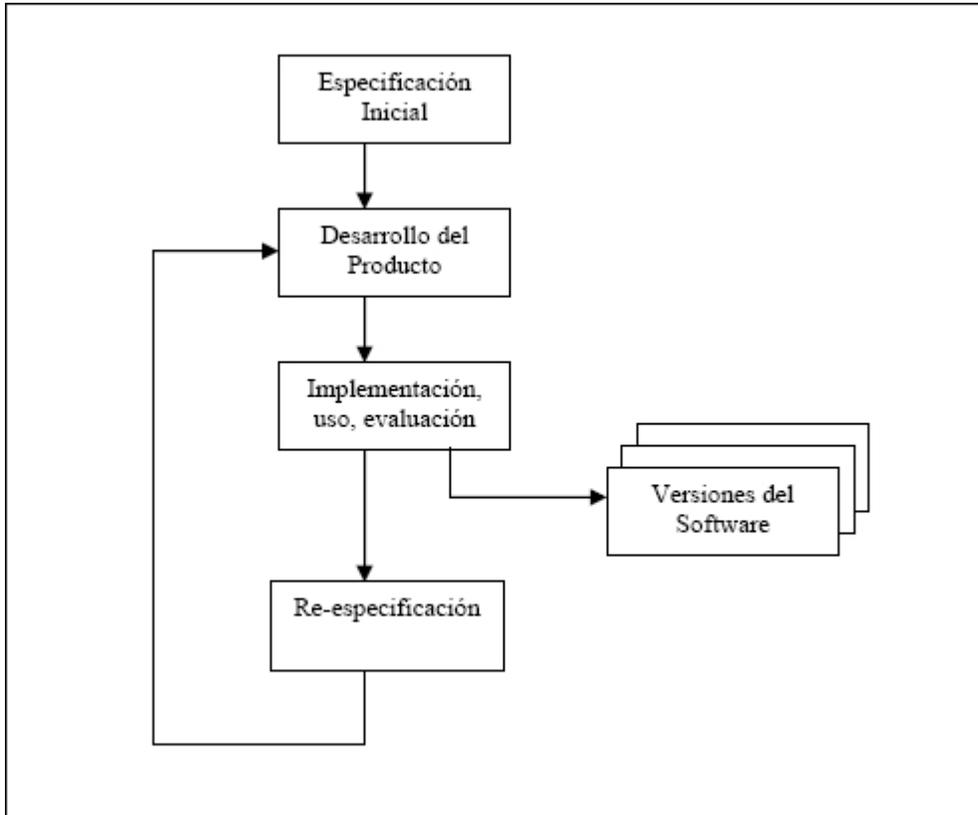


Figura 5. Etapas del Desarrollo Evolutivo

Ventajas:

Muy efectivo para proyectos pequeños.

Funciona bien si el cliente con el desarrollador llegan a un buen entendimiento inicial en cuanto a la modalidad de trabajo, dejando en claro que se trabajará con prototipos y que las distintas entregas no serán una versión final, sino que serán iteraciones para refinar lo que será la última versión del producto. De no quedar esto claro, este punto se convierte en una desventaja para este paradigma.

Asegura el cumplimiento total (o en una gran cantidad) de los requerimientos, dado que el cliente puede ir viendo lo que se hace, realizar acotaciones o nuevos requerimientos, con el fin de que el sistema entregado, sea lo más cercano a lo especificado.

Desventajas:

No es posible ver el proceso. Es difícil documentar todo si se trabaja de manera rápida, por lo que se desarrolla e implementa sin una documentación que pueda reflejar las distintas versiones del sistema.

El cliente puede no entender que lo que se le entrega es un prototipo, pensar que es el producto final y desmoralizarse por tener un sistema que no cumple con los requerimientos exigidos.

2.2.3.-Desarrollo en Espiral

Este paradigma consiste en mostrar el proceso de desarrollo del software como una serie de actividades donde una continúa de la anterior, y que se representa como un espiral. Las etapas internas del espiral, pueden ser las etapas iniciales y así ir aumentando a medida que se va avanzando en el desarrollo del sistema. Cada ciclo se divide en 4 áreas: definición de objetivos, en donde se definen los objetivos específicos, las restricciones y se desarrolla un plan de administración; la segunda etapa es la evaluación y reducción de riesgos, donde se detallan los riesgos que puede seguir el proyecto y el plan de contingencia para identificar, estimar y ver las formas de gestionar el riesgo, buscando alternativas de solución en caso de presentarse éstos. La tercera etapa, es la de desarrollo y validación, donde se elige un modelo para el desarrollo del sistema. Por último, la cuarta etapa, la de planeación, es aquella en que el proyecto se evalúa y se decide si se debe continuar con el proyecto, tomando las consideraciones para seguir con el siguiente ciclo del espiral.

Ventajas

A diferencia de los otros paradigmas, el riesgo se considera de manera explícita para cada una de las etapas, conduciendo con esto a una mejor planificación y disminución de los riesgos.

Útil para proyectos pequeños.

Desventajas

Es necesario controlar muy bien el riesgo. En caso de no ser así, riesgos pequeños pueden transformarse en problemas mayores a medida que avanzan las distintas etapas del proyecto.

Puede considerarse más complejo, dado que no existen fases fijas. Este modelo puede contener otros modelos. Esto puede ser una gran dificultad dado que la capacidad de trabajar con más de un modelo en un proyecto tiene estrecha relación con la experiencia y conocimientos que tenga el equipo que lleva a cabo el proyecto.

2.2.4.-Modelo de Métodos Formales

Este modelo pretende especificar el sistema por medio de actividades matemáticas, es decir, aplicar notación de tipo matemática para especificar, desarrollar y verificar un sistema. El uso de herramientas matemáticas rigurosas puede ayudar a eliminar muchos problemas que presenten otros paradigmas, dado que mediante estas actividades se puede corregir o evitar la ambigüedad, las inconsistencias, etc. [WEB7].

Las herramientas matemáticas utilizadas, permiten verificar programas, detectar errores y corregirlos.

Ventajas

Asegura un índice bajo de errores, dado que el desarrollo del sistema está basado sobre una sólida base de actividades matemáticas.

El tener herramientas matemáticas puede ser muy efectivo dado que no permite intuición (y por ende posibles errores) como los demás paradigmas de desarrollo de software.

Desventajas

Toma demasiado tiempo.

Costos elevados.

No existe tanta información respecto a este paradigma, por lo que su utilización requiere un estudio más minucioso en comparación a los otros paradigmas de desarrollo.

Si el cliente no tiene mucho conocimiento técnico, este modelo se hace difícil de ocupar.

2.2.5.-Proceso Unificado de Desarrollo

El Proceso Unificado es un proceso de desarrollo de software configurable que se adapta a proyectos que varían en tamaño y complejidad. Se basa en muchos años de experiencia en el uso de la tecnología de objetos en el desarrollo de software de misión crítica en una variedad de industrias [WEB12].

El Proceso Unificado guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado mientras se balancean los requerimientos del negocio, el tiempo utilizado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requerimientos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requerimientos y a la arquitectura. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones. El Proceso Unificado es soportado por herramientas que automatizan entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

El ciclo del Proceso Unificado consta de cuatro etapas: Inicio, donde se planifica el proyecto; Elaboración, donde se establece un plan y una arquitectura acorde a las necesidades del sistema; la etapa de Construcción donde es desarrollado el sistema y la etapa de Transición, en la cual el sistema es proporcionado a los usuarios finales.

Ventajas

Puede ser iterativo o lineal, dependiendo del tamaño del sistema.

Permite trabajar con un enfoque dirigido a la utilización del sistema (casos de uso).

Permite identificar y gestionar riesgos.

Se centra en la arquitectura del sistema.

Desventajas

La documentación y diseño de diagramas para el desarrollo del proyecto puede provocar un aumento de los tiempos para llevar a cabo el sistema si estos no son considerados y asumidos en etapas iniciales del proyecto.

No es fácil la planificación del proyecto mediante este paradigma.

2.2.6.-Selección del Paradigma a utilizar

Con la idea de hacer la mejor elección de un paradigma de desarrollo para el sistema, es que se tomó el sistema y evaluó su desarrollo mediante los distintos paradigmas nombrados anteriormente. La idea de esto, era ver el comportamiento del sistema ante las diversas características, ventajas y desventajas de cada paradigma. De esta manera se pudo ver que características facilitan el desarrollo del sistema en cuestión y cuales lo complican o simplemente no son útiles para el desarrollo por lo que hacen que el paradigma sea inutilizable para este caso.

A continuación se nombran las características de los paradigmas, pero a diferencia de la descripción anterior, acá se nombrarán las características que sirven y aquellas que no sirven para que el proyecto sea desarrollado bajo cada paradigma.

La idea de este punto es resaltar las características más destacadas del paradigma y poder ubicar falencias difíciles de encontrar a simple vista. Conociendo la manera en que se daban a conocer dichos aspectos, se puede lograr identificar cual es el paradigma de desarrollo que se comporta de mejor manera para llevar a cabo el desarrollo del sistema.

Al comparar el proyecto con los distintos paradigmas, destacan las características que se nombran a continuación:

Modelo en cascada

Este paradigma muestra una serie de características que no hacen posible la realización deseada para el proyecto. Puede que ofrezca virtudes, pero son tantas las falencias que presenta que se decidió no realizar el sistema basado en este paradigma. Estas falencias son las siguientes:

Sólo muestra los resultados al final del proyecto, lo cual es un punto en contra dado que el cliente quiere estar en amplio contacto con el desarrollo del sistema, por lo que quiere ir viendo avances y resultados en la medida que pasa el tiempo, por lo que, tener resultados sólo al final del proyecto no permite manejar el desarrollo del sistema de la manera requerida.

El cliente, a pesar de haber solicitado ciertos requerimientos, dejó abierta la posibilidad de seguir haciendo más requerimientos en la medida que pase el tiempo y él vaya reconociendo más falencias del sistema actual que puedan ser solucionadas. Esto no es manejable por este modelo, dado que los requerimientos son hechos al inicio del proyecto y se trabaja con éstos hasta el final.

Por estos motivos se decidió no realizar el proyecto siguiendo este modelo.

Desarrollo Evolutivo

Este modelo muestra ciertas características que podrían ser útiles para el desarrollo del sistema. Estas principalmente están dadas por la capacidad que da el desarrollo evolutivo de incluir en gran porcentaje la participación del cliente durante todas las fases del proceso. Por otro lado permite incluir requerimientos a lo largo del proyecto.

Los inconvenientes que se produjeron al simular el sistema bajo este paradigma son los siguientes:

Bajo este modelo no se realiza tanta documentación, sino que el trabajo se realiza concentrado más en la creación de los prototipos. El cliente solicitó la documentación del proyecto, con el fin de tener la información del sistema que pueda ser útil a futuro para fines de mantención y actualizaciones.

Al conocer los gustos y exigencias del cliente, es fácil darse cuenta que ante un prototipo, éste piense que es el sistema final y en vez de aportar con los cambios que se deben hacer, puede pensar que el sistema no cumple con lo que él esperaba y desilusionarse.

Desarrollo en Espiral

El desarrollo en espiral brinda características muy útiles respecto al tratamiento del riesgo por lo que apoya a un desarrollo de sistemas que contempla las contingencias que puedan existir y con esto prever una solución anticipada a posibles problemas. Con esto ayuda al manejo y cumplimiento de los tiempos programados, debido a que permite saber qué acciones se deben llevar a cabo ante situaciones inesperadas permitiendo a los desarrolladores tomar las medidas necesarias ante algún imprevisto, para lograr los objetivos planteados de la manera programada.

En realidad, este modelo presta una gran utilidad para el desarrollo del proyecto, pero presenta el inconveniente de que el modelo no presenta fases fijas, es probable pasar por una misma etapa reiteradas veces con el fin de conseguir que ésta quede bien definida para pasar a la siguiente.

Cabe destacar que la experiencia de los desarrolladores bajo este paradigma es baja, y el no conocer herramientas para el uso de este modelo puede traer pérdida de tiempo, no cumplir con los objetivos planteados y no poder realizar el sistema de la mejor manera posible.

Otra desventaja es la poca comunicación con el cliente. Si bien puede existir comunicación, ésta no se da en la cantidad, ni con la fluidez esperada. Como último punto se puede ver que este modelo no asegura resultados rápidos, lo cual es una limitación importante para el desarrollo del sistema bajo este modelo, dadas las exigencias del cliente.

Modelo de Métodos Formales

Simplemente se decidió no usar este paradigma por diversos motivos, entre los que se pueden mencionar:

Baja productividad al no conocer de buena manera el paradigma.

No tener nociones de herramientas útiles para el uso del paradigma.

Dificultad para los desarrolladores de formular técnicas o métodos matemáticos para el desarrollo del sistema.

Ante dichas características se decidió no utilizar este modelo para el desarrollo del sistema.

Proceso Unificado de Desarrollo

Este paradigma presenta una serie de características que permiten una realización adecuada del sistema, dado que las características de este modelo fueron sin dudas las que de mejor manera se comportaron ante todas las circunstancias requeridas por el desarrollo del sistema para poder entregar una buena solución al cliente. Dentro de estas características se puede encontrar herramientas de fácil uso, o que son conocidas por los desarrolladores, asegurando al máximo una buena producción por parte de éstos. Por otra parte, se tiene una buena documentación respecto a la utilización del paradigma, lo que es de gran ayuda para el desarrollo del sistema y para aclarar situaciones de duda. Si bien, se puede notar un paradigma que no tiene una fácil planificación, se pueden observar características buenas que provienen del modelo evolutivo, las que mezcladas con la documentación, la experiencia de los desarrolladores y una estructura de desarrollo bien diseñada permiten realizar el sistema de la mejor manera posible.

Entre las características que mejor se adaptan al desarrollo del sistema, destacan:

Comunicación con el cliente.

Tiempo corto para obtener resultados visibles.

Enfocado al desarrollo.

Permite requerimientos dentro de las distintas etapas de desarrollo.

Iteraciones.

Entrega de documentos de avance del proyecto.

Debido a todas estas características, se decidió realizar el proyecto bajo el paradigma de Proceso Unificado de Desarrollo.

Con la idea de hacer más entendible la elección del paradigma, se creó una tabla (Tabla 1) la cual muestra una serie de características y los distintos paradigmas, marcando aquellas características que posee el modelo. Sólo se pusieron características ventajosas, para poder ver mejor que virtudes ofrecían ciertos paradigmas en comparación a los demás.

	Modelo en Cascada	Desarrollo Evolutivo	Desarrollo en Espiral	Modelo de Métodos Formales	Proceso Unificado de Desarrollo
Corto tiempo para obtener resultados visibles.		✓	✓		✓
Facilidad de planificación.	✓	✓	✓		
Requerimientos en distintas etapas.		✓	✓		✓
Entrega de documentación.	✓			✓	✓
Orientado al desarrollo.		✓	✓		✓
Manejo de riesgos.	✓		✓		✓
Documentación disponible acerca del paradigma.	✓	✓			✓
Adaptable al proyecto.					✓
Experiencia de los desarrolladores.	✓				✓
Buena comunicación con el cliente.		✓			✓
Iteraciones.		✓	✓		✓
Disponibilidad de herramientas.	✓	✓	✓		✓
Cliente relacionado con el desarrollo del sistema.		✓			✓

Tabla 1. Comparación de Metodologías

2.3.-METODOLOGÍAS DE DESARROLLO DE SOFTWARE

2.3.1.-Análisis Clásico

Esta metodología se caracteriza por hacer uso de flujogramas y relatos para poder explicar el desarrollo de un sistema. Esto trae consigo grandes desventajas, dado que, estos flujogramas pueden ser difíciles de entender; siendo este mismo punto el problema con los relatos, dado que pueden obviar algún dato de importancia o no dar a entender lo que realmente se pretende. Por otra parte, los flujogramas pueden ser demasiado lineales, lo que ante cualquier

proyecto de software puede ser inútil dado que los sistemas no suelen ser lineales en su totalidad.

Este enfoque tiene una serie de desventajas, entre las que se pueden encontrar:

Redundancia al repetir la misma información en distintas partes del documento.

Inconsistencia, al estar repetida la información.

Ambigüedad, dado que al ser la especificación de carácter textual, puede ser interpretada de diferentes maneras por usuarios, analistas, diseñadores y programadores.

Difícil mantención, dado que para introducir cambios se debe revisar todo el documento.

Al usarse narraciones, no se posee componentes que puedan ser reutilizables para desarrollos posteriores, por lo que el uso de esta metodología para el desarrollo de un sistema hace que todo el trabajo realizado no sea reutilizable en lo absoluto.

2.3.2.-Análisis Estructurado

En este tipo de análisis se utiliza una notación que muestra el flujo y el contenido de la información. El sistema se divide funcionalmente y dependiendo de los distintos comportamientos que este muestre, se establece lo que se debe construir. Existen dos enfoques de este análisis: Enfoque estructurado clásico y Enfoque estructural esencial.

Enfoque estructurado clásico: Consiste en describir el sistema actual, para luego redefinir aquello que hace, con el fin de conformar una nueva versión para el proceso. Se termina con la decisión acerca de cual será el grado de automatización que se le incorporará al sistema actual a modo de satisfacer las necesidades del cliente.

Enfoque estructural esencial: El modelo esencial del sistema, es un modelo de lo que el sistema debe hacer para satisfacer los requerimientos del usuario, sin importar de que manera esto se implementará. La idea es centrarse en lo que se desea hacer con la idea de que todo es realizable para cumplir las necesidades del cliente.

2.3.3.-Análisis Orientado a Objetos

Este análisis se basa en clases y objetos los que pueden ser cualquier cosa real o abstracta. Los objetos existen para prestar servicios ya que al pasarle ciertos parámetros, los objetos realizan una acción determinada. Esta operación ejecuta el método apropiado, generando una respuesta [WEB16].

En este enfoque se modela en términos de tipos de objetos y de lo que a estos le ocurre o lo que estos realizan. Un tipo de objeto puede tener varios subtipos. Además como una clase implementa el tipo de objeto, una subclase puede heredar propiedades de su clase padre; puede heredar su estructura de datos y los métodos, o algunos de sus métodos.

Existen dos tipos de modelos para este enfoque:

Análisis de la estructura de objetos: Este análisis guía a la representación de los tipos de objetos que le son propios al sistema y las relaciones que mantienen mediante diagramas de relación entre estos.

Análisis de Comportamiento de Objetos: Permite ver que es lo que le ocurre a los objetos con el paso del tiempo. Los estados en los cuales puede estar un objeto y las transiciones de estado en que puede estar, se muestran mediante diagramas de transición de estados, los que permiten ver el comportamiento de un objeto en distintas ocasiones.

2.3.4.-Metodología a utilizar

La metodología a utilizar para el desarrollo de SAPP es la de Orientación a Objetos. Esto recae en el hecho que el paradigma de desarrollo a utilizar es el Proceso Unificado de Desarrollo, el cual posee un alto grado de compatibilidad con la utilización de esta metodología. Por otro lado las características de la metodología de Orientación a Objetos se adaptan de buena manera al proyecto desarrollado en comparación a las otras metodologías.

Los puntos a destacar de la metodología son los que se refieren a la claridad de los diagramas, el entendimiento por parte del cliente y del desarrollador, la reusabilidad, la rapidez del desarrollo, la comunicación con el cliente y el tipo de modelamiento. Todos estos puntos en

conjunto permiten que la metodología seleccionada funcione de muy buena manera sobre el desarrollo de SAPP.

2.4.-HERRAMIENTAS A UTILIZAR

2.4.1.-Herramienta de diseño: UML

UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos [WEB14].

UML se quiere convertir en un lenguaje estándar con el que sea posible modelar todos los componentes del proceso de desarrollo de aplicaciones. Sin embargo, hay que tener en cuenta un aspecto importante del modelo: no pretende definir un modelo estándar de desarrollo, sino únicamente un lenguaje de modelado.

En UML los procesos de desarrollo son diferentes según los distintos dominios de trabajo; no puede ser el mismo el proceso para crear una aplicación en tiempo real, que el proceso de desarrollo de una aplicación orientada a gestión, por poner un ejemplo.

Las diferencias son muy marcadas y afectan a todas las fases del proceso. El lenguaje UML recomienda utilizar los procesos que otras metodologías tienen definidos.

UML es una técnica de modelado de objetos y como tal supone una abstracción de un sistema para llegar a construirlo en términos concretos. El modelado no es más que la construcción de un modelo a partir de una especificación [WEB11].

Un modelo es una abstracción de algo, que se elabora para comprender ese algo antes de construirlo. El modelo omite detalles que no resultan esenciales para la comprensión del original y por lo tanto facilita dicha comprensión.

Artefactos para el Desarrollo de Proyectos

Un artefacto es una información que es utilizada o producida mediante un proceso de desarrollo de software. Pueden ser artefactos un modelo, una descripción o un software. Los

artefactos de UML se especifican en forma de diagramas, éstos junto con la documentación sobre el sistema constituyen los artefactos principales que el modelador puede observar.

Se necesita más de un punto de vista para llegar a representar un sistema. UML utiliza diagramas gráficos para obtener estos distintos puntos de vista de un sistema:

Algunos diagramas de UML son [WEB15]:

Diagrama de Estructura Estática.

Diagrama de Casos de Uso.

Diagrama de Secuencia.

Diagrama de Colaboración.

Diagrama de Estados.

2.4.2.-Lenguaje de programación: Java

La Programación Orientada a Objetos transforma a la programación como una extensión natural del pensamiento. Normalmente las personas entienden el mundo construyendo modelos mentales con porciones del propio mundo, estos modelos son una vista simplificada de como algunas cosas trabajan y con las cuales se puede interactuar, es lo que se conoce como abstracción.

El lenguaje de programación Java, fue diseñado con el propósito de crear un lenguaje que pudiera funcionar en redes computacionales heterogéneas (redes de computadores formadas por más de un tipo de computador, ya sean PC, MAC's, estaciones de trabajo, etc.), y que fuera independiente de la plataforma en la que se vaya a ejecutar. Esto significa que un programa Java puede ejecutarse en cualquier máquina o plataforma.

Las características que un lenguaje de programación debe poseer para ser orientado a objeto son las siguientes:

Encapsulamiento Significa que el objeto es auto-contenido, o sea que la misma definición del objeto incluye tanto los datos que éste usa (atributos) como los procedimientos (métodos) que

actúan sobre los mismos. Cuando se utiliza programación orientada a objetos, se definen clases (que definen objetos genéricos) y la forma en que los objetos interactúan entre ellos, a través de mensajes. Al crear un objeto de una clase dada, se dice que se crea una instancia de la clase, simplemente un objeto. Por ejemplo, una clase podría ser "autos", y un auto dado es una instancia de la clase.

Herencia

Es un mecanismo que define nuevos objetos con base en los existentes. Java la soporta como extensión de clase, definiendo una nueva clase con base en otra ya existente sin modificarla. A la primera clase se le llama subclase o clase extendida, recibe los atributos y métodos de una superclase y agrega otros de su propiedad. Aunque también es posible que la herede dado el surgimiento de jerarquías. Java soporta la herencia sencilla a través de extensión de clase. En este tipo de herencia una clase se extiende a partir de una sola superclase.

Polimorfismo El polimorfismo en Java consiste en declarar y definir varios métodos con el mismo nombre, pero con diferente número y/o tipo de argumentos y que realizan diferentes operaciones.

Ligadura Dinámica Las clases de Java se vinculan dinámicamente al momento de la carga. Por lo tanto, la adición de nuevos métodos y campos de los datos de las clases, no requieren de una recompilación de las clases del cliente. Las aplicaciones pueden ejecutar instrucciones para buscar campos y métodos y luego utilizarlos en cualquier momento. Al llamar a un método, Java busca si está dentro de los métodos asociados al objeto que lo llama, si no lo encuentra, sube a la superclase de ésta y lo busca allí, y así sucesivamente.

Por todo lo nombrado anteriormente Java es un lenguaje de programación completamente orientado a objetos. Ya que éste un programa consta de una o más clases, éstas pueden organizarse en paquetes. Las clases de Java definen objetos de software encerrando los miembros de los datos (campos) y a los miembros de función (métodos). Los miembros pueden designarse como privados, protegidos de paquete o públicos, lo que proporciona una manera conveniente de definir la interfaz pública y el dominio privado de un objeto.

El lenguaje Java se diseñó con las siguientes características:

Simple: Elimina la complejidad de los lenguajes como "C", mantiene las facilidades básicas de lenguaje en un mínimo y proporciona un gran número de extras con las librerías de clases.

Familiar: La sintaxis de Java es muy similar al lenguaje C++, además se han eliminado ciertas características, como los punteros.

Robusto: El sistema de Java maneja la memoria del computador por lo que no es necesario preocuparse por punteros, memoria que no se esté utilizando, etc.

Seguro: El sistema de Java tiene ciertas políticas que evitan se puedan codificar virus con este lenguaje. Existen muchas restricciones, especialmente para los applets, que limitan lo que se puede y no puede hacer con los recursos críticos de un computador.

Portable: Como el código compilado de Java (conocido como byte code) es interpretado, un programa compilado de Java puede ser utilizado por cualquier computador que tenga implementado el intérprete de Java.

Multithreaded (multihilos): El lenguaje soporta la concurrencia a través de hilos, o sea que puede ejecutar varios procedimientos en paralelo.

Interpretado y compilado a la vez: Java es compilado, en la medida en que su código fuente se transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. Por otra parte, es interpretado, ya que los bytecodes se pueden ejecutar directamente sobre cualquier máquina a la cual se haya portado el intérprete.

Dinámico: Java no requiere compilar todas las clases de un programa para que funcione, si se realiza una modificación a una clase, Java se encarga de realizar un Dynamic Bynding o un Dynamic Loading para encontrar las clases.

Ventajas que posee Java:

Es un lenguaje que se diseñó desde cero, por lo cual no tuvo que ser compatible con código existente.

Es un lenguaje orientado a objetos y por lo tanto la mayoría de las variables son objetos.

Ausencia de preprocesador, no existe #defines ni macros, ni compilación condicional dentro del propio lenguaje.

Ausencia de sobrecarga de operadores.

Ausencia de punteros para el programador. Java hace más eficiente las aplicaciones ya que no posee aritmética de punteros que es la causa de muchos errores que no se detectan durante la depuración, sino que se detectan en tiempo de ejecución (en ocasiones detectados por los usuarios finales).

No es necesario liberar la memoria, puesto que esto se realiza mediante un proceso de recolección de basura (garbage collection). El sobre costo de la recolección de basura no es superior al 100%, es decir que si una aplicación se tarda un tiempo x en liberar memoria, con el recolector de basura no se tardará más de $2x$ en tiempo. Cuando se elaboran aplicaciones de más de 1000 o 10000 líneas, o aplicaciones realizadas por un grupo de personas, es muy difícil controlar la liberación de espacios

en memoria, Java ofrece el recolector de basura que se encarga de liberar la memoria en el momento preciso, sin que esto sea una preocupación para los diseñadores.

No es necesario recurrir a ninguna librería de otro fabricante.

Independencia tanto del código fuente como del código compilado de la plataforma en la que se está compilando.

Es un lenguaje de tamaño reducido, todo el paquete de desarrollo ocupa menos de 10 Mb. en disco.

Desventajas:

Los bytecodes tienen una menor velocidad de ejecución, debido a que en cada plataforma tiene que ser interpretado para su ejecución y esto en programas extensos es más evidente.

Las mayúsculas y minúsculas son diferentes, no sólo a nivel de código sino también a nivel de archivos, por lo cual el programador debe tener mucho cuidado al escribir el código de un programa.

La estructura de las clases debe corresponder a la estructura de los directorios, para utilizar una clase se debe conocer exactamente como están estructuradas las carpetas y los archivos en donde se ubican las clases. El uso o manejo del recolector de basura, representa un sobre costo importante para el usuario de la aplicación, por la continua verificación que realiza para determinar cuando liberar memoria de un objeto, aunque los autores del lenguaje argumentan que este sobre costo no es considerable si se toma en cuenta que se disminuye el tiempo de desarrollo y la confiabilidad del software resultante.

2.4.3.-Definición de JSP

Java Server Pages (JSP) es una tecnología que permite mezclar HTML estático con HTML generado dinámicamente [WEB10].

Ventajas de JSP:

La parte dinámica está escrita en Java, lo que hace que sea una herramienta poderosa y fácil de usar.

Es portable a otros sistemas operativos y servidores Web.

Permite realizar todas las acciones que se pueden hacer mediante Servlets, pero con un código más simple y fácil de editar.

Permite también el uso de Servlets en vez de un programa separado para generar las partes dinámicas.

Presenta las características de los JavaScripts, pero con características adicionales, como por ejemplo el envío de formularios y acceso a los recursos en el lado del servidor (Bases de datos, catálogos, etc.).

2.4.4.-Servidor de datos: SQL Server

SQL Server es un sistema de gestión de bases de datos relacionales (SGBD) basado en el lenguaje SQL, capaz de poner a disposición de muchos usuarios grandes cantidades de datos de manera simultánea [WEB9].

Entre sus características figuran:

Soporte de transacciones.
Gran estabilidad.
Gran seguridad.
Escalabilidad.
Soporte a procedimientos almacenados.

Incluye también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente. Permite trabajar en modo cliente-servidor donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo accedan a la información. Además permite administrar información de otros servidores de datos. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle o Sybase.

SQL como Lenguaje

Primero que todo debe quedar claro que SQL no es un lenguaje de programación como C o Pascal, ya que no posee sentencias de selección o iteración (IF ... THEN, FOR, WHILE, etc.). Por lo tanto, cuando se desean utilizar los datos de la BD para realizar algún proceso, se debe programar en otro lenguaje (lenguaje anfitrión), desde el cual se hacen llamadas a sentencias SQL (SQL embebido) cuando se desea interactuar con la BD.

Además, como lenguaje es bastante natural y fácil de leer, y se aplica de forma natural al problema de implementar BD Relacionales, pero como consecuencia de esto es poco estructurado y utiliza muchas palabras superfluas (que, de omitirse, podrían restar claridad a las sentencias).

SQL permite abstraer al usuario de la estructura lógica de la BD mediante las vistas, aunque éstas imponen bastantes restricciones a los procesos de insertar y modificar datos en ellas. Además, habiendo sido estandarizado ya hace mucho tiempo, las BD Relacionales en SQL son muy portables, ya que existen muchos SGBD que lo soportan.

Al ser SQL un lenguaje declarativo, es muy fácil verificar los programas, ya que simplemente se declara lo que se desea como resultado de la ejecución.

SQL y los SGBD

SQL es mucho más que un lenguaje de consulta de BD, ya que permite:

Definir la estructura de los datos;
Recuperar y manipular datos;
Administrar y controlar el acceso a los datos;
Compartir datos de forma concurrente;
Asegurar su integridad.

Lo anterior se hace basándose siempre en el Modelo Relacional de datos. En consecuencia, SQL permite hacer efectivos los objetivos de los SGBD, permitiendo a los usuarios utilizar todo el potencial de los SGBD.

2.4.5.-Microsoft Visio

Microsoft Visio es un programa para la creación de diagramas que puede ayudar a crear diagramas empresariales y técnicos en los que se documenten y organicen ideas, procesos y sistemas complejos. Los diagramas creados en Visio permiten clara, concisa y eficazmente visualizar y comunicar información, de formas no posibles si sólo se utiliza texto y números **[WEB8]**.

Microsoft Visio es un paquete que emplea la creación y diseño de los diagramas, al cien por ciento de su capacidad. Es decir, Microsoft Visio proporciona una amplia creatividad en el diseño de los diagramas, al igual que muchos tipos de diagramas que se pueden formular, desde un simple diagrama de flujo hasta un diagrama de lluvia de ideas con su esquema correspondiente, organigramas y mapas de sitios Web.

Además, Microsoft Visio posee todos los diagramas que utiliza UML para sus representaciones gráficas.

Finalmente, Microsoft Visio es de fácil uso y ofrece mucha flexibilidad con respecto al manejo de los diagramas con los que se está trabajando.

2.5.-ANÁLISIS DE RIESGO

Dentro de lo que es el desarrollo de un sistema, se deben tomar en cuenta ciertos factores que pueden incidir en lo que será el desarrollo y puesta en marcha de éste. Por ende, se debe hacer un análisis a todas las posibles falencias que se podrían presentar mientras el sistema se desarrolla y luego se ejecuta.

Existen tres tipos de riesgos, que son:

Los riesgos del proyecto, estos son los riesgos que amenazan al plan del proyecto. Es decir, si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se retrase y que los costos aumenten.

Los riesgos técnicos, estos son los que amenazan la calidad y la planificación temporal del software que hay que producir. Si un riesgo técnico se convierte en realidad, la implementación puede llegar a ser difícil o imposible.

Los riesgos del negocio, estos amenazan la viabilidad del software a construir. Los riesgos del negocio a menudo ponen en peligro el proyecto o el producto.

Tomando en cuenta los tipos de riesgos y su descripción, se llevará a cabo un análisis de riesgos que puedan influir en el desarrollo y la puesta en marcha del sistema, ya que existen ciertos tipos de riesgos que si no se toman en cuenta pueden llevar a que todo el desarrollo sea en vano.

En la Tabla 2 se muestran los principales riesgos que se identificaron en el proyecto. La tabla además permite conocer el tipo de riesgo, el impacto que éste tendría en el proyecto, la probabilidad de que ocurra y el tiempo de duración que tomaría el riesgo en caso de ocurrir.

La Tabla 3 por su parte, muestra los valores porcentuales asociados a la Tabla 2. Con esto se pretende dar a entender de mejor manera los resultados mostrados por dicha tabla.

Riesgo	Tipo			Impacto			Probabilidad			Marco de tiempo		
	Proyecto	Técnico	Negocio	Marginal	Critico	Catastrófico	Muy Probable	Probable	Improbable	Corto Plazo	Medio Plazo	Largo Plazo
Recursos limitados	X				X			X			X	
Planificación temporal errónea.	X				X			X		X		
Poca experiencia del equipo de trabajo	X				X			X			X	
Subestimación del tamaño del sistema	X				X				X	X		
Mala acotación de los límites del sistema	X				X			X			X	
Errores de codificación		X			X				X		X	
Sistema de difícil mantenimiento		X				X			X			X
Datos incorrectos entregados por el usuario del sistema			X			X		X			X	
El sistema no cumple las expectativas del usuario			X	X					X			X
El usuario no utiliza el sistema y sigue trabajando con los métodos antiguos			X		X				X		X	
Poca comunicación entre el usuario del sistema y los desarrolladores de éste			X			X			X	X		
La institución no pone a disposición el espacio que se necesita en la red			X			X			X		X	

Tabla 2. Identificación del Riesgo

Atributo	Valor	Descripción
Impacto	Catastrófico	Pérdida del sistema. Costo >50%
	Crítico	Recuperación de la capacidad operativa. Costo >10% (<50%)
	Marginal	Costo <10%
Probabilidad	Muy Probable	> 70%
	Probable	De 30% a 70%
	Improbable	< 30%
Marco de tiempo	Corto plazo	30 días
	Medio plazo	1 a 4 meses
	Largo plazo	Más de 4 meses

Tabla 3. Valores Asociados a Riesgos Identificados

2.5.1.-Gestión del Riesgo

Una vez identificados los riesgos a los que se ve enfrentado el proyecto, es necesario establecer a priori, las medidas que permiten evitar la aparición de un riesgo así como también el curso de acción a seguir en caso de que surja uno de los riesgos en el transcurso del proyecto.

La Tabla 4 muestra de qué manera se gestiona el riesgo en SAPP, indicando en riesgo encontrado, su descripción, la estrategia con la que se tratará y la categoría de éste, la cual define si la estrategia es de tipo preventiva y evita la ocurrencia de una situación riesgosa (mitigación) o si permite conocer las acciones a tomar después que un riesgo ocurrió con el fin de que el impacto causado por la aparición de éste no sea tan alto (contingencia).

Riesgo	Estrategia	Categoría
Recursos limitados	Se utilizará HW y SW perteneciente al equipo de desarrollo, como también se trabajará con el que posee la Escuela de Ingeniería Informática de la	Mitigación

	PUCV.	
Planificación temporal errónea	Se realizarán reuniones frecuentes, además se pondrá un gran énfasis en cumplir con las fechas estipuladas en la carta gantt.	Mitigación
Poca experiencia del equipo de trabajo	Investigar los temas sobre los que se tiene poco dominio y complementar conocimiento.	Mitigación
Subestimación del tamaño del sistema	Si el tamaño del sistema es mayor al estimado, es necesario fijar mayores tiempos de trabajo. Además se dará énfasis en los módulos con mayor prioridad, con el fin de que se completen en primera instancia aquellos módulos que dan las principales funcionalidades al sistema y que después se vayan incorporando dichos módulos cuya importancia es más baja.	Contingencia
Mala acotación de los límites del sistema	Teniendo claros los requerimientos y efectuando reuniones periódicas con el cliente y usuario.	Mitigación
Errores de codificación	Probando cada módulo por separado antes de seguir avanzando. Revisando que las reparaciones de errores no produzcan errores en otros módulos (regresión). Fijar tiempos de programación adicionales a los pactados inicialmente con el fin de recuperar el tiempo perdido en la programación del código al cual se le detectó el error, así como el tiempo utilizado para corregirlo	Mitigación/ Contingencia

Interfaces complejas para usuarios del sistema	Mediante pequeños prototipos de interfaz mostrados a los usuarios, para obtener acotaciones respecto a las interfaces y poder diseñarlas apoyadas en instrucciones brindadas por el usuario.	Mitigación
Sistema de difícil mantención	Se desarrollará de forma limpia y clara, además de documentar los hechos importantes.	Mitigación
Datos incorrectos entregados por el usuario del sistema	Corregir los módulos que presentan los errores. Pactar nuevos tiempos para el desarrollo del sistema.	Contingencia.
El sistema no cumple las expectativas del usuario	Reuniones periódicas con el fin de que el usuario conozca el sistema que está siendo construido.	Mitigación
El usuario no utiliza el sistema y sigue trabajando con los métodos antiguos	Se incitará al usuario a utilizar el sistema, enseñándole las virtudes que éste posee.	Contingencia
Poca comunicación entre el usuario del sistema y los desarrolladores de éste	Mediante reuniones periódicas y pactadas con anterioridad.	Mitigación
La institución no ponga a disposición el espacio que se necesita en la red	Se verá la manera de obtener un espacio en la red, ya sea pagado o no.	Contingencia

Tabla 4. Gestión del Riesgo

CAPÍTULO 3: DESARROLLO

3.1.-INTRODUCCIÓN

Hasta este punto se ha evaluado la situación actual del cliente, se han revisado las distintas alternativas para la solución, eligiendo paradigma, metodología y herramientas, se verificó la factibilidad del sistema mediante un estudio de distintas áreas de éste y se analizó los posibles riesgos.

Ahora es necesario mostrar de qué forma se llevará a cabo la solución propuesta al cliente, considerando los requerimientos que éste tenga respecto a las necesidades que él espera satisfacer con el sistema.

La idea de este punto es demostrar la funcionalidad que tendrá el sistema, en relación a cómo será construido considerando el uso de gráficos de UML con la idea de abstraer en mayor nivel las capacidades y funcionalidades del sistema, con la idea que el cliente pueda ver de qué manera se desarrollará el sistema deseado.

3.2.-DEFINICIÓN DE REQUERIMIENTOS

3.2.1.-Requerimientos Funcionales

Permitir el ingreso de datos de alumnos, con el fin de que estos puedan ser utilizados para generar los reportes del sistema. Cabe destacar que este ingreso no debe ser una especie de matrícula sino que un ingreso de datos específicos para el uso de estos por parte de la Oficina Técnica de Estudios Avanzados.

Permitir administrar programas de postítulo, alumnos, profesores y asignaturas.

Permitir generar listas de curso de los distintos programas.

Permitir la creación de listas e informes estadísticos, los que puedan ser creados según la necesidad del usuario, en cuanto éste pueda ingresar parámetros para poder generar las listas y estadísticas que él requiera.

Permitir la realización de búsquedas de alumnos.

Permitir búsqueda y visualización de los datos de los programas de postítulo.

3.2.2.-Requerimientos No Funcionales

El sistema debe dar seguridad a los datos almacenados en el sentido de guardarlos y permitir su respaldo con el fin de asegurar la integridad de estos.

Dar mayor confiabilidad en los informes estadísticos emitidos.

Mejorar la eficiencia en el manejo de datos.

Disminuir el tiempo en el desarrollo de informes estadísticos.

Brindar seguridad en el acceso a los datos por medio de usuarios y contraseñas.

Ser de fácil uso.

Tiempos de respuesta razonables. Evitar lo tiempos de respuesta muy prolongados que hagan suponer un error en el sistema.

3.3.-ANÁLISIS

3.3.1.-Diagrama general de Casos de Uso

En la etapa de análisis se elaboraron una serie de diagramas. La Figura 6 muestra el Diagrama General de Casos de Uso el cual tiene como principal función mostrar la forma en la que interactúa el sistema con los usuarios para conseguir un objetivo o requerimiento [WEB 17].

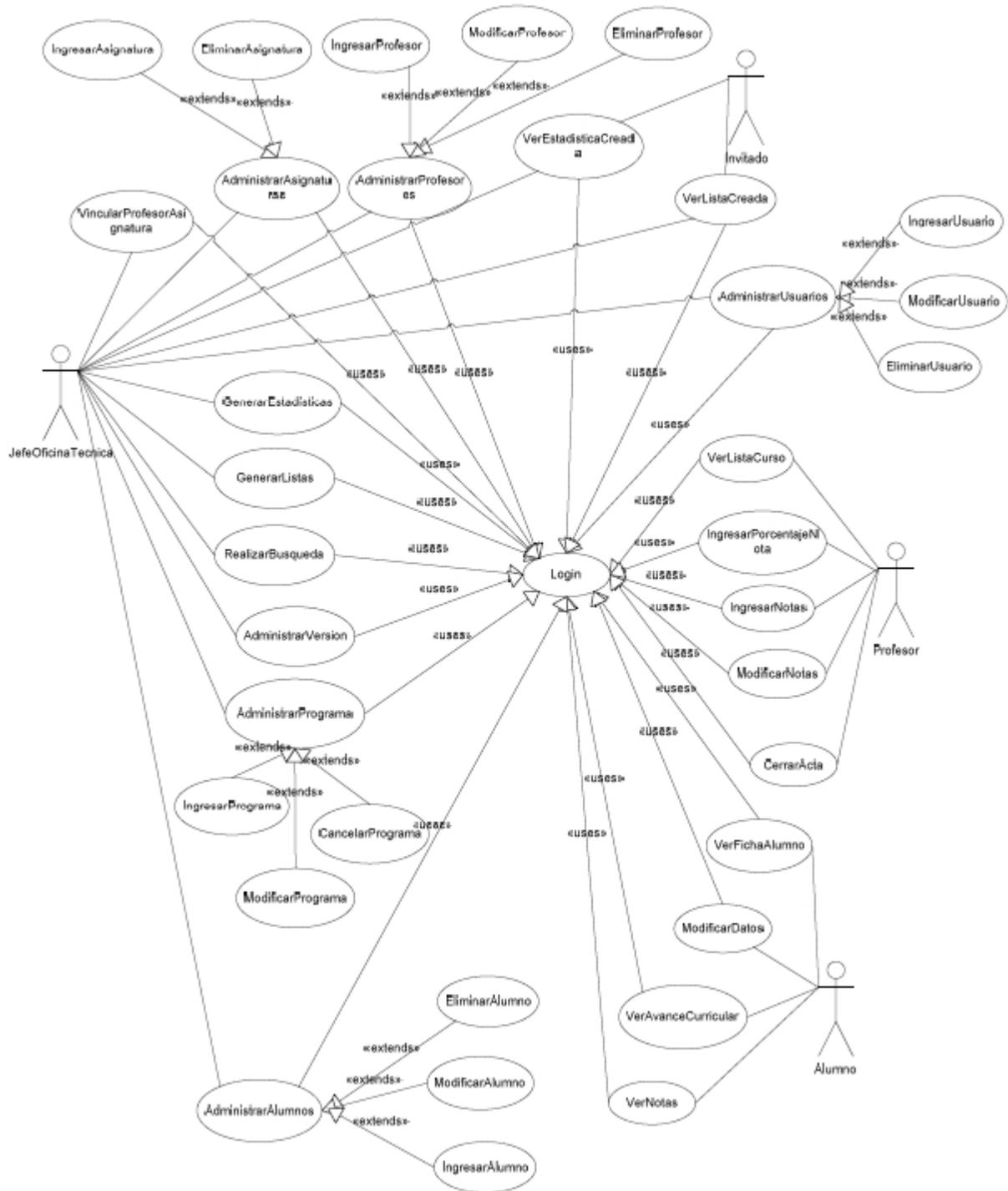


Figura 6. Diagrama General de Casos de Uso

3.3.2.-Casos de Uso

A continuación se detalla la especificación de los casos de uso mostrados en el Diagrama General de Casos de Uso. Con ésta especificación se pretende mostrar de qué manera funciona cada uno de los Casos de Uso, dando a conocer los actores implicados, el flujo que sigue el Caso de Uso, las condiciones previas para el funcionamiento del Caso de Uso, las condiciones posteriores a su ejecución, entre otros.

Caso de Uso	Generar Estadísticas
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Generar una estadística.
Precondiciones	Jefe Oficina Técnica autenticado en el sistema.
Poscondiciones	Se almacena la estadística en el sistema.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Generar Estadística”. 2. Sistema despliega formulario de creación de estadística, mostrando todos los parámetros necesarios para dicha creación. 3. Jefe Oficina Técnica selecciona los parámetros necesarios para la estadística deseada. 4. Sistema despliega la estadística con los datos correspondientes a los parámetros ingresados por el Jefe Oficina Técnica. Se muestra una tabla de datos y el gráfico que representa dicha tabla. Además se muestran los parámetros seleccionados por el Jefe Oficina Técnica para la generación de la estadística y por último la estadística es almacenada automáticamente.

Extensiones	4a.-Búsqueda no retorna resultados. -Sistema despliega mensaje “No se encontraron registros”. -Jefe Oficina Técnica realiza nuevamente una búsqueda. Ante el mismo resultado, termina el caso de uso.
Requerimientos Especiales	Base de Datos poblada.
Frecuencia de Ocurrencia	Ante demanda.

Tabla 5. Especificación Caso de Uso: GenerarEstadísticas

Caso de Uso	GenerarListas
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Generar una lista.
Precondiciones	Jefe Oficina Técnica autenticado en el sistema.
Poscondiciones	Se almacena la lista en el sistema.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Generar Listas”. 2. Sistema despliega formulario de creación de listas, mostrando todos los parámetros necesarios para dicha creación. 3. Jefe Oficina Técnica selecciona los parámetros necesarios para la lista deseada. 4. Sistema despliega la lista con los datos correspondientes a los parámetros ingresados por el Jefe Oficina Técnica, la lista es almacenada automáticamente.

Extensiones	4a.-Búsqueda no retorna resultados. -Sistema despliega mensaje “No se encontraron registros”. -Jefe Oficina Técnica realiza nuevamente una búsqueda. Ante el mismo resultado, termina el caso de uso.
Requerimientos Especiales	Base de Datos poblada.
Frecuencia de Ocurrencia	Ante demanda.

Tabla 6. Especificación Caso de Uso: GenerarListas

Caso de Uso	RealizarBusqueda
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Realizar una búsqueda de distinto tipo..
Precondiciones	Jefe Oficina Técnica autenticado en el sistema.
Poscondiciones	Se despliegan los datos correspondientes a los filtros de búsqueda seleccionados por el Jefe Oficina Técnica.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Búsquedas”. 2. Sistema despliega formulario de búsquedas, mostrando todos los filtros de búsqueda. 3. Jefe Oficina Técnica selecciona los parámetros necesarios para realizar la búsqueda. 4. Sistema despliega los datos correspondientes a los filtros de búsqueda seleccionados por el Jefe Oficina Técnica.

Extensiones	4a.-Búsqueda no retorna resultados. -Sistema despliega mensaje “No se encontraron registros”. -Jefe Oficina Técnica realiza nuevamente una búsqueda. Ante el mismo resultado, termina el caso de uso.
Requerimientos Especiales	Base de Datos poblada.
Frecuencia de Ocurrencia	Ante demanda.

Tabla 7. Especificación Caso de Uso: RealizarBúsqueda

Caso De Uso	AdministrarProgramas
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Administrar los Programas de Postítulos.
Precondiciones	Jefe Oficina Técnica debidamente autenticado.
Poscondiciones	Se ingresa al caso de uso correspondiente dependiendo de la selección hecha por el Jefe de la Oficina Técnica.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Administrar Programas de Postítulo”. 2. El sistema despliega menú con las distintas opciones a realizar: Ingresar, modificar y cancelar Programa. 3. Jefe Oficina Técnica selecciona la opción que desea realizar.
Extensiones	No Existen

Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Trimestral, semestral, anualmente o bajo demanda.

Tabla 8. Especificación Caso de Uso: AdministrarProgramas

Caso de Uso	IngresarPrograma
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Se agregará un nuevo Programa de Postítulo en el sistema.
Precondiciones	Administrar Programas De Postítulo
Poscondiciones	Programa ingresado al sistema
Escenario Principal	1. Jefe Oficina Técnica elige la opción “Ingresar Programa de Postítulo”. 2. Sistema despliega formulario para la creación de nuevos Programas de Postítulo en el sistema. 3. Jefe Oficina Técnica ingresa datos del nuevo Programa y ejecuta la creación. 4. Sistema almacena el nuevo plan de Postítulo.
Extensiones	3a. Jefe Oficina Técnica no ingresa todos los datos o ingresa datos en formato incorrecto. -Sistema despliega mensaje de alerta indicando que faltan datos o que se ingresen de forma correcta. Se vuelve al formulario de ingreso. 3b. Jefe Oficina Técnica sale del escenario de ingreso de nuevos Programas sin ejecutar la creación. - Fin del caso de uso. No se almacenan los datos

	ingresados. Se vuelve al formulario de ingreso.
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Trimestral, semestral o anualmente.

Tabla 9. Especificación Caso de Uso: IngresarPrograma

Caso de Uso	ModificarPrograma
Actor Principal	Jefe Oficina Técnica.
Participantes e Intereses	Jefe Oficina Técnica: Modificar los datos de un Programa de Postítulo en el sistema.
Precondiciones	1. Administrar Programas De Postítulo
Poscondiciones	Los datos modificados deben quedar almacenados en la base de datos y se deben reflejar en todas las instancias en que sea utilizado el Programa modificado.

Escenario Principal	1. Jefe Oficina Técnica elige la opción modificar Programa de Postítulo. 2. Sistema despliega lista de Programas de Postítulo. 3. Jefe Oficina Técnica selecciona Programa a modificar. 4. Sistema despliega datos del Programa a modificar en formulario para realizar los cambios. 5. Jefe Oficina Técnica modifica los valores de los campos editables del formulario y presiona botón guardar. 6. Sistema despliega programa con los datos modificados
Extensiones	5a. Jefe Oficina Técnica no ingresa todos los datos o ingresa datos en formato incorrecto. -Sistema despliega mensaje de alerta indicando que faltan datos o que están en un formato incorrecto. Se vuelve al formulario de modificación de programas. 5b. Jefe Oficina Técnica sale del escenario de modificación sin confirmar la modificación. -Fin del caso de uso. Los datos no se modifican. Se vuelve a la lista de programas de postitulo (selección de programa a modificar).
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario modificar un programa de postítulo.

Tabla 10. Especificación Caso de Uso: ModificarPrograma

Caso de Uso	CancelarPrograma
-------------	------------------

Actor Principal	Jefe Oficina Técnica.
Participantes e Intereses	Jefe Oficina Técnica: Se cancelará un Programa de Postítulo en el sistema.
Precondiciones	1. Administrar Programas De Postítulo
Poscondiciones	Programa cancelado
Escenario Principal	1. Jefe Oficina Técnica elige la opción cancelar programa de Postítulo. 2. Sistema despliega todos los Programas de Postítulos existentes en el sistema. 3. Jefe Oficina Técnica selecciona plan a cancelar. 4. Sistema despliega confirmación. 5. Jefe Oficina Técnica confirma cancelación.
Extensiones	4a. Jefe Oficina Técnica sale del escenario sin confirmar la eliminación. -Fin del caso de uso. No se elimina el programa. Se vuelve a la lista de Programas de Postítulo (selección de programa a eliminar).
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario cancelar un Programa de Postítulo.

Tabla 11. Especificación Caso de Uso: CancelarPrograma

Caso de Uso	AdministrarAlumnos
Actor Principal	Jefe Oficina Técnica

Participantes e Intereses	Jefe Oficina Técnica: Ingresar, modificar o eliminar los datos de un alumno en el sistema.
Precondiciones	1. Jefe de la Oficina Técnica debe estar debidamente autenticado. 2. Alumno debe estar matriculado en algún Programa de Postítulo.
Poscondiciones	Se ingresa al caso de uso correspondiente dependiendo de la selección hecha por el Jefe de la Oficina Técnica.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Administrar Alumnos”. 2. El sistema despliega menú con las distintas opciones a realizar: Ingresar, modificar y eliminar alumno. 3. Jefe Oficina Técnica selecciona la opción que desea realizar.
Extensiones	No Existen
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario administrar datos de alumnos.

Tabla 12. Especificación Caso de Uso: AdministrarAlumnos

Caso de Uso	IngresarAlumno
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Ingresar de manera efectiva los datos de un alumno en el sistema.

Precondiciones	1. Administrar Alumnos.
Poscondiciones	Los datos del alumno quedan ingresados en el sistema. El alumno queda asociado a un Programa de Postítulo.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Ingresar Alumno”. 2. Jefe Oficina Técnica ingresa datos necesarios para el registro de un nuevo alumno en el sistema, además selecciona el programa al cual pertenecerá el alumno. 3. Sistema muestra datos del alumno ingresado.
Extensiones	2a.-Jefe Oficina Técnica no ingresa todos los datos o ingresa datos en formato incorrecto. -Sistema despliega mensaje de alerta indicando que faltan datos o que están en un formato incorrecto. Se vuelve al formulario de ingreso de alumnos. 2b.-Alumno ya fue ingresado con anterioridad al programa de postítulo. -Sistema despliega mensaje indicando que el alumno ya fue ingresado al sistema. 2c.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Al oficializar la matricula de un alumno en un Programa de Postítulo.

Tabla 13. Especificación Caso de Uso: IngresarAlumno

Caso de Uso	ModificarAlumno
-------------	-----------------

Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Modificar de manera efectiva los datos de un alumno en el sistema.
Precondiciones	1. Administrar Alumnos. 2. El alumno debe estar registrado en el sistema
Poscondiciones	Los datos del alumno son modificados. Esto se debe reflejar en todas las instancias en que aparezcan los datos del alumno en el sistema.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Modificar Alumno”. 2. Jefe Oficina Técnica selecciona programa de postítulo al cual pertenece el alumno buscado. 3. Sistema despliega lista de alumnos correspondientes al programa seleccionado 4. Jefe Oficina Técnica selecciona uno de los registros encontrados, para su modificación 5. Jefe Oficina Técnica modifica los valores editables del alumno seleccionado, y presiona el botón guardar. 6. Sistema muestra datos del alumno modificado.
Extensiones	7a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario modificar los datos de un alumno.

Tabla 14. Especificación Caso de Uso: ModificarAlumno

Caso de Uso	EliminarAlumno
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Eliminar de manera efectiva los datos de un alumno en el sistema.
Precondiciones	1. Administrar Alumnos. 2. El alumno debe estar registrado en el sistema.
Poscondiciones	El alumno se “Desactiva” del sistema., y queda almacenado para estadísticas históricas.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Eliminar Alumno”. 2. Jefe Oficina Técnica selecciona programa de postítulo al cual pertenece el alumno buscado. 3. Sistema despliega lista de alumnos correspondientes al programa seleccionado. 4. Jefe Oficina Técnica selecciona uno de los registros encontrados para su eliminación. 5. Jefe Oficina Técnica confirma eliminación de los datos.
Extensiones	5a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario eliminar los datos de un alumno.

Tabla 15. Especificación Caso de Uso: EliminarAlumno

Caso de Uso	AdministrarAsignaturas
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Ingresar, o eliminar los datos de una asignatura.
Precondiciones	Jefe Oficina Técnica debe estar debidamente autenticado en el sistema.
Poscondiciones	No Existen
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Administrar Asignaturas”. 2. El sistema despliega menú con las distintas opciones a realizar: Ingresar y eliminar asignatura. 3. Jefe Oficina Técnica selecciona la opción que desea realizar.
Extensiones	No Existen
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario administrar datos de una asignatura..

Tabla 16. Especificación Caso de Uso: AdministrarAsignaturas

Caso de Uso	IngresarAsignatura
Actor Principal	Jefe Oficina Técnica

Participantes e Intereses	Jefe Oficina Técnica: Ingresar de manera efectiva los datos de una nueva asignatura en el sistema.
Precondiciones	Administrar Asignaturas.
Poscondiciones	Asignatura ingresada correctamente al sistema y vinculada a un programa
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Asignaturas” desplegada en la tabla de los programas ingresados al sistema 2. Sistema levanta una ventana de ingreso de asignaturas para el programa seleccionado. 3. Jefe Oficina Técnica ingresa datos necesarios para el registro de una nueva asignatura en el sistema. 4. Jefe Oficina Técnica confirma el ingreso.
Extensiones	3a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso. Los datos de la nueva asignatura no quedan almacenados en el sistema.
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario ingresar una nueva asignatura en el sistema.

Tabla 17. Especificación Caso de Uso: IngresarAsignatura

Caso de Uso	EliminarAsignatura
Actor Principal	Jefe Oficina Técnica

Participantes e Intereses	Jefe Oficina Técnica: Eliminar de manera efectiva los datos de una asignatura del sistema.
Precondiciones	Administrar Asignaturas. Debe existir al menos una asignatura en el sistema para su eliminación.
Poscondiciones	Asignatura eliminada del sistema.
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Asignaturas” desplegada en la tabla de los programas ingresados al sistema 2. Sistema levanta una ventana listando las asignaturas ingresadas al sistema 3. Jefe Oficina Técnica selecciona asignatura a eliminar 4. Sistema elimina asignatura de los registros.
Extensiones	2a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario eliminar los datos de un usuario del sistema.

Tabla 18. Especificación Caso de Uso: EliminarAsignatura

Caso de Uso	AdministrarProfesores
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Ingresar, modificar o eliminar los datos de un profesor en el sistema.

Precondiciones	1. Jefe Oficina Técnica debe estar debidamente identificado.
Poscondiciones	No Existen
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Administrar Profesores”. 2. El sistema despliega menú con las distintas opciones a realizar: Ingresar, modificar y eliminar profesores. 3. Jefe Oficina Técnica selecciona la opción que desea realizar.
Extensiones	No Existen
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario administrar datos de un profesor del sistema.

Tabla 19. Especificación Caso de Uso: Administrar Profesores

Caso de Uso	Ingresar Profesor
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Ingresar de manera efectiva los datos de un profesor en el sistema.
Precondiciones	1. Administrar Profesores.
Poscondiciones	No Existen

Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Ingresar Profesor”. 2. Jefe Oficina Técnica ingresa datos necesarios para el registro de un nuevo profesor en el sistema. 3. Sistema despliega datos del profesor ingresado
Extensiones	2a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario ingresar los datos de un nuevo profesor del sistema.

Tabla 20. Especificación Caso de Uso: IngresarProfesor

Caso de Uso	ModificarProfesor
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Modificar de manera efectiva los datos de un profesor del sistema.
Precondiciones	1. Administrar Profesores. 2. El usuario debe estar registrado en el sistema
Poscondiciones	No Existen

Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Modificar Profesor”. 2. Sistema despliega profesores existentes 3. Jefe Oficina Técnica selecciona uno de los registros listados, para su modificación 4. Jefe Oficina Técnica modifica los valores editables del profesor seleccionado y presiona el botón guardar. 5. Sistema despliega profesor con los datos debidamente modificados
Extensiones	4a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario modificar los datos de un profesor.

Tabla 21. Especificación Caso de Uso: ModificarProfesor

Caso de Uso	EliminarProfesor
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Eliminar de manera efectiva los datos de un profesor del sistema.
Precondiciones	1. Administrar Profesor. 2. El profesor debe estar registrado en el sistema.
Poscondiciones	Profesor eliminado del sistema, sólo queda en el historial para aproximadas listas o estadísticas.

Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Eliminar Profesor”. 2. Sistema despliega los profesores existentes 3. Jefe Oficina Técnica selecciona uno de los registros encontrados para su eliminación. 4. Sistema despliega mensaje de confirmación. 5. Jefe Oficina Técnica confirma eliminación.
Extensiones	5a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario eliminar los datos de un profesor del sistema.

Tabla 22. Especificación Caso de Uso: Eliminar Profesor

Caso De Uso	VerEstadisticaCreada
Actor Principal	Jefe Oficina Técnica. Invitado
Participantes e Intereses	Jefe Oficina Técnica, Invitado: Revisar una estadística previamente almacenada en el sistema
Precondiciones	Debe existir al menos una estadística creada en el sistema.
Poscondiciones	Vista de estadística creada en el sistema

Escenario Principal	1. Actor selecciona la opción “Ver Estadísticas”. 2. Sistema muestra una serie de opciones por las cuales se buscarán las estadísticas creadas 3. Actor selecciona opciones de búsqueda 4. Sistema despliega un listado de las estadísticas almacenadas en el sistema con respecto a los parámetros de búsqueda. 5. Actor selecciona la estadística deseada. 6. Sistema despliega la estadística seleccionada mostrando la información solicitada y su gráfico correspondiente.
Extensiones	2a. No Existen estadísticas almacenadas. -Sistema despliega un mensaje de alerta indicando que no se encuentran estadísticas disponibles.
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 23. Especificación Caso de Uso: VerEstadísticaCreada

Caso De Uso	VerListaCreada
Actor Principal	Jefe Oficina Técnica. Invitado
Participantes e Intereses	Jefe Oficina Técnica, Invitado: Revisar una lista previamente almacenada en el sistema
Precondiciones	Debe existir al menos una lista creada en el sistema.
Poscondiciones	Vista de lista creada en el sistema.

Escenario Principal	1. Actor selecciona la opción “Ver Listas”. 2. Sistema muestra una serie de opciones por las cuales se busca las listas creadas 3. Actor selecciona opciones de búsqueda 4. Sistema despliega un registro de todas las listas almacenadas en el sistema con respecto a los parámetros de búsqueda. 5. Actor selecciona la lista deseada. 6. Sistema despliega la lista seleccionada.
Extensiones	2a. No Existen listas almacenadas. -Sistema despliega un mensaje de alerta indicando que no se encuentran listas disponibles.
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 24. Especificación Caso de Uso: VerListaCreada

Caso De Uso	IngresarNotas
Actor Principal	Profesor
Participantes e Intereses	Profesor: Ingresar las notas parciales a los alumnos que cursan una asignatura dictada.
Precondiciones	1. Profesor ingresado al sistema con anterioridad. 2. Profesor autenticado en el sistema. 3. Programa y asignatura seleccionada por el profesor 4. Las notas de la asignatura deben tener previamente un porcentaje, el cual sirve para el cálculo del promedio final.

Poscondiciones	Las notas de los alumnos quedan almacenadas en el sistema, para un posterior calculo de promedio que permitirá ver si el alumno esta aprobado o reprobado en la asignatura seleccionada.
Escenario Principal	1. Profesor selecciona la opción “Ingresar Notas”. 2. Sistema despliega lista de alumnos pertenecientes al curso seleccionado. 3. Profesor ingresa las notas parciales a cada uno de los alumnos del curso. 4. Profesor confirma el ingreso de las notas presionando el botón guardar.
Extensiones	4a. Profesor sale del escenario de ingreso de notas, sin guardar los cambios. -Las notas no quedan almacenadas.
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 25. Especificación Caso de Uso: IngresarNotas

Caso De Uso	ModificarNotas
Actor Principal	Profesor
Participantes e Intereses	Profesor: Modificar las notas parciales a los alumnos que cursan una asignatura dictada.

Precondiciones	1. Profesor ingresado al sistema con anterioridad. 2. Profesor autenticado en el sistema. 3. Programa y asignatura seleccionada por el profesor 4. Debe haber ingresada al menos una nota en la asignatura.
Poscondiciones	Las notas de los alumnos quedan modificadas.
Escenario Principal	1. Profesor selecciona la opción “Modificar Notas” 2. Sistema despliega lista de alumnos pertenecientes al curso seleccionado mostrando un link al lado derecho de cada registro, donde al presionarlo las notas que fueron ingresadas se convierten en un campo editable. 3. Profesor modifica las notas parciales a los alumnos que sea necesario. 4. Profesor confirma la modificación de las notas presionando el botón guardar.
Extensiones	3a. Profesor sale del escenario de modificación de notas, sin guardar los cambios. -Las notas no quedan almacenadas.
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 26. Especificación Caso de Uso: ModificarNotas

Caso de Uso	VerFichaAlumno
Actor Principal	Alumno
Participantes e Intereses	Alumno: Ver ficha de datos personales.

Precondiciones	1.-Alumno ingresado con anterioridad en el sistema 2.- Alumno autenticado en el sistema
Poscondiciones	Se muestra ficha de alumno
Escenario Principal	1. Alumno ingresa al sistema mediante su nombre de usuario y contraseña. 2. Sistema despliega la ficha del alumno.
Extensiones	No Existen.
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 27. Especificación Caso de Uso: VerFichaAlumno

Caso de Uso	VerAvanceCurricular
Actor Principal	Alumno
Participantes e Intereses	Alumno: Modificar datos personales.
Precondiciones	1. Alumno ingresado con anterioridad en el sistema 2. Alumno autenticado en el sistema
Poscondiciones	Los datos del alumno quedan modificados.

Escenario Principal	1. Sistema despliega la ficha del alumno. 2. Alumno presiona la opción “Avance Curricular”. 3. Sistema despliega el avance curricular del alumno, mostrando las asignaturas que el alumno ha cursado y las que no ha cursado.
Extensiones	No Existen.
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 28. Especificación Caso de Uso: VerAvanceCurricular

Caso de Uso	VerNotas
Actor Principal	Alumno
Participantes e Intereses	Alumno: ver notas
Precondiciones	1.-Alumno ingresado con anterioridad en el sistema 2.- Alumno autenticado en el sistema
Poscondiciones	Los datos del alumno quedan modificados.
Escenario Principal	1. Sistema despliega la ficha del alumno. 2. Alumno presiona la opción “Ver Notas”. 3. Sistema despliega asignaturas cursadas por el alumno, y aquellas que estén siendo cursadas. 4. Alumno selecciona asignatura. 5. Sistema despliega las notas parciales del alumno y el promedio de éste.

Extensiones	No Existen.
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 29. Especificación Caso de Uso: Ver Notas

Caso De Uso	VerListaCurso.
Actor Principal	Profesor
Participantes e Intereses	Profesor: Conocer la lista de alumnos de un curso.
Precondiciones	1. Profesor ingresado al sistema con anterioridad. 2. Profesor autenticado en el sistema. 3. Programa y asignatura seleccionada por el profesor.
Poscondiciones	Listado de alumnos perteneciente al curso seleccionado
Escenario Principal	1. Profesor selecciona la opción “Ver Lista de un Curso”. 2. Sistema despliega todos los alumnos que pertenecen al programa y a la asignatura seleccionada con anterioridad
Extensiones	No Existen.
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 30. Especificación Caso de Uso: VerListaCurso

Caso de Uso	VincularProfesorAsignatura
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Vincular profesor a asignatura.
Precondiciones	1. Jefe Oficina Técnica autenticado en el sistema 2. Administrar profesor 3. profesor ingresado en el sistema con anterioridad
Poscondiciones	Profesor vinculado a asignatura
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción asignaturas en la tabla que se listan los profesores existentes en el sistema 2. Sistema levanta una ventana donde se muestran los programa, versión y asignatura existentes. 3. Jefe Oficina Técnica selecciona programa, versión y asignatura a vincular y presiona el botón vincular. 4. Sistema despliega en una tabla los profesores vinculados a las versiones de las asignaturas seleccionadas.
Extensiones	3a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso 3b.-el profesor fue vinculado a otra asignatura del mismo periodo, sistema despliega mensaje de advertencia indicando que el profesor ya está impartiendo una asignatura en el periodo seleccionado.

Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario eliminar los datos de un profesor del sistema.

Tabla 31. Especificación Caso de Uso: Vincular Profesor Asignatura

Caso de Uso	Administrar Versión
Actor Principal	Jefe Oficina Técnica.
Participantes e Intereses	Jefe Oficina Técnica: Asignar una fecha en la que se dictará un Programa y el profesor a cargo de dictarlo.
Precondiciones	Programa previamente ingresado en el sistema.
Poscondiciones	Versiones creadas o cerradas

Escenario Principal	1. Jefe Oficina Técnica elige la opción “Administrar Versiones”. 2. Sistema despliega en una nueva ventana programas existentes en el sistema 3. Jefe Oficina Técnica selecciona programa 4. Sistema despliega todas las versiones que se dictan del programa seleccionado, además muestra la opción crear nueva versión. 5. Jefe Oficina Técnica selecciona versión que desea administrar o pincha la opción crear nueva versión. 6. Sistema despliega dependiendo si es una nueva versión o no, el estado de esta última (periodo en curso, cerrados y por impartirse) 7. Jefe Oficina Técnica selecciona cerrar o impartir periodo según sea el caso. 8. Sistema ejecuta opción seleccionada por el Jefe Oficina Técnica.
Extensiones	7a. Jefe Oficina Técnica sale del escenario de ingreso de administración de versiones sin realizar ninguna acción -Fin del caso de uso
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Trimestral, semestral o anualmente.

Tabla 32. Especificación Caso de Uso: AdministrarVersión

Caso De Uso	CerrarActas
Actor Principal	Profesor
Participantes e Intereses	Profesor: Cerrar el acta de una asignatura.

Precondiciones	1.-Profesor ingresado al sistema con anterioridad. 2.- Profesor autenticado en el sistema. 3.-Programa y asignatura seleccionada por el profesor 4.-Debe haber ingresada al menos una asignatura para su modificación.
Poscondiciones	1. Se cierra el acta de la asignatura. 2. Es posible imprimir el acta de la asignatura cerrada para la firma del profesor a cargo de dictarla.
Escenario Principal	1. Profesor selecciona la opción “Cerrar Acta” 2. Sistema muestra los alumnos del curso y sus notas parciales. 3. Sistema solicita confirmación de cierre de acta. 4. Profesor confirma el cierre del acta presionando el botón cerrar acta.
Extensiones	6a. Profesor sale del escenario de cierre de actas, sin confirmar la operación. -No se cierra el acta, se pueden seguir ingresando y modificando notas en el curso seleccionado..
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Fin de un periodo académico.

Tabla 33. Especificación Caso de Uso: CerrarActas

Caso de Uso	ModificarDatos
Actor Principal	Alumno

Participantes e Intereses	Alumno: Modificar datos personales.
Precondiciones	1. Alumno ingresado con anterioridad en el sistema 2. Alumno autenticado en el sistema
Poscondiciones	Los datos del alumno quedan modificados.
Escenario Principal	1. Sistema despliega la ficha del alumno. 2. Alumno presiona la opción “Modificar Datos”. 3. Sistema despliega en forma editable los datos que pueden ser modificados por el alumno. 4. Alumno ingresa nuevos valores en los campos editables. 5. Alumno confirma la modificación presionando el botón guardar.
Extensiones	5a. Alumno sale del escenario de modificación sin confirmar la operación. -Fin de Caso de Uso. No se almacenan los datos modificados.
Requerimientos Especiales	Sólo pueden estar en forma editable por el alumno, aquellos campos que no tengan información relevante. Esto se entiende como dirección, correo electrónico, teléfono, entre otros.
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 34. Especificación Caso de Uso: ModificarDatos

Caso de Uso	Login
-------------	-------

Actor Principal	Invitado Jefe Oficina Técnica Profesor Alumno
Participantes e Intereses	Invitado, Jefe Oficina Técnica, Profesor, Alumno: Hacer uso del sistema.
Precondiciones	1. Usuario debe tener creada una cuenta para hacer uso del sistema
Poscondiciones	Usuario tiene acceso sólo a las funciones del sistema que son permitidas por su perfil.
Escenario Principal	1. Usuario ingresa su nombre de usuario y contraseña. 2. Sistema valida que los datos ingresados sean correctos.
Extensiones	2.a.-Nombre de usuario o contraseña incorrectos - Sistema despliega mensaje “usuario no encontrado”. - usuario debe reingresar nombre de usuario y contraseña.
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Diariamente

Tabla 35. Especificación Caso de Uso: Login

Caso de Uso	AdministrarUsuarios
Actor Principal	Jefe Oficina Técnica

Participantes e Intereses	Jefe Oficina Técnica: Ingresar, modificar o eliminar los datos de un usuario en el sistema.
Precondiciones	1. Jefe Oficina Técnica debe estar debidamente autenticado en el sistema.
Poscondiciones	No Existen
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Administrar Usuarios”. 2. El sistema despliega menú con las distintas opciones a realizar: Ingresar, modificar y eliminar usuario. 3. Jefe Oficina Técnica selecciona la opción que desea realizar.
Extensiones	No Existen
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario administrar datos de un usuario del sistema.

Tabla 36. Especificación Caso de Uso: AdministrarUsuarios

Caso de Uso	IngresarUsuario
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Ingresar de manera efectiva los datos de un usuario en el sistema.
Precondiciones	1. Administrar Usuarios.

Poscondiciones	Usuario ingresado al sistema
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Ingresar Usuario”. 2. Jefe Oficina Técnica ingresa datos necesarios para el registro de un nuevo usuario en el sistema. 3. Sistema muestra datos del usuario ingresado.
Extensiones	2a.-Jefe Oficina Técnica no ingresa todos los datos o ingresa datos en formato incorrecto. -Sistema despliega mensaje de alerta indicando que faltan datos o que están en un formato incorrecto. Se vuelve al formulario de ingreso de usuarios. 2b.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario ingresar los datos de un nuevo usuario del sistema.

Tabla 37. Especificación Caso de Uso: IngresarUsuarios

Caso de Uso	ModificarUsuario
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Modificar de manera efectiva los datos de un usuario del sistema.
Precondiciones	1. Administrar Usuarios. 2. El usuario debe estar registrado en el sistema

Poscondiciones	No Existen
Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Modificar Usuario”. 2. Sistema despliega los usuarios existentes en el sistema 3. Jefe Oficina Técnica selecciona uno de los registros listados, para su modificación 4. Jefe Oficina Técnica modifica los valores editables del usuario seleccionado. 5. Sistema despliega usuario con los datos debidamente modificados
Extensiones	7a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario modificar los datos de un usuario.

Tabla 38. Especificación Caso de Uso: Modificar Usuarios

Caso de Uso	EliminarUsuario
Actor Principal	Jefe Oficina Técnica
Participantes e Intereses	Jefe Oficina Técnica: Eliminar de manera efectiva los datos de un usuario del sistema.
Precondiciones	1. Administrar Usuarios. 2. El usuario debe estar registrado en el sistema.
Poscondiciones	No Existen

Escenario Principal	1. Jefe Oficina Técnica selecciona la opción “Eliminar Usuario”. 2. Sistema despliega los usuarios existentes 3. Jefe Oficina Técnica selecciona uno de los registros encontrados para su eliminación. 4. Sistema despliega mensaje de confirmación. 5. Jefe Oficina Técnica confirma eliminación.
Extensiones	5a.-Jefe Oficina Técnica cancela la operación -Fin Caso de uso
Requerimientos Especiales	No Existen.
Frecuencia de Ocurrencia	Cuando sea necesario eliminar los datos de un usuario del sistema.

Tabla 39. Especificación Caso de Uso: EliminarUsuario

Caso De Uso	IngresarPorcentajeNotas
Actor Principal	Profesor
Participantes e Intereses	Profesor: Ingresar los porcentajes que tendrá cada una de las notas parciales en una asignatura, para el cálculo del promedio final de los alumnos.
Precondiciones	1. Profesor ingresado al sistema con anterioridad. 2. Profesor autenticado en el sistema. 3. Programa y asignatura seleccionada por el profesor.
Poscondiciones	Los promedios de los alumnos estarán dados en base al porcentaje que se le asigne a cada una de las notas

	parciales.
Escenario Principal	1. Profesor selecciona la opción “Ingresar Porcentajes Asignatura”. 2. Profesor ingresa el número de nota y el porcentaje que tendrá ésta dentro del promedio. 3. Profesor confirma el ingreso presionando el botón guardar.
Extensiones	3a. Profesor sale del escenario de ingreso de porcentajes de notas, sin guardar los cambios. -Los porcentajes no quedan almacenadas. 3b. La suma de los porcentajes de las notas distinto al 100%. -Se despliega un mensaje de error indicando que la suma de porcentajes debe ser igual al 100%.
Requerimientos Especiales	No Existen
Frecuencia de Ocurrencia	Ante Demanda.

Tabla 40. Especificación Caso de Uso: IngresarPorcentajeNotas

3.3.3.-Tabla de Priorización de Casos de Uso

Si bien todos los Casos de Uso tienen un grado de importancia dentro del sistema, son sólo algunos los que representan las principales funcionalidades de éste. Las principales características del sistema, como por ejemplo la creación de estadísticas o listas son representadas por ciertos Casos de Uso.

Ante esta situación se creó una tabla que permite ver con mayor claridad cuáles son los Casos de Uso que mejor definen las principales funciones que tiene el sistema, los que permiten alimentar el sistema para su funcionamiento y aquellos cuyas funciones no son imprescindibles para el sistema.

La tabla 41 muestra dicha priorización, nombrando cada uno de los Casos de Uso, la prioridad que tienen en el sistema y el motivo por el cual fueron considerados dentro de cada categoría de priorización.

Caso de Uso	Prioridad	Motivo
	Alta	Estos casos de uso definen la funcionalidad básica del sistema, dado que las funciones comprendidas en estos solucionan la mayor parte de los problemas.
GenerarEstadísticas		
GenerarListas		
RealizarBusqueda		
AdministrarProgramas		
IngresarPrograma		
ModificarPrograma		
CancelarPrograma		
AdministrarAlumnos		
IngresarAlumno		
ModificarAlumno		
EliminarAlumno	Media	Estos casos de uso, si bien no son los que dan la mayor funcionalidad al sistema,
AdministrarAsignaturas		

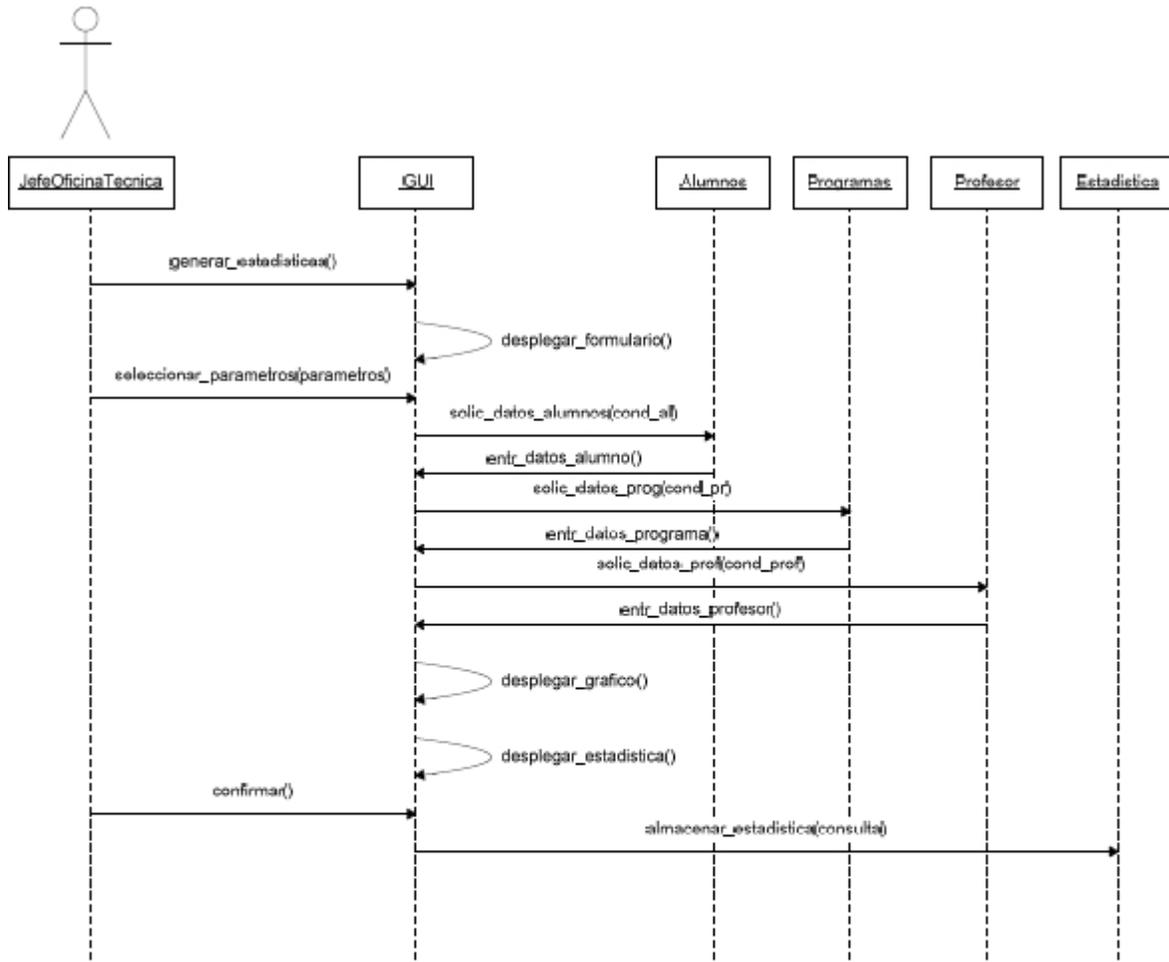
IngresarAsignatura		<p>tienen un grado de importancia medio, dado que son los encargados de alimentar de datos al sistema o de prestar función con resultados visibles, pero que no son considerados de mayor importancia dentro de las características del sistema.</p>
EliminarAsignatura		
AdministrarProfesores		
IngresarProfesor		
ModificarProfesor		
EliminarProfesor		
VerEstadisticaCreada		
VerListaCreada		
IngresarNotas		
ModificarNotas		
VerFichaAlumno		
VerAvanceCurricular		
VerNotas		
VerListaCurso		
VincularProfesorAsignatura		
	Baja	Se consideró de prioridad baja estos casos de uso debido a que las funciones prestadas
AdministrarVersión		

CerrarActas		por estos no son imprescindibles para el funcionamiento del sistema.
ModificarDatos		
Login		
AdministrarUsuarios		
IngresarUsuarios		
ModificarUsuarios		
EliminarUsuario		
IngresarPorcentajeNotas		

Tabla 41. Priorización de Casos de Uso

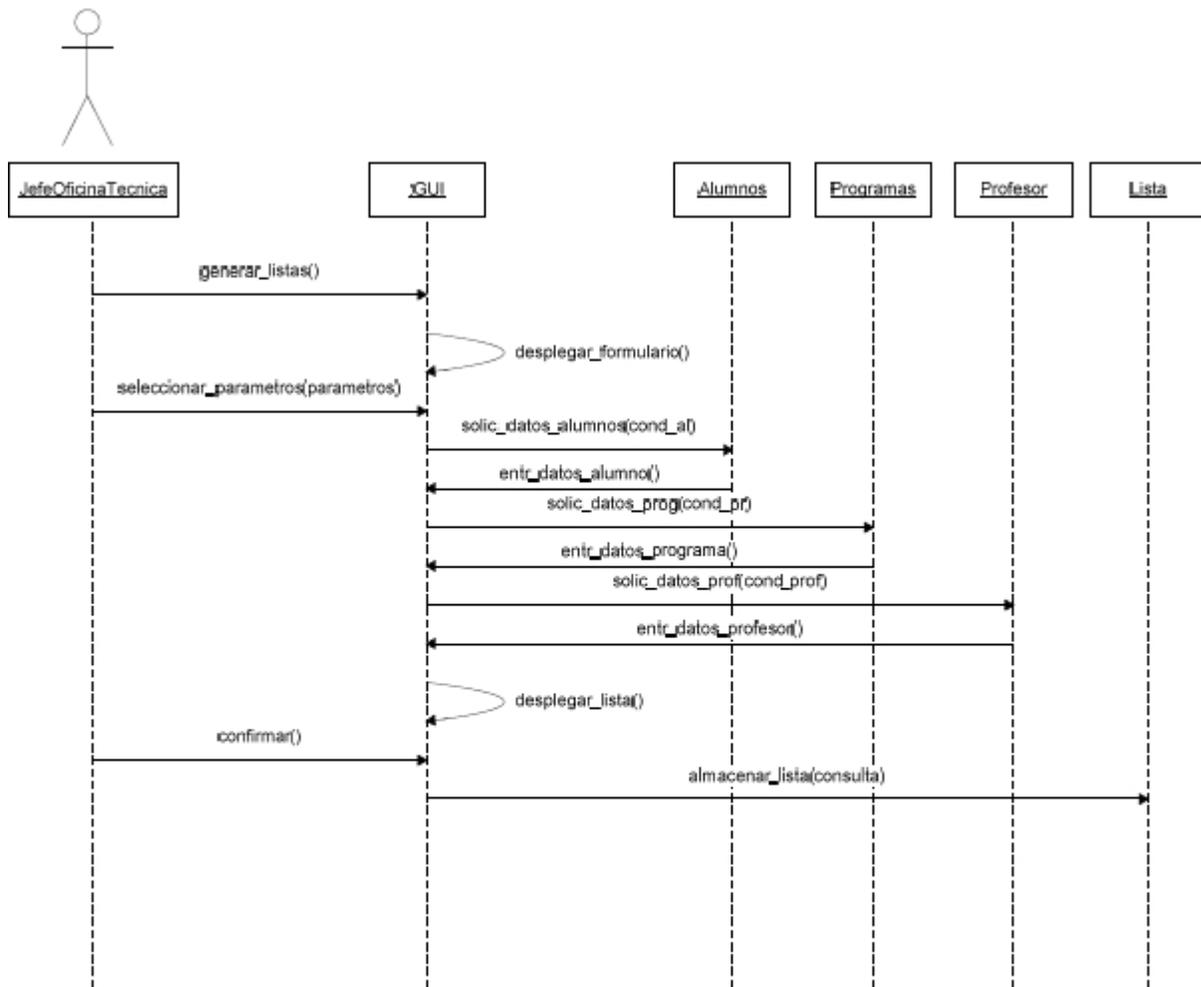
3.3.4.-Diagramas de Secuencia

Los diagramas de las Figuras 7 a 17 muestran los Diagramas de Secuencia para aquellos Casos de Uso que fueron considerados de prioridad alta. Estos diagramas tienen como finalidad mostrar los eventos generados por actores externos, el orden en que estos eventos se producen y los eventos entre sistemas para el escenario específico de un caso de uso [LARMAN].



- 1 Jefe Oficina Técnica selecciona la opción "Generar Estadística"
 - 2 Sistema despliega formulario de creación de estadística, mostrando todos los parámetros necesarios para dicha creación.
 - 3 Jefe Oficina Técnica selecciona los parámetros necesarios para la estadística deseada
 - 4 Sistema despliega la estadística con los datos correspondientes a los parámetros ingresados por el Jefe Oficina Técnica. Se muestra una tabla de datos y el gráfico que representa dicha tabla.
- Además se muestran los parámetros seleccionados por el Jefe Oficina Técnica para la generación de la estadística y por último la estadística es almacenada automáticamente.

Figura 7. Diagrama de Secuencia – Generar Estadísticas



1. Jefe Oficina Técnica selecciona la opción 'Generar Listas'.
2. Sistema despliega formulario de creación de listas, mostrando todos los parámetros necesarios para dicha creación
3. Jefe Oficina Técnica selecciona los parámetros necesarios para la lista deseada
4. Sistema despliega la lista con los datos correspondientes a los parámetros ingresados por el Jefe Oficina Técnica, la lista es almacenada automáticamente

Figura 8. Diagrama de Secuencia – Generar Listas

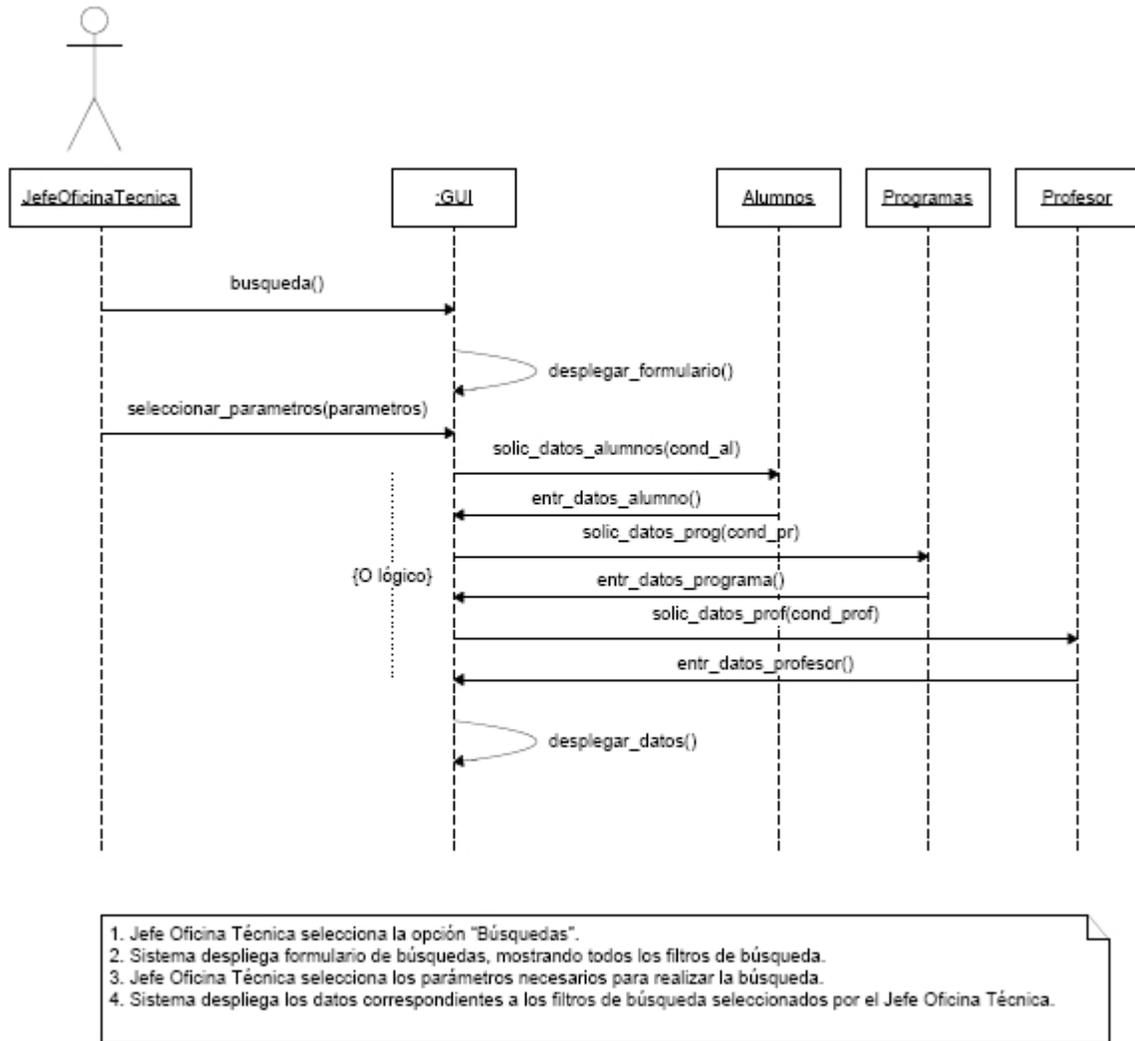


Figura 9. Diagrama de Secuencia – Realizar Búsqueda



Figura 10. Diagrama de Secuencia – Administrar Programas

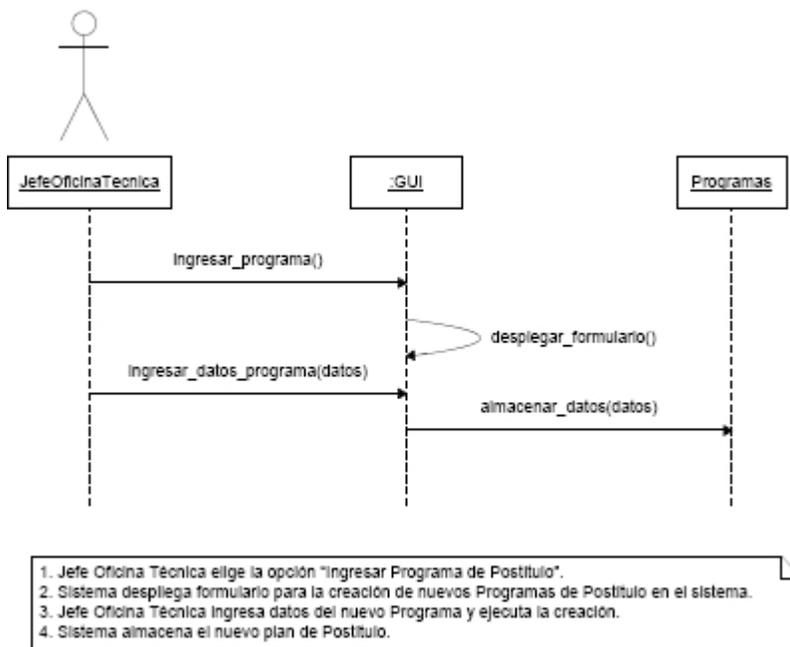
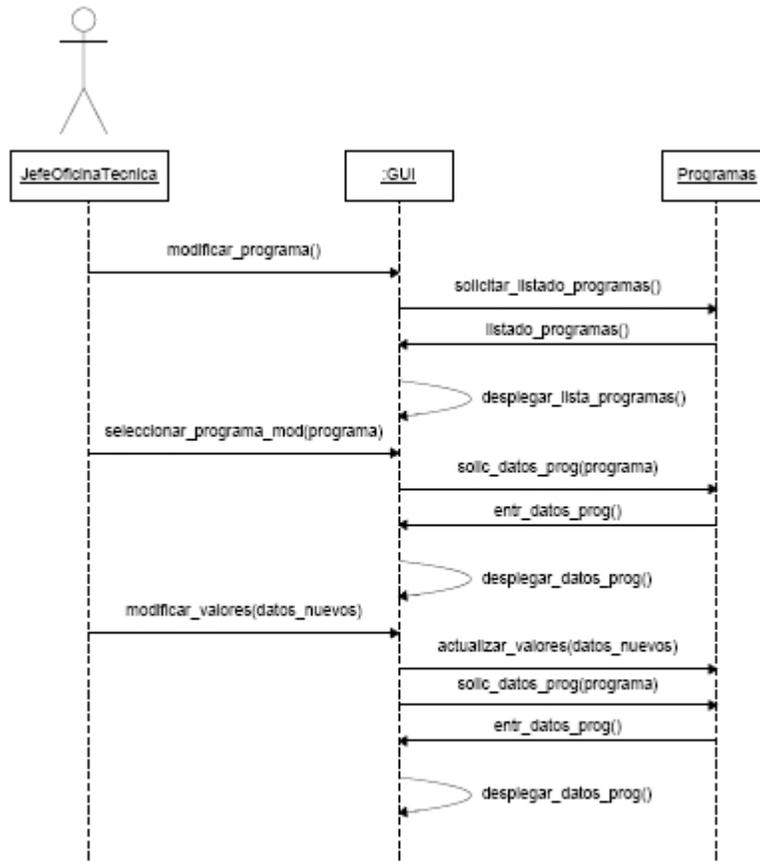


Figura 11. Diagrama de Secuencia – Ingresar Programa



1. Jefe Oficina Técnica elige la opción modificar Programa de Posttulo.
2. Sistema despliega lista de Programas de Posttulo.
3. Jefe Oficina Técnica selecciona Programa a modificar.
4. Sistema despliega datos del Programa a modificar en formulario para realizar los cambios.
5. Jefe Oficina Técnica modifica los valores de los campos editables del formulario y presiona boton guardar.
6. Sistema despliega programa con los datos modificados

Figura 12. Diagrama de Secuencia – Modificar Programa

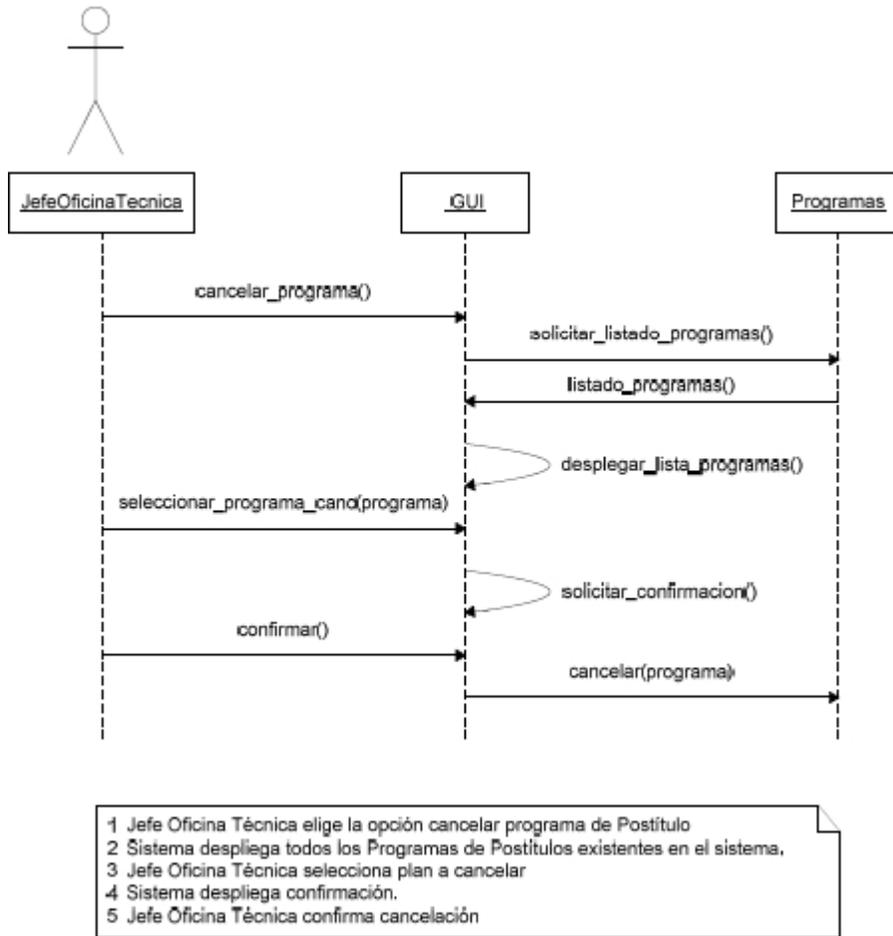
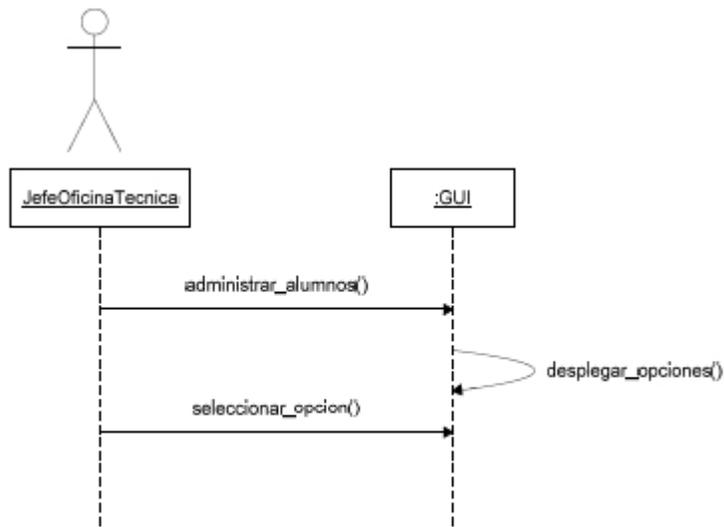
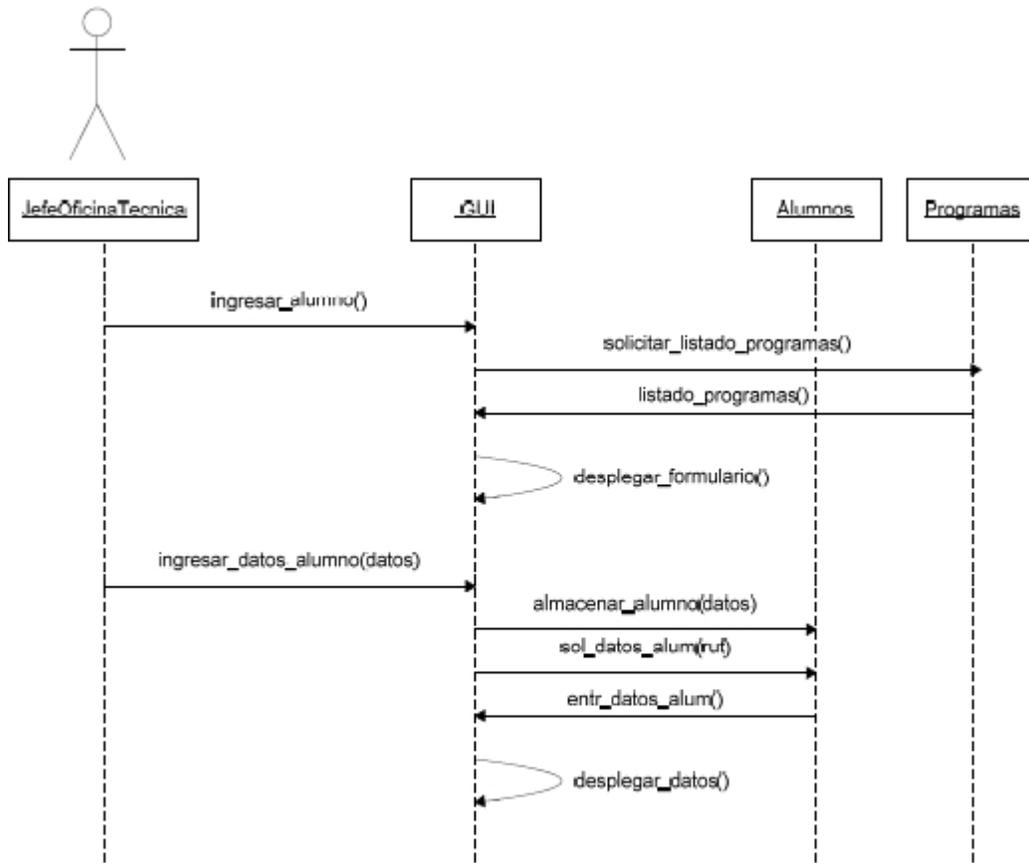


Figura 13. Diagrama de Secuencia – Cancelar Programa



- 1 Jefe Oficina Técnica selecciona la opción "Administrar Alumnos"
- 2 El sistema despliega menú con las distintas opciones a realizar Ingresar modificar y eliminar alumno.
- 3 Jefe Oficina Técnica selecciona la opción que desea realizar.

Figura 14. Diagrama de Secuencia – Administrar Alumnos



1. Jefe Oficina Técnica selecciona la opción "Ingresar Alumno"
2. Jefe Oficina Técnica ingresa datos necesarios para el registro de un nuevo alumno en el sistema, además selecciona el programa al cual pertenecerá el alumno
3. Sistema muestra datos del alumno ingresado

Figura 15. Diagrama de Secuencia – Ingresar Alumno

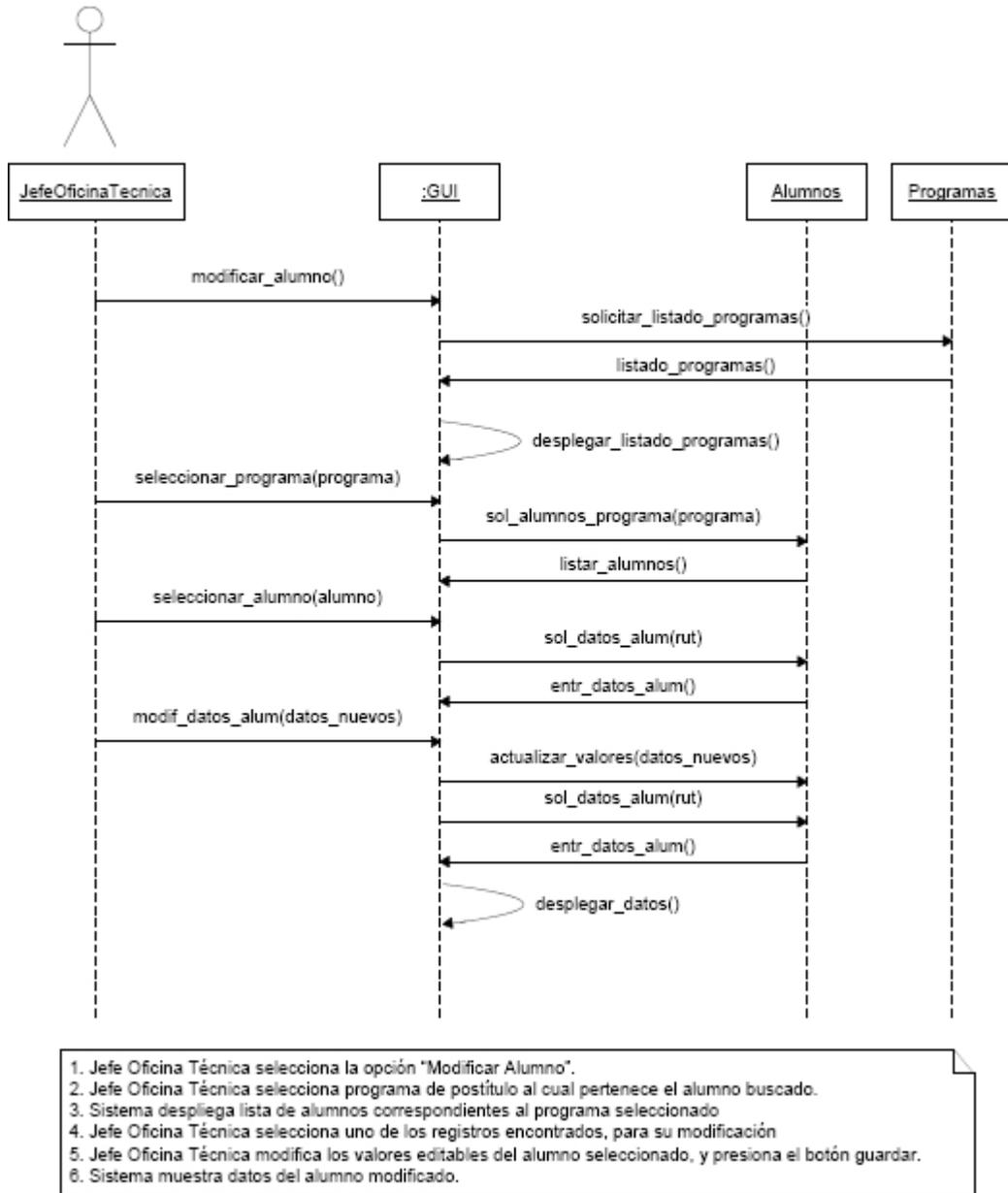


Figura 16. Diagrama de Secuencia – Modificar Alumno

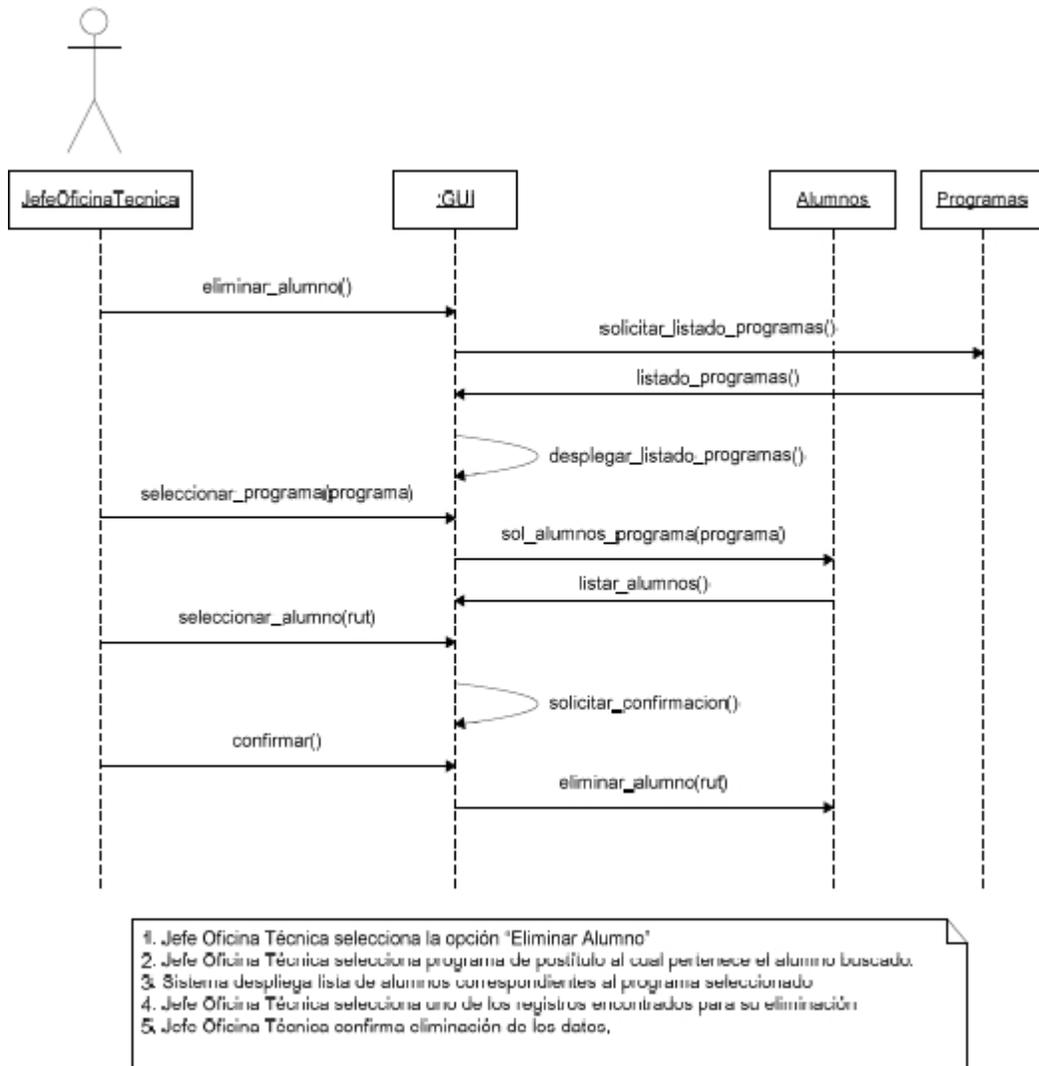


Figura 17. Diagrama de Secuencia – Eliminar Alumnos

3.3.5.-Contratos

Las Tablas 42 a 52 muestran los Contratos provenientes de los Diagramas de Secuencia del Sistema. Estos Contratos son para explicar detalladamente las funciones internas de cada Diagrama de Secuencia, con el fin de lograr un mayor entendimiento de la manera en que dichas funcionalidades son llevadas a cabo.

Nombre	Responsabilidad	Precondiciones	Poscondiciones
generar_estadisticas()	Jefe Oficina Técnica selecciona la opción de generar estadísticas.	Usuario debidamente identificado.	Se debe desplegar el formulario de creación de estadísticas.
Seleccionar_parametros(parámetros)	Jefe Oficina Técnica selecciona los parámetros necesarios para obtener la estadística deseada.		Confirmar la estadística en caso de que esta sea la correcta, para su almacenamiento.
Solic_datos_alumnos(cond_al)	Seleccionar los datos de alumnos que sean necesarios para la creación de la estadística.		Se obtienen los datos de alumnos necesarios para la estadística.
Solic_datos_prog(cond_pr)	Seleccionar los datos de programas que sean necesarios para la creación de la estadística.		Se obtienen los datos de programas que sean necesarios para la creación de la estadística.
Solic_datos_prof(cond_prof)	Seleccionar los datos de profesores que sean necesarios para la creación de la estadística.		Se obtienen los datos de profesores que sean necesarios para la creación de la estadística.

Confirmar()	Jefe Oficina Técnica confirma estadística.		Almacenar estadística en base de datos.
almacenar_estadística(consulta)	Almacenar estadística en base de datos.		Resultados de la estadística quedan almacenados en el sistema, para una posterior visualización.

Tabla 42. Contratos para caso de uso: Generar Estadísticas

Nombre	Responsabilidad	Precondiciones	Poscondiciones
generar_listas()	Jefe Oficina Técnica selecciona la opción de generar listas.	Usuario debidamente identificado.	Se debe desplegar el formulario de creación de listas.
seleccionar_parametros(parámetros)	Jefe Oficina Técnica selecciona los parámetros necesarios para obtener la lista deseada.		Confirmar la lista en caso de que esta sea la correcta, para su almacenamiento.
solic_datos_alumnos(cond_al)	Seleccionar los datos de alumnos que sean necesarios para la creación de la lista.		Se obtienen los datos de alumnos necesarios para la lista.

solic_datos_prog(cond_p r)	Seleccionar los datos de programas que sean necesarios para la creación de la lista.		Se obtienen los datos de programas que sean necesarios para la creación de la lista.
solic_datos_prof(cond_p rof)	Seleccionar los datos de profesores que sean necesarios para la creación de la lista.		Se obtienen los datos de profesores que sean necesarios para la creación de la lista.
Confirmar()	Jefe Oficina Técnica confirma lista.		Almacenar lista en base de datos.
almacenar_estadística(co nsulta)	Almacenar lista en base de datos.		Resultados de la lista quedan almacenados en el sistema, para una posterior visualización.

Tabla 43. Contratos para caso de uso: Generar Listas

Nombre	Responsabilidad	Precondiciones	Poscondiciones
busqueda()	Jefe Oficina Técnica selecciona la opción de búsquedas.	Usuario debidamente identificado.	Se debe desplegar el formulario búsqueda.
seleccionar_parametros(parámetros)	Jefe Oficina Técnica selecciona los parámetros necesarios para		Confirmar la lista en caso de que esta sea la correcta, para su

	realizar la búsqueda.		almacenamiento.
solic_datos_alumnos(cond_al)	Seleccionar los datos de alumnos que sean necesarios para realizar la búsqueda.		Se obtienen los datos del alumno o los alumnos que cumplen con los criterios de búsqueda seleccionados.
solic_datos_prog(cond_pr)	Seleccionar los datos de programas que sean necesarios para realizar la búsqueda.		Se obtienen los datos del programa o los programas que cumplen con los criterios de búsqueda seleccionados.
solic_datos_prof(cond_prof)	Seleccionar los datos de profesores que sean necesarios para realizar la búsqueda.		Se obtienen los datos del profesor o los profesores que cumplen con los criterios de búsqueda seleccionados.

Tabla 44. Contratos para caso de uso: Realizar Búsqueda

Nombre	Responsabilidad	Precondiciones	Poscondiciones
--------	-----------------	----------------	----------------

administrar_programas()	Jefe Oficina Técnica selecciona la opción para administrar los programas del sistema.	Usuario debidamente identificado.	Seleccionar opción a realizar.
seleccionar_opcion()	Jefe Oficina Técnica selecciona opción a realizar.		

Tabla 45. Contratos para caso de uso: Administrar Programas

Nombre	Responsabilidad	Precondiciones	Poscondiciones
ingresar_programa()	Jefe Oficina Técnica selecciona la opción para ingresar un Programa de Postítulo en el sistema.	Administrar programas.	
ingresar_datos_programa (datos)	Jefe Oficina Técnica ingresa los datos necesarios para crear un nuevo programa en el sistema.		Los datos del programa quedan almacenados en el sistema.
almacenar_datos(datos)	Los datos del nuevo programa ingresado quedan almacenados	Todos los datos necesarios deben haber sido	El nuevo programa queda disponible en el sistema para su

	en el sistema.	ingresados.	uso.
--	----------------	-------------	------

Tabla 46. Contratos para caso de uso: Ingresar Programa

Nombre	Responsabilidad	Precondiciones	Poscondiciones
modificar_programa()	Jefe Oficina Técnica selecciona la opción para modificar un Programa de Postítulo en el sistema.	Administrar programas.	
solicitar_listado_programas()	Sistema obtiene la lista de todos los programas de postítulo existentes.	Es necesario que existan registros de programas en el sistema.	Se despliega un listado de los programas registrados en el sistema.
seleccionar_programa_mod(programa)	Jefe Oficina Técnica selecciona programa a modificar.	Listado de los distintos programas de postítulo registrados en el sistema.	Despliegue de los datos del programa seleccionado para su modificación.
solic_datos_prog(programa)	Sistema obtiene los datos del programa seleccionado		Se despliegan los datos del programa seleccionado

modificar_valores(datos_nuevos)	Jefe Oficina Técnica modifica los datos del programa seleccionado.	Haber seleccionado un programa de postítulo.	Los nuevos datos del programa quedan almacenados en el sistema.
actualizar_valores(datos_nuevos)	Sistema almacena los nuevos valores del programa seleccionado.		

Tabla 47. Contratos para caso de uso: Modificar Programa

Nombre	Responsabilidad	Precondiciones	Poscondiciones
cancelar_programa()	Jefe Oficina Técnica selecciona la opción para cancelar un Programa de Postítulo en el sistema.	Administrar programas.	
solicitar_listado_programas()	Sistema obtiene la lista de todos los programas de postítulo existentes.	Es necesario que existan registros de programas en el sistema.	Se despliega un listado de los programas registrados en el sistema.
seleccionar_programa_cancelar(programa)	Jefe Oficina Técnica selecciona programa que desea cancelar..		Se debe solicitar una confirmación previa a la cancelación del

			programa.
confirmar()	Jefe Oficina Técnica confirma la cancelación del sistema.	Haber seleccionado un Programa de Postítulo para cancelar.	Almacenar cambios en base de datos.
cancelar(programa)	Sistema cambia el estado del programa seleccionado a "cancelado".		

Tabla 48. Contratos para caso de uso: Cancelar Programa

Nombre	Responsabilidad	Precondiciones	Poscondiciones
administrar_alumnos()	Jefe Oficina Técnica selecciona la opción para administrar los alumnos del sistema.	Usuario debidamente identificado.	Seleccionar opción a realizar.
seleccionar_opcion()	Jefe Oficina Técnica selecciona opción a realizar.		

Tabla 49. Contratos para caso de uso: Administrar Alumnos

Nombre	Responsabilidad	Precondiciones	Poscondiciones
--------	-----------------	----------------	----------------

ingresar_alumno(datos alumno)	Jefe Oficina Técnica selecciona la opción para ingresar un alumno en el sistema.	Administrar alumnos	Nuevo alumno ingresado al sistema y relacionado a un programa de postítulo.
solicitar_listado_programas()	Sistema obtiene la lista de todos los programas de postítulo existentes.	Es necesario que existan registros de programas en el sistema.	Se selecciona el programa al cual pertenecerá el alumno.
ingresar_datos_alumno(datos)	Jefe Oficina Técnica ingresa los datos del nuevo alumno en el sistema. Se selecciona el Programa de Postítulo al cual pertenecerá el alumno.		Los datos del alumno quedan almacenados en el sistema.
almacenar_alumno(datos)	Los datos del alumno quedan almacenados en el sistema.	Todos los datos necesarios deben haber sido ingresados.	Los datos del alumno quedan disponibles en el sistema para su uso.
sol_datos_alum(rut)	Sistema obtiene los datos del alumno ingresado para su posterior despliegue.		Se despliegan los datos del alumno.

Tabla 50. Contratos para caso de uso: Ingresar Alumno

Nombre	Responsabilidad	Precondiciones	Poscondiciones
modificar_alumno()	Jefe Oficina Técnica selecciona la opción para modificar un alumno en el sistema.	Administrar alumnos.	
solicitar_listado_programas()	Sistema obtiene la lista de todos los programas de postítulo existentes.	Es necesario que existan registros de programas en el sistema.	Se selecciona el programa al que pertenece el alumno.
seleccionar_programa(programa)	Jefe Oficina Técnica selecciona programa al que pertenece el alumno a modificar.		
sol_alumnos_programa(programa)	Sistema obtiene lista de alumnos pertenecientes al programa seleccionado	El programa seleccionado deber tener al menos un alumno asociado.	Se selecciona uno de los alumnos pertenecientes al programa para su posterior modificación
seleccionar_alumno(alumno)	Jefe Oficina Técnica selecciona alumno a modificar.		Modificación de los datos del alumno
sol_datos_alum(rut)	Sistema obtiene los datos del alumno		Se despliegan los datos del alumno

	seleccionado.		seleccionado.
modif_datos_alum(datos_nuevos)	Jefe Oficina Técnica modifica los valores del alumno seleccionado.		Datos modificados quedan almacenados en el sistema.
actualizar_valores(datos_nuevos)	Los datos modificados del alumno son almacenados en el sistema.		Los datos del alumno quedan almacenados en el sistema para su posterior uso.

Tabla 51. Contratos para caso de uso: Modificar Alumno

Nombre	Responsabilidad	Precondiciones	Poscondiciones
eliminar_alumno()	Jefe Oficina Técnica selecciona la opción para eliminar un alumno del sistema.	Administrar alumnos.	
solicitar_listado_programas()	Sistema obtiene la lista de todos los programas de postítulo existentes.	Es necesario que existan registros de programas en el sistema.	Se selecciona el programa al que pertenece el alumno.
seleccionar_programa(programa)	Jefe Oficina Técnica selecciona programa al que pertenece el		

	alumno a eliminar.		
sol_alumnos_programa(programa)	Sistema obtiene lista de alumnos pertenecientes al programa seleccionado	El programa seleccionado deber tener al menos un alumno asociado.	Se selecciona uno de los alumnos pertenecientes al programa para su posterior eliminación.
seleccionar_alumno(rut)	Jefe Oficina Técnica selecciona alumno a eliminar.		Eliminación de los datos del alumno
confirmar()	Jefe Oficina Técnica confirma la eliminación del alumno.	Haber seleccionado un alumno para eliminar.	Almacenar cambios en base de datos.
eliminar_alumno(rut)	Sistema borra los datos del alumno seleccionado.		

Tabla 52. Contratos para caso de uso: Eliminar Alumno

3.4.-DISEÑO

Los distintos diagramas elaborados para construir el sistema tienen distintas funciones. Una función importante es la de ser un lenguaje que permita dar a conocer un sistema mediante gráficos para lograr un mayor entendimiento y una abstracción del sistema a construir.

Otra función de estos diagramas es el ser la base para los pasos siguientes de la construcción de un sistema. Es así entonces que los diagramas realizados en la etapa de análisis son los cimientos para la etapa de diseño, en la cual, basado en pasos anteriores se puede plasmar el sistema que permitirá solucionar los problemas existentes en la Oficina Técnica [LARMAN].

La etapa de diseño contempla una serie de diagramas, los cuales nacen a partir de la etapa de análisis, donde mediante Casos de Uso y Diagramas de Secuencia, se van descubriendo las clases conceptuales y las interacciones entre éstas, permitiendo con esto, estructurar el sistema a construir mediante diagramas.

La Figura 18 muestra el modelo de dominio, el cual es una abstracción de la realidad llevada a un gráfico que muestra las entidades presentes en la vida real y las relaciones entre estas entidades. Esto sirve de base para los pasos próximos, los que contemplan el uso de dichas clases conceptuales para ir estructurando el sistema, pasando del punto de vista de la vida real, hasta un punto de construcción del Software que tiene como objetivo satisfacer la necesidades del cliente, aminorando los problemas existentes en la Oficina Técnica de Estudios Avanzados de la Pontificia Universidad Católica de Valparaíso.

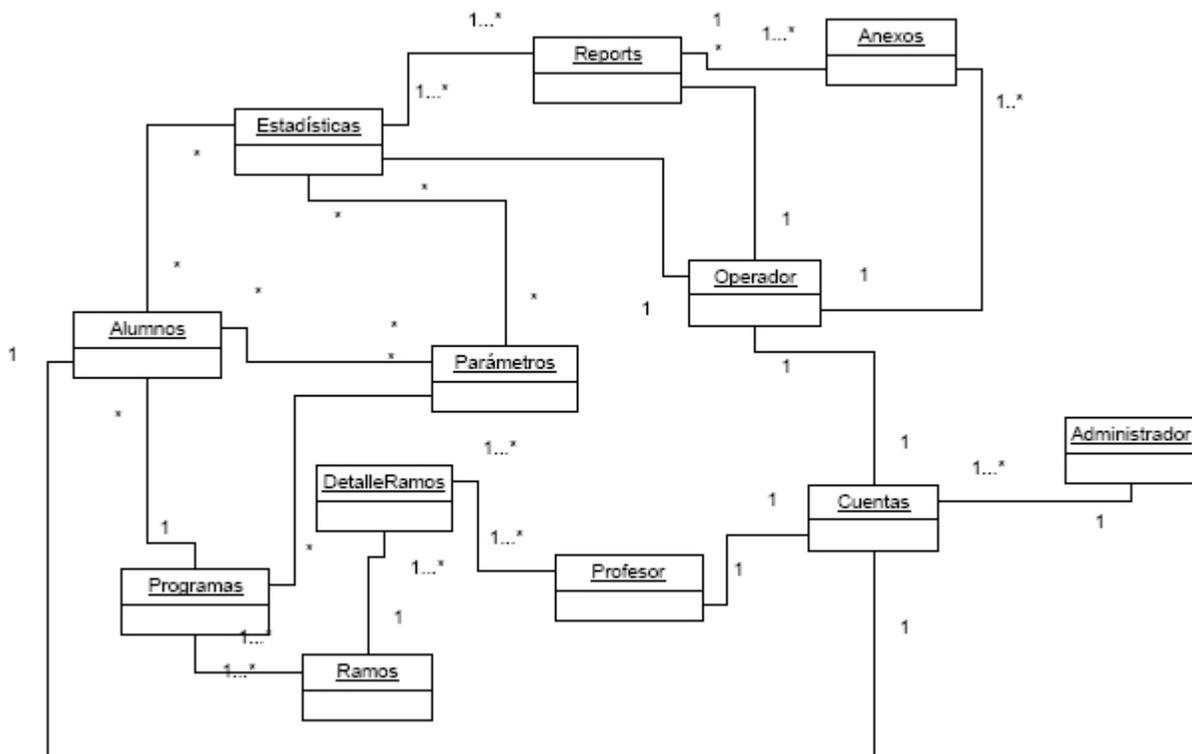


Figura 18. Modelo del Dominio

En la Figura 19 se muestra el Diagrama de Clases, este diagrama muestra las clases que serán implementadas para el desarrollo del sistema y la relación existente entre éstas. Además se pueden conocer los atributos que tendrá cada una de estas clases, lo que permitirá identificarlas y conocer en cierta medida la función que dicha clase tendrá en el sistema.

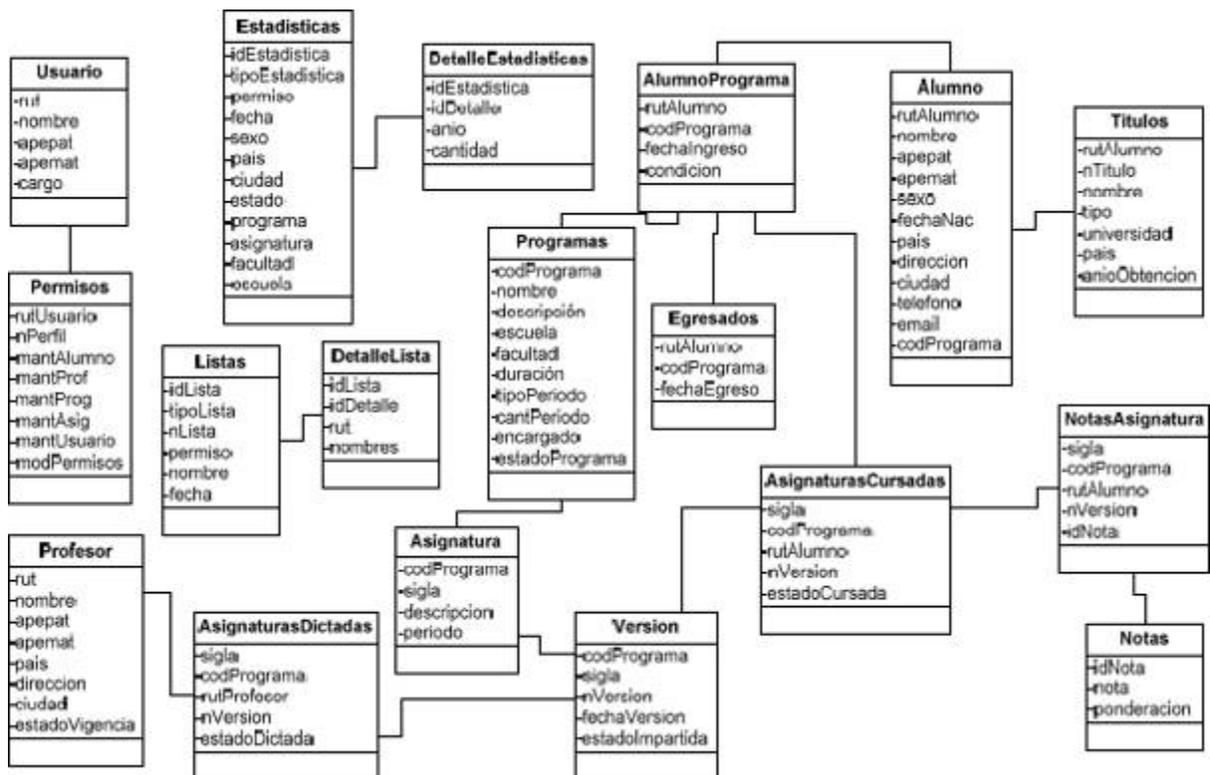


Figura 19. Diagrama de Clases

A continuación, las Figuras 20 y 21 muestran el Diagrama de la Base de Datos. Este diagrama es el que muestra detalladamente cómo es la Base de Datos del sistema, dando a conocer las tablas a utilizar y las relaciones existentes entre éstas.

Cabe destacar que la implementación de la Base de Datos, basada en este diagrama, tiene una importancia fundamental en el proyecto, ya que será la encargada de almacenar los datos que alimentan al sistema para que éste pueda desempeñar sus funciones. Es por este motivo que el

Diagrama de la Base de Datos debe ser explícito y no debe dar cabida a dudas o a redundancias [WEB13].

El estudio realizado para llegar a este Diagrama de Base de Datos considera todos los pasos realizados anteriormente para el desarrollo del sistema, naciendo a partir de los Diagramas de Caso de Uso, Diagramas de Secuencia, Contratos, Modelo de Dominio y Diagrama de Clases, dado que dichos diagramas permiten identificar los distintos componentes que tendrá el sistema y que se deben considerar al momento de pensar en el almacenamiento de los datos.

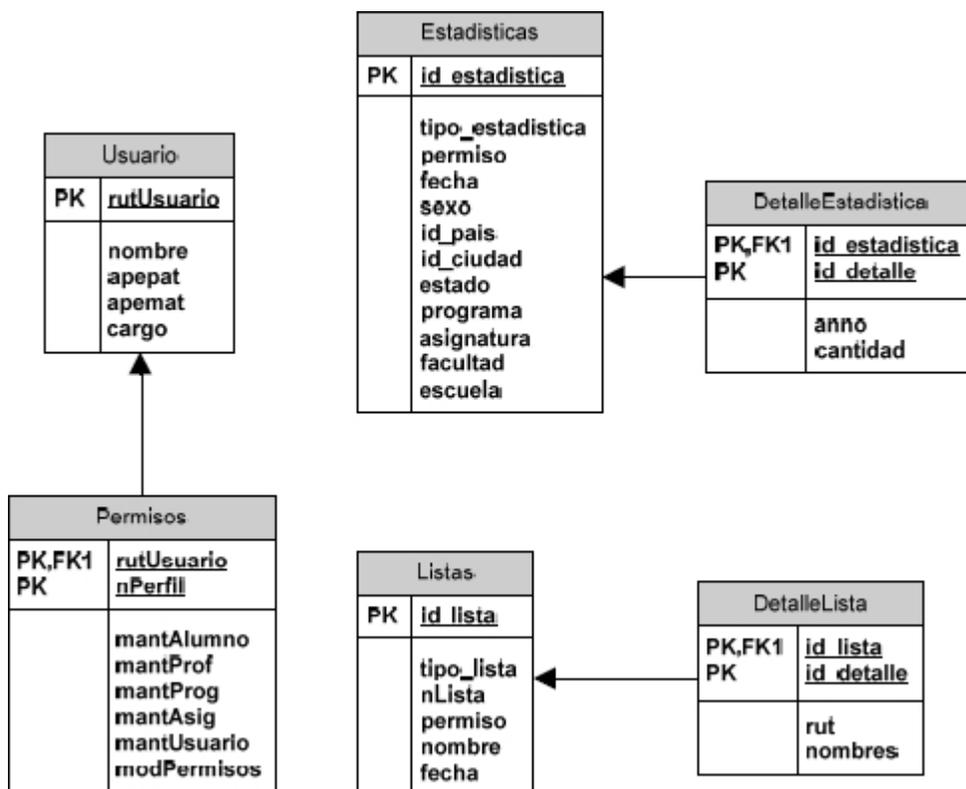


Figura 20. Diagrama BD (parte1)

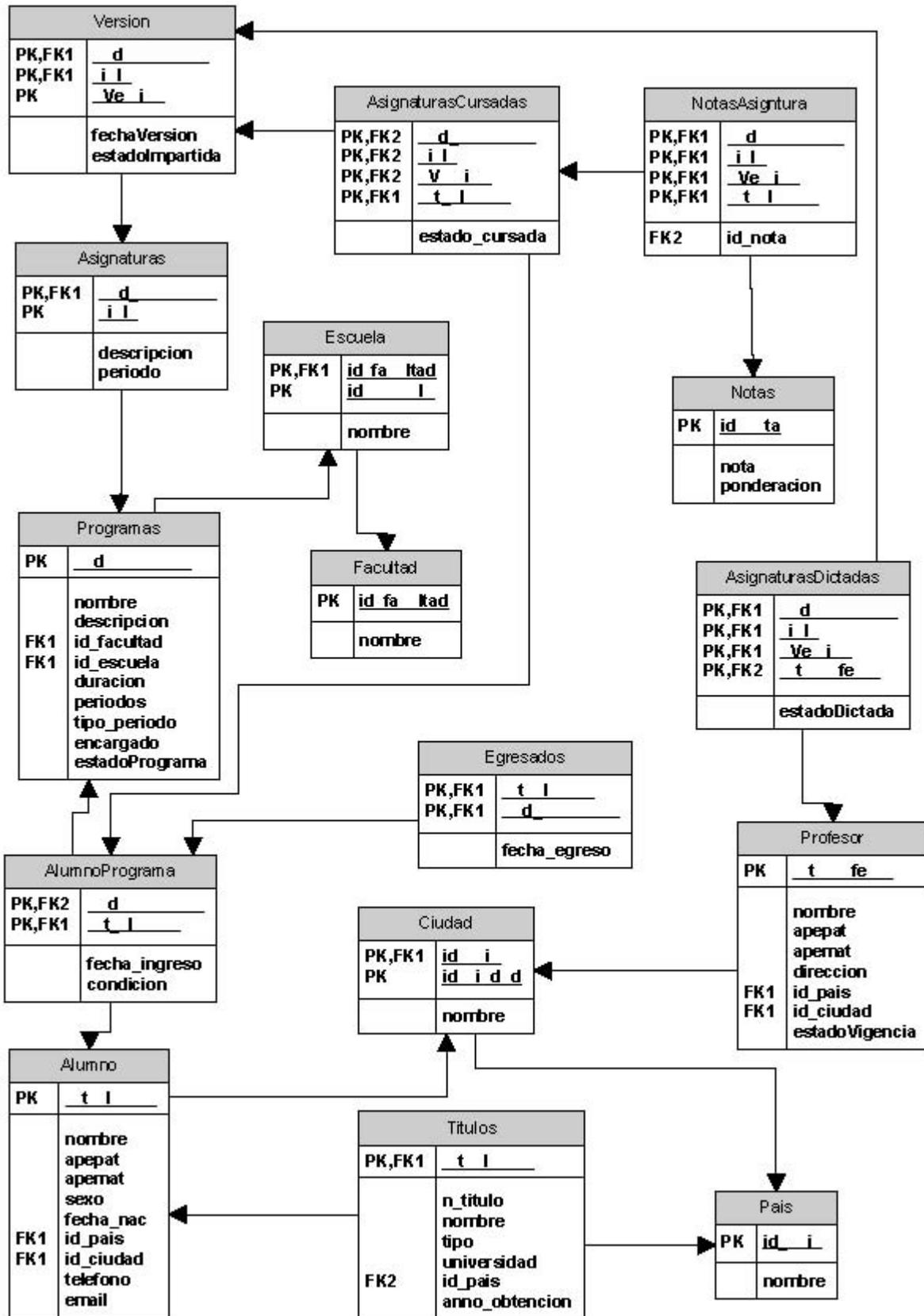


Figura 21. Diagrama BD (parte 2)

En el Diagrama de Componentes que se muestra en la Figura 22 se muestran los distintos elementos contemplados para la implementación de cada una de las distintas capas del sistema.

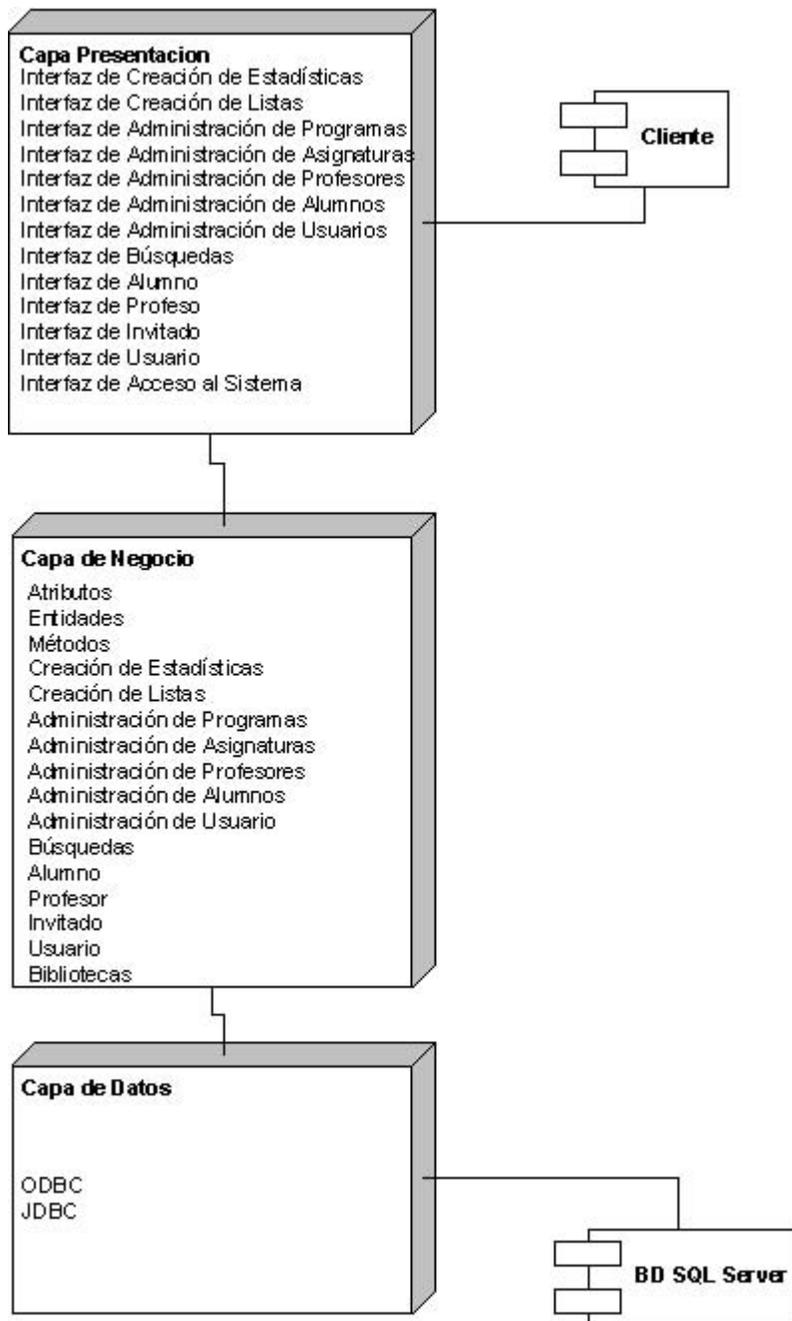


Figura 22. Diagrama de Componentes

Para especificar el Diagrama de Componentes, se muestra y explica el detalle de los componentes de cada una de las capas del sistema.

Los componentes de cada capa simbolizan al conjunto de interfaces que conformará a cada capa. Por ejemplo, el componente “Interfaz de Acceso al Sistema” contempla la realización de la interfaz de acceso así como también la de error, cambio de contraseña u otra interfaz que sea necesaria para dichos efectos.

La Figura 23 muestra los componentes que conforman la Capa de Presentación del sistema. En esta figura se puede ver gráficamente cual es el conjunto de interfaces que tendrá incluida dicha capa.

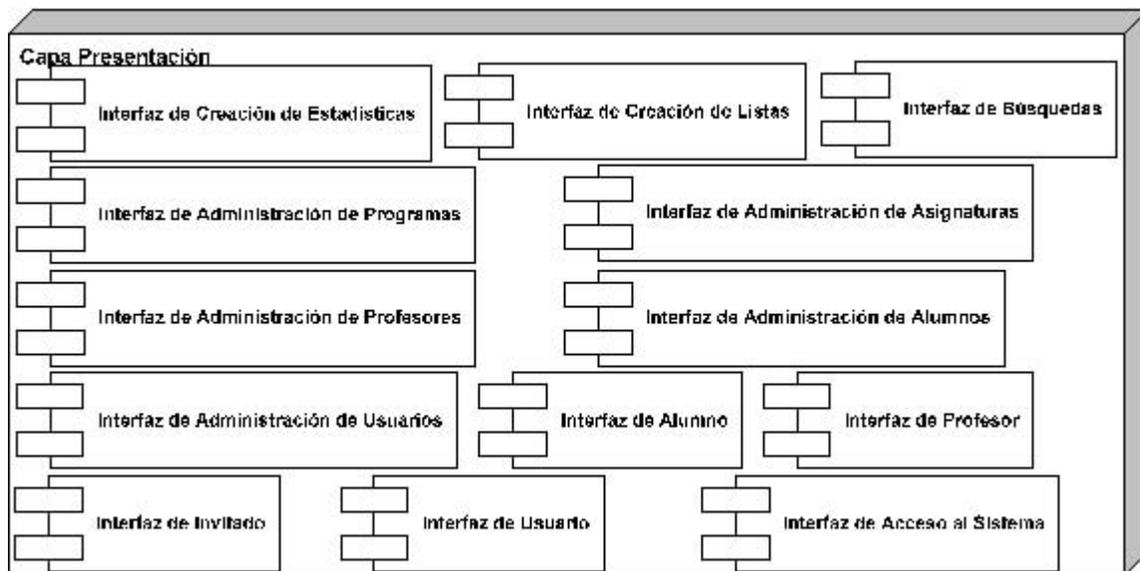


Figura 23. Detalle del diagrama de componentes – Capa Presentación

En la capa de negocio que se muestra en la Figura 24, se pueden encontrar las clases, atributos, métodos y entidades que se usarán en la construcción y que permitirán la interacción entre las interfaces y los datos. Los componentes que comienzan con un símbolo * son archivos que contienen funcionalidades específicas para cada uno de los módulos de la capa de presentación. Estos pueden estar dados por paquetes de clases que contienen funciones ocupadas sólo para un módulo (o interfaz) determinada.

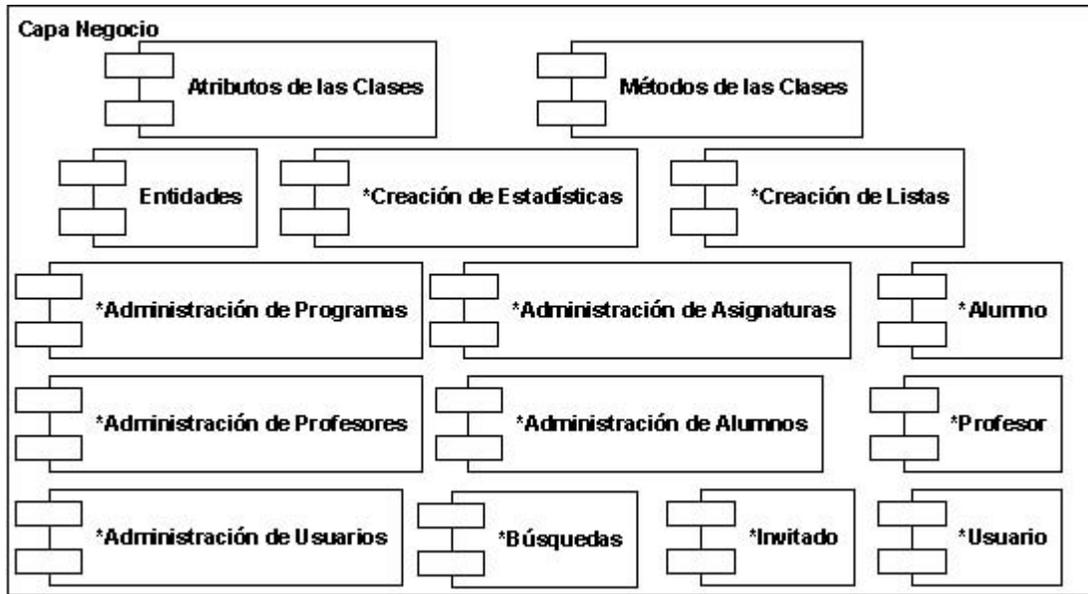


Figura 24. Detalle del diagrama de componentes – Capa Negocio

Por último, la capa de datos, Figura 25, muestra el repositorio de datos (BD SQL Server) y los drivers que permitirán la conexión de las capas hacia los datos. Estos drivers son el ODBC (*Open DataBase Connectivity*) y el JDBC (*Java DataBase Connectivity*).

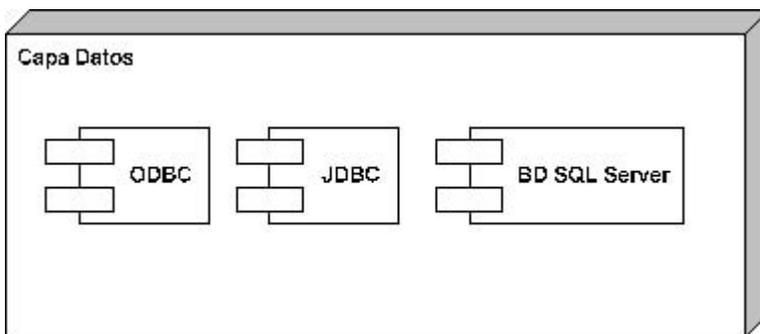


Figura 25. Detalle del diagrama de componentes – Capa Datos

El Diagrama Físico de Componentes detallado en la Figura 26, tiene como función mostrar todos los componentes físicos que se incluirán en el sistema. Para esto se consideran todos los directorios que tendrán alguna participación en el sistema, considerando para esto, los directorios de imágenes, de almacenamiento de interfaces, bibliotecas de datos, clases, etc. Se

considera también el sistema completo como un directorio con subdirectorios en los que se encuentra toda la información necesaria para el buen funcionamiento de SAPP.

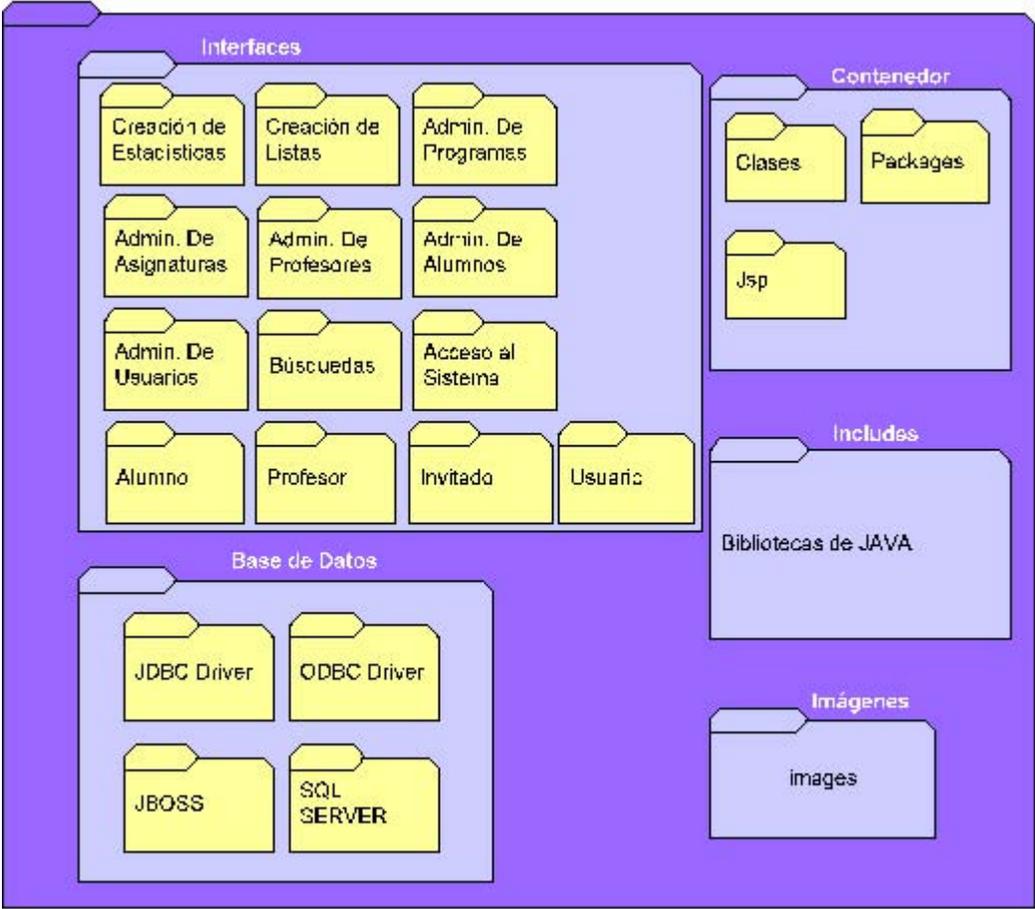


Figura 26. Diagrama Físico de Componentes

La Figura 27 muestra como se compondrá cada uno de los módulos del sistema. Es necesario considerar que la seguridad es transversal a cada uno de los módulos y que se integrará mediante código Java sobre cada página creada.

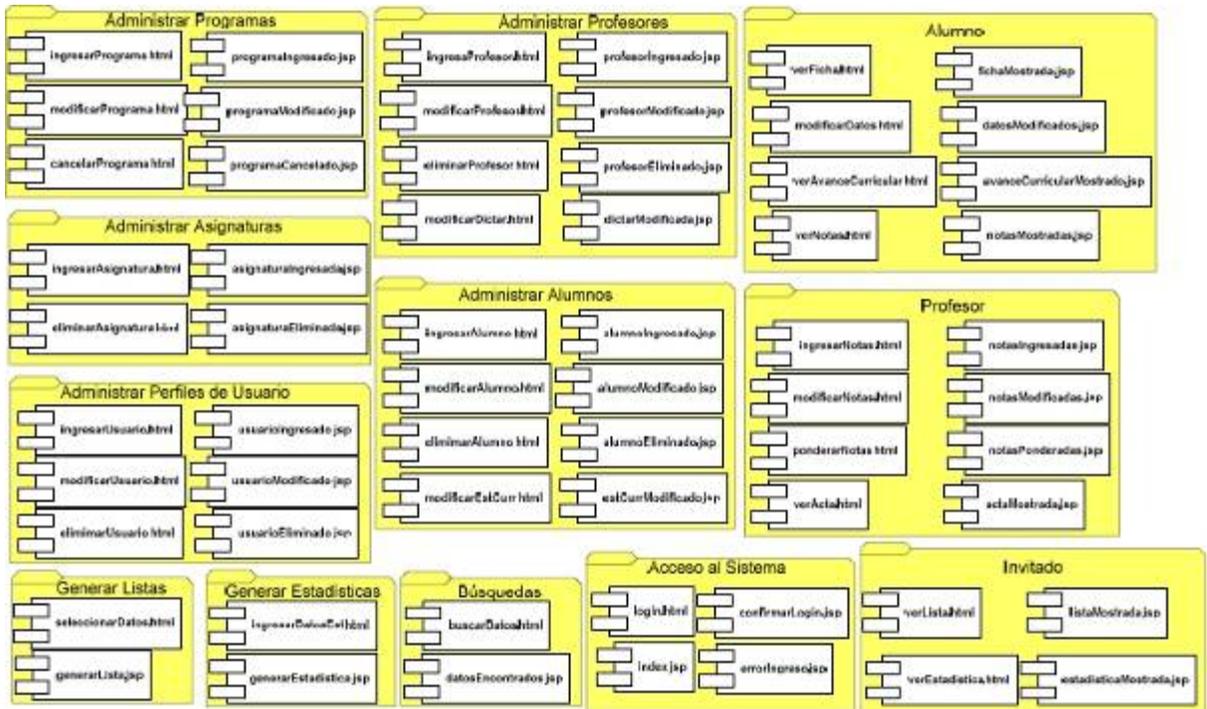


Figura 27. Detalle de los módulos

3.5.-ESTÁNDARES DE CODIFICACIÓN

Con el fin de que los programadores del proyecto trabajen en forma coordinada, es necesario especificar estándares de codificación. Estos estándares deben ser prácticos, prudentes y deben permitir la creación de un código armonioso, como si un único programador hubiera escrito todo el código de una sola vez.

La legibilidad del código fuente repercute directamente en lo bien que un programador comprende un sistema de software. La mantenibilidad del código es la facilidad con que el sistema de software puede modificarse para añadirle nuevas características, modificar las ya existentes, depurar errores, o mejorar el rendimiento. Si el código no ha seguido un estándar esta tarea resultará tediosa, compleja y requerirá muchos recursos.

La adopción de un estándar de codificación sólo es viable si se sigue desde el principio hasta el final del proyecto de software. No es práctico, ni prudente, imponer un estándar de codificación una vez iniciado el trabajo.

Usar técnicas de codificación sólidas y realizar buenas prácticas de programación con vistas a generar un código de alta calidad es de gran importancia para la calidad del software y para obtener un buen rendimiento. Además, si se aplica de forma continuada un estándar de codificación bien definido, se utilizan técnicas de programación apropiadas, y posteriormente se efectúan revisiones del código de rutinas, caben muchas posibilidades de que un proyecto de software se convierta en un sistema de software fácil de comprender y de mantener.

Las técnicas de codificación incorporan muchos aspectos del desarrollo del software. Aunque generalmente no afectan a la funcionalidad de la aplicación, sí contribuyen a una mejor comprensión del código fuente.

Las técnicas de codificación que se implementan son las que se describen a continuación.

Utilizar variables de poca longitud.

Mantener el tiempo de vida de las variables tan corto como sea posible, esto es muy importante por ejemplo cuando se utiliza un recurso finito como una conexión a una Base de Datos.

Utilizar procedimientos y variables con un solo propósito. Se evitará crear procedimientos multipropósito que lleven a cabo una variedad de funciones no relacionadas.

Dentro de una clase, se evitará el uso de variables públicas, en cambio se utilizarán procedimientos y propiedades que accedan a dichas variables (privadas), así se tendrá una capa de encapsulación y se brindará la posibilidad de validar valores de cambio sobre las mismas, antes de manipularlas directamente.

Optimizar las conexiones a datos usando credenciales específicas de usuario, ya que éstas no podrán ser reutilizadas por el administrador de conexiones.

Evitar el uso de conversión de tipos de datos ya que esto puede generar resultados imprevistos, sobre todo cuando dos variables están involucradas en una sentencia.

Utilizar en la medida de lo posible rutinas de manejo de excepciones.

Ser específico cuando se declaren objetos que puedan generar colisión, por ejemplo si tiene dos métodos con el mismo nombre en diferentes clases.

El esquema de nombres es una de las ayudas más importantes para entender el flujo lógico de una aplicación. Un nombre debe más bien expresar el "qué" que el "cómo", por ejemplo es más claro nombrar un procedimiento de acceso a datos seleccionarPrograma() que realizarConsultaSelect(), porque lo que importa (para que otra persona entienda el código) es que se supone que hace el método y no cómo lo hace.

Otra directiva es la de utilizar nombres tan largos como para que se entiendan pero a la vez tan cortos como para que no den información irrelevante.

Los nombres de todas las estructuras de código deben ser en español.

El nombre de la clase y el archivo fuente deben ser iguales.

Evitar nombres imprecisos que permitan interpretaciones subjetivas, como por ejemplo X1 o X2 para una variable.

En la POO es redundante incluir nombres de clases en el nombre de las propiedades de clases, como por ejemplo Programa.ProgramaIngresar, en su lugar, se utilizará Programa.Ingresar, pues el nombre de la clase ya contiene dicha información.

Las variables miembro se escribirán con letra minúscula. En caso que el nombre de la variable sea de más de una palabra, la primera palabra será con minúsculas y la primera letra de cada palabra concatenada será en mayúscula. Por ejemplo la variable se denominará "codigoPrograma" en vez de "CodigoPrograma", "codigo_programa" u otra combinación.

Para el caso de una variable de poco uso, que deba aparecer sólo en unas cuantas líneas de código, se empleará un nombre descriptivo. Se utilizará nombres de variables de una sola letra, como i o j sólo para índices (por ejemplo ciclos for).

Los archivos y los nombres de carpetas, al igual que los nombres de procedimientos, deberán describir claramente su finalidad.

3.6.-PLAN DE PRUEBAS

A excepción de los programas pequeños, los sistemas no se prueban como una simple unidad monolítica. Los sistemas grandes se construyen a partir de subsistemas que a su vez se construyen a partir de módulos que están compuestos de procedimientos y funciones. Por lo tanto, el proceso de prueba se lleva a cabo en etapas en las que las pruebas se aplican de forma incremental en conjunto con la implementación del sistema.

Se han reconocido los siguientes módulos dentro del sistema:

Módulo de creación de Estadísticas.

Módulo de creación de Listas.

Módulo de Administración de Programas de Postítulo.

Módulo de Administración de Asignaturas.

Módulo de Administración de Profesores.

Módulo de Administración de Alumnos.

Módulo de Búsquedas.

Módulo de Administración de Perfiles de Usuario.

La etapa de prueba se lleva a cabo en cuatro subetapas:

Prueba de unidades.

Prueba de integración.

Prueba de sistema.

Prueba de aceptación.

Prueba de unidades

Se prueban los componentes individuales para asegurarse de que operan correctamente. Cada uno se prueba de forma independiente, sin los otros componentes del sistema. Consiste esencialmente en probar cada método de las clases implementadas por separado.

Como este tipo de prueba es el primero de la etapa de pruebas, posee una gran importancia, ya que si las unidades están mal evaluadas no tiene sentido evaluar los módulos, ya que si las unidades poseen errores, los módulos los acarrearán.

Por lo antes dicho, es preciso probar el flujo de datos de la interfaz del módulo, además de las estructuras de datos locales. Durante la prueba de unidad se debe comprobar el impacto de los datos globales sobre el módulo.

Los módulos que requieren de prueba unitaria son:

Módulo de Administración de Programas de Postítulo.

Módulo de Administración de Asignaturas.

Módulo de Administración de Profesores.

Módulo de Administración de Alumnos.

Módulo de Administración de Perfiles de Usuario.

Módulo Alumno.

Módulo Profesor.

Módulo Usuario.

Módulo Invitado.

Prueba de integración

Una vez que se han probado los componentes individuales de programa, deben integrarse para crear un sistema parcial, o completo. Este proceso de integración comprende la construcción del sistema y probar el sistema resultante con respecto a los problemas que surjan de las interacciones de los componentes.

Debido a que el software orientado a objeto no posee una estructura de control jerárquica, las estrategias convencionales de integración ascendente y descendente poseen un significado pequeño. Por lo mismo, existen dos distintas estrategias para evaluar la integración de los módulos, siendo éstas:

Prueba basada en hilos.

Prueba basada en usos.

Prueba basada en hilos

Esta integra el conjunto de clases necesarias para responder a una entrada o evento del sistema. Cada hilo se integra y prueba individualmente.

Prueba basada en usos

Consiste en dividir las clases en capas e ir probando por cada nivel, la primera capa debe poseer todas las clases independientes, luego se prueba el siguiente nivel donde se encuentran las llamadas capas dependientes, que son las que utilizan las dependientes, y así sucesivamente hasta llegar hasta el sistema completo.

En este caso se utilizará la prueba basada en usos, ya que se puede apreciar con mayor claridad las capas y las interacciones entre las clases, por lo mismo, se ocupará menos tiempo en este tipo de integración.

Prueba de sistema

Luego de llevar a cabo la prueba de unidad y la de integración, se debe probar el sistema en si. Este tipo de prueba consiste en encontrar los errores que son el resultado de interacciones no previstas entre los módulos y la interfaz. También pretende validar que el sistema cumpla sus requerimientos funcionales y no funcionales además de las características adicionales a dichos requerimientos que tenga el sistema.

Este tipo de prueba está constituida por otras pruebas, estas pruebas en si poseen distintos propósitos, pero en su conjunto apuntan a verificar que se hayan integrado adecuadamente todos los elementos del sistema y que realizan las funciones apropiadas. Estas pruebas son:

Prueba de regresión.

Prueba de seguridad.

Prueba de resistencia.

Prueba de rendimiento.

Prueba de regresión.

La prueba de regresión permite verificar que ante alguna reparación de un error encontrado, el sistema siga consistente y que dicha reparación no haya alterado algún otro componente del sistema.

Prueba de seguridad.

Esta prueba verifica que los mecanismos de protección incorporados en el sistema lo protegerán, de hecho, de accesos impropios.

Prueba de resistencia.

Esta prueba ejecuta un sistema de forma que demande recursos en cantidad, frecuencia o volúmenes anormales, en otras palabras, quiere llegar a ver hasta qué punto de carga resiste el sistema.

Prueba de rendimiento.

Verifica el rendimiento del software en tiempo de ejecución dentro del contexto de un sistema integrado. Este tipo de prueba es usado para sistemas de tiempo real y empotrados. Por lo mismo esta prueba no es considerada dentro del desarrollo del proyecto.

Prueba de aceptación.

Ésta es la etapa final en el proceso de pruebas antes de que el sistema se ponga en operación. Esta prueba la debe realizar el cliente. Debido a la diferencia entre los datos reales y de prueba, la prueba de aceptación revela errores y omisiones en la definición de requerimientos del sistema. También revela donde los problemas de requerimientos en los recursos del sistema no cumplen las necesidades del usuario o donde el desempeño del sistema es inaceptable. Dentro de esta prueba cae lo que es el manual de usuario.

Para realizar las tres primeras etapas del plan de prueba, que son de unidad, integración y sistema se utilizarán ciertos métodos los cuales serán:

Caja blanca.		
Caja		negra.
○	Partición	equivalente.
○	Análisis de valores límites.	

Caja blanca.

La prueba de caja blanca es un método de diseño de casos de prueba que usa la estructura de control de diseño procedimental para obtener los casos de prueba. Mediante los métodos de prueba de caja blanca, se pueden obtener los casos de prueba que garanticen que:

Se ejerciten por lo menos una vez todos los caminos independientes de cada módulo.
Se ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
Se ejecuten todos los bucles en sus límites y con sus límites operacionales.
Se ejerciten las estructuras internas de datos para asegurar su validez.

Además, considera el hecho de los errores de escritura que se puedan producir al momento de trasladar el código fuente a un lenguaje de programación determinado.

Caja negra.

Este método de prueba se centra en los requisitos funcionales de software, es decir, permite al ingeniero de software obtener conjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa. Este método es un complemento al de caja blanca, ya que intenta descubrir diferentes tipos de errores que el método de caja blanca no ha encontrado. La prueba de caja negra intenta encontrar errores que caen en las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización y de terminación.

La caja negra se identifica como una entidad reconocible a la cual llegan diferentes entradas y de la cual salen una o varias salidas. El funcionamiento de ésta consiste en que la entrada que llega sea procesada dentro de la caja, y la salida que proporcione sea la transformación de dicha entrada, no importando el proceso que se lleve a cabo para transformar los valores de entrada en salidas.

Partición equivalente.

La partición equivalente es un método de prueba que divide el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de errores que, de otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico. La partición equivalente se dirige a la definición de casos de prueba que descubran clases de errores, reduciendo así el número total de casos de prueba que hay que diseñar.

Análisis de valores límites.

Como los errores tienden a darse más en los límites del campo de entrada que en centro, se ha desarrollado este método, que consiste en una técnica de diseño de casos de prueba que complementa a la partición equivalente. En lugar de seleccionar cualquier elemento de equivalencia, esta técnica lleva la elección de casos de prueba en los extremos de la clase. Por el mismo lado, en vez de centrarse solamente en las condiciones de entrada, este método obtiene casos de prueba también para campos de salida.

3.6.1.-Tipos de Pruebas Utilizadas

Las pruebas a considerar en SAPP son las pruebas unitarias e integradas antes de la implementación del sistema y las pruebas de aceptación una vez que el sistema esté implementado. Estas pruebas aún no han sido llevadas a cabo debido a que aún no se ha llegado a la implementación del sistema.

Las pruebas unitarias e integradas son realizadas utilizando el método de Caja Negra, propiamente tal, acompañado esto de inspecciones de código a modo de Caja Blanca.

Las pruebas de aceptación deben ser ejecutadas por los usuarios del sistema, donde la idea es utilizar el sistema realizando las funciones cotidianas de cada usuario (previa capacitación). La idea de realizar estas pruebas es que el usuario pueda encontrar errores en el sistema que no hayan sido captados en las etapas de pruebas anteriores. La elección de los tipos de prueba se basa en el hecho que las pruebas unitarias se van haciendo durante toda la programación del sistema. Al verlo de un punto de vista práctico, el programador va probando cada una de las piezas de software que va desarrollando. De esta misma manera se va revisando el código, se depura y optimiza. Las pruebas integradas van desarrollándose en la medida que existen módulos relacionados entre sí. Estas pruebas consideran las interacciones con la base de datos.

Una vez terminada la programación se ejecutan pruebas diseñadas previamente, las cuales tienen como fin probar todas las distintas funcionalidades del sistema.

Además de las pruebas diseñadas, y las realizadas en el momento de la programación, se incluye en la etapa de pruebas un checklist de consideraciones mínimas. Este checklist tiene

una serie de aspectos que deben ser considerados al momento de programar el sistema y que no tiene un fin funcional, sino que apunta más a la usabilidad del sistema y al correcto funcionamiento de éste a nivel de interfaz.

3.6.2.-Casos de Prueba

Con el fin de testear el sistema, se llevaron a cabo una serie de pruebas a nivel funcional, las que verificaban las distintas partes que componían SAPP. Mediante estas pruebas se logró encontrar errores y eliminarlos. Estas pruebas unitarias fueron parte de un flujo que permitió encontrar errores una vez integrado el sistema completo.

Además de las pruebas unitarias, se llevó a cabo un checklist de consideraciones mínimas, el cual fijó reglas a cumplir, que aseguraban una mayor calidad del sistema. Las reglas de este checklist eran demasiado pequeñas para ser probadas por si solas como pruebas unitarias, pero la omisión de dichas reglas no permite llevar a cabo un sistema de calidad. La función del checklist es tener en cuenta las validaciones, los campos en blanco, errores de ortografía, entre otros, por lo que su utilización durante la programación, sirvió, junto a los estándares de programación, para desarrollar un sistema robusto y detallado.

A continuación se detallan las pruebas unitarias y el checklist de consideraciones mínimas, realizados sobre la aplicación.

Pruebas Unitarias

CASO DE PRUEBA N°1.		
Módulo Relacionado	Crear Programa	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Crear correctamente un programa ingresando valores válidos en todos los campos	Se debe crear correctamente el programa.	OK

obligatorios.		
Crear programa sin ingresar valores en ninguno de sus campos.	No se debe permitir la creación del programa. Se debe desplegar un mensaje de error indicando que faltan datos.	OK
Crear programa ingresando valores sólo en algunos de los campos obligatorios.	No se debe permitir la creación del programa. Se debe desplegar un mensaje de error indicando que faltan datos.	OK
Crear programa ingresando un código de programa ya existente.	No se debe permitir la creación del programa. Se debe desplegar un mensaje de error indicando que el programa ingresado ya existe.	OK
Crear correctamente un programa y verificar que la creación fue exitosa (Búsqueda de programas).	En el escenario de búsqueda se debe desplegar correctamente los datos del programa creado.	OK
Asignar correctamente la dicción de un programa	El programa debe quedar con la dicción asociada al periodo señalado	OK
Asignar una dicción en un periodo anterior	No se debe permitir asignar una dicción a un programa para un periodo de tiempo ya expirado.	OK

Verificar asignaturas de un programa	Se deben desplegar correctamente las asignaturas que el programa tiene.	OK
Activar un programa sin que éste tenga asignaturas asignadas	No se debe permitir activar un programa sin asignaturas.	OK

Tabla 53. Caso de prueba 1: Crear Programa

CASO DE PRUEBA N°2.		
Módulo Relacionado	Modificar Programa	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Modificar los valores de todos los campos obligatorios, ingresando valores válidos.	Se debe modificar correctamente el programa.	OK
Modificar programa dejando en blanco los campos obligatorios de éste.	No se debe permitir la modificación del programa. Se debe desplegar un mensaje de error indicando que faltan datos.	OK
Modificar programa dejando en blanco algunos de los campos obligatorios e ingresando valores válidos en los otros.	No se debe permitir la modificación del programa. Se debe desplegar un mensaje de error indicando que faltan datos.	OK

Modificar un programa ingresando un código de programa ya existente.	No se debe permitir la modificación del programa. Se debe desplegar un mensaje de error indicando que el programa ingresado ya existe.	OK
Modificar correctamente un programa y verificar que la modificación fue exitosa (Búsqueda de programas).	En el escenario de búsqueda se debe desplegar correctamente los datos del programa modificado.	OK
Modificar los datos de un programa y salir del escenario de modificación sin guardar los cambios realizados. Verificar los datos del programa (Búsqueda de Programas).	Los datos del programa deben encontrarse sin modificar.	OK

Tabla 54. Caso de prueba 2: Modificar Programa

CASO DE PRUEBA N°3.		
Módulo Relacionado	Cancelar Programa	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Seleccionar un programa para cancelar y cancelarlo.	Se debe realizar correctamente la cancelación del programa seleccionado. En el escenario de búsqueda NO debe aparecer el	OK

	programa cancelado.	
--	---------------------	--

Tabla 55. Caso de prueba 3: Cancelar Programa

CASO DE PRUEBA N°4.		
Módulo Relacionado	Crear Asignatura	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Crear una asignatura ingresando valores válidos en todos los campos obligatorios.	Se debe crear correctamente la asignatura	OK
Crear una asignatura ingresando valores en algunos de sus campos obligatorios	No se debe permitir la creación de la asignatura. Se debe desplegar un mensaje indicando que la faltan datos	OK
Crear una asignatura sin ingresar valores en sus campos obligatorios	No se debe permitir la creación de la asignatura. Se debe desplegar un mensaje indicando que la faltan datos	OK
Vincular una asignatura a un programa	Se debe vincular la asignatura al programa seleccionado	OK

Tabla 56. Caso de prueba 4: Crear Asignatura

CASO DE PRUEBA N°5.

Módulo Relacionado			Eliminar Asignatura		
Prueba Realizada		Resultado Esperado		Estado de la Prueba	
Eliminar una asignatura del sistema		Se debe eliminar correctamente la asignatura seleccionada		OK	
Eliminar una asignatura que fue previamente eliminada		No se debe permitir dicha eliminación. La asignatura eliminada no debe aparecer nuevamente en la lista de asignaturas que se pueden eliminar		OK	
Eliminar una asignatura y crear una nueva asignatura con los datos de la asignatura eliminada		Se debe crear correctamente la asignatura		OK	

Tabla 57. Caso de prueba 5: Eliminar Asignatura

CASO DE PRUEBA N°6.

Módulo Relacionado			Crear Alumno		
Prueba Realizada		Resultado Esperado		Estado de la Prueba	
Crear un alumno ingresando valores validos en todos los campos obligatorios.		Se debe crear correctamente el alumno		OK	

Crear un alumno dejando en blanco alguno de los valores obligatorios.	No se debe permitir la creación del alumno. Se debe desplegar un mensaje indicando que faltan datos	OK
Crear un alumno dejando en blanco todos los campos obligatorios	No se debe permitir la creación del alumno. Se debe desplegar un mensaje indicando que faltan datos	OK
Crear un alumno ingresando valores en los campos obligatorios pero sin asociar un título previo al alumno	No se debe permitir la creación del alumno. Se debe desplegar un mensaje indicando que el alumno debe tener al menos un título académico previo para poder ser ingresado.	OK
Crear un alumno ingresando el Rut de un alumno ya existente	No se debe permitir la creación del alumno. Se debe desplegar un mensaje indicando que el Rut ingresado ya existe en el sistema	OK
Verificar la creación de un alumno, ingresándolo correctamente y verificando los datos en la lista de alumnos del programa al cual se ingresó	Se debe desplegar los datos del nuevo alumno en la lista de alumnos pertenecientes al programa.	OK
Ingresar los datos para la creación de un nuevo alumno y salir del escenario de creación sin guardar los datos ingresados	No se debe crear el alumno.	OK

Tabla 58. Caso de prueba 6: Crear Alumno

CASO DE PRUEBA N°7.

Caso de Prueba N°7		
Módulo Relacionado	Modificar Alumno	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Modificar un alumno ingresando valores validos en todos los campos obligatorios.	Se debe modificar correctamente el alumno	OK
Modificar un alumno dejando en blanco alguno de los valores obligatorios.	No se debe permitir la modificación del alumno. Se debe desplegar un mensaje indicando que faltan datos	OK
Modificar un alumno dejando en blanco todos los campos obligatorios	No se debe permitir la modificación del alumno. Se debe desplegar un mensaje indicando que faltan datos	OK
Modificar un alumno ingresando el Rut de un alumno ya existente	No se debe permitir la modificación del alumno. Se debe desplegar un mensaje indicando que el Rut ingresado ya existe en el sistema	OK
Modificar un alumno y verificar la modificación en su ficha.	Se debe desplegar los datos modificados en la ficha del alumno	OK

Modificar los datos de un alumno y salir del escenario de modificación sin guardar los cambios	No se deben modificar los datos del alumno	OK
Ingresar como alumno al sistema y modificar datos propios	Sólo puede desplegarse de forma editable, el e-mail y el teléfono del alumno en cuestión.	OK
Ingresar como alumno al sistema y modificar el nombre propio	No se debe permitir cambiar datos que no sean o el e-mail o el teléfono del alumno.	OK

Tabla 59. Caso de prueba 7: Modificar Alumno

CASO DE PRUEBA N°8.

Módulo Relacionado		
Eliminar Alumno		
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Eliminar un alumno del sistema	Se debe eliminar correctamente el alumno seleccionado	OK
Eliminar un alumno que fue previamente eliminado	No se debe permitir dicha eliminación. El alumno eliminado no debe aparecer nuevamente en la lista de alumnos que se pueden eliminar	OK

Eliminar un alumno y verificar su eliminación mediante el acceso al sistema.	No se debe permitir el acceso al sistema al alumno eliminado	OK
Eliminar un alumno y crear un nuevo alumno con los datos de alumno eliminado	Se debe crear correctamente el alumno	OK

Tabla 60. Caso de prueba 8: Eliminar Alumno

CASO DE PRUEBA N°9.		
Módulo Relacionado	Crear Usuario	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Crear un usuario ingresando valores validos en todos los campos obligatorios.	Se debe crear correctamente el usuario	OK
Crear un usuario dejando en blanco alguno de los valores obligatorios.	No se debe permitir la creación del usuario. Se debe desplegar un mensaje indicando que faltan datos	OK
Crear un usuario dejando en blanco todos los campos obligatorios	No se debe permitir la creación del usuario. Se debe desplegar un mensaje indicando que faltan datos	OK

Crear un usuario ingresando el Rut de un usuario ya existente	No se debe permitir la creación del usuario. Se debe desplegar un mensaje indicando que el Rut ingresado ya existe en el sistema	OK
Verificar la creación de un usuario, ingresándolo correctamente y la creación mediante el acceso al sistema	Se debe permitir el acceso al sistema al usuario recién creado	OK
Ingresar los datos para la creación de un nuevo usuario y salir del escenario de creación sin guardar los datos ingresados	No se debe crear el usuario.	OK
Asignar permisos a un nuevo usuario y verificarlos en el acceso al sistema	Al ingresar al sistema, deben estar disponibles todas las acciones que fueron permitidas al momento de asignar el permiso al usuario.	OK
Asignar permisos a un usuario e intentar realizar acciones que no fueron permitidas en los permisos otorgados	No se debe permitir realizar acciones que no fueron asociadas al momento de la asignación de permisos al nuevo usuario.	OK

Tabla 61. Caso de prueba 9: Crear Usuario

CASO DE PRUEBA N° 10.

Módulo Relacionado

Modificar Usuario

Prueba Realizada	Resultado Esperado	Estado de la Prueba
Modificar un usuario ingresando valores validos en todos los campos obligatorios.	Se debe modificar correctamente el usuario	OK
Modificar un usuario dejando en blanco alguno de los valores obligatorios.	No se debe permitir la modificación del usuario. Se debe desplegar un mensaje indicando que faltan datos	OK
Modificar un usuario dejando en blanco todos los campos obligatorios	No se debe permitir la modificación del usuario. Se debe desplegar un mensaje indicando que faltan datos	OK
Modificar un usuario ingresando el Rut de un usuario ya existente	No se debe permitir la modificación del usuario. Se debe desplegar un mensaje indicando que el Rut ingresado ya existe en el sistema	OK
Modificar un usuario, realizando cambios en sus datos y en sus permisos y verificar la modificación mediante el acceso al sistema	Se debe desplegar los datos modificados del usuario y se debe tener acceso a las nuevas funcionalidades dadas mediante la modificación de los permisos	OK
Modificar los datos de un usuario y salir del escenario de modificación sin guardar los	No se deben modificar los datos del usuario	OK

cambios		
Ingresar como usuario con permisos de sólo lectura al sistema y modificar datos propios	No se debe permitir la modificación	OK

Tabla 62. Caso de prueba 10: Modificar Usuario

CASO DE PRUEBA N°11.		
Módulo Relacionado	Eliminar Usuario	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Eliminar un usuario del sistema	Se debe eliminar correctamente el usuario seleccionado	OK
Eliminar un usuario que fue previamente eliminado	No se debe permitir dicha eliminación. El usuario eliminado no debe aparecer nuevamente en la lista de usuarios que se pueden eliminar	OK
Eliminar un usuario y verificar su eliminación mediante el acceso al sistema.	No se debe permitir el acceso al sistema al usuario eliminado	OK
Eliminar un usuario y crear un nuevo usuario con los datos de	Se debe crear correctamente el usuario	OK

usuario eliminado		
-------------------	--	--

Tabla 63. Caso de prueba 11: Eliminar Usuario

CASO DE PRUEBA N°12.		
Módulo Relacionado	Login	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Ingresar como administrador al sistema	Se deben desplegar todas las funciones que tiene el administrador del sistema	OK
Ingresar como administrador del sistema y realizar el ingreso de un nuevo programa	Se debe permitir el acceso a dichas funcionalidades	OK
Ingresar como administrador del sistema e ingresar al escenario de modificación de un alumno	Se debe desplegar la ficha del alumno con todos los datos de forma editable	OK
Ingresar como alumno al sistema e ingresar al escenario de modificación	Sólo se debe permitir la modificación del propio e-mail y número telefónico	OK
Ingresar como usuario no administrador que no posea permisos de creación, e ingresar un nuevo alumno	No se debe permitir el acceso al módulo de creación de alumnos	OK

Ingresar como invitado al sistema y revisar una lista previamente creada	Se debe desplegar correctamente la lista seleccionada	OK
Ingresar como invitado al sistema y revisar una estadística previamente creada	Se debe desplegar correctamente la estadística seleccionada	OK
Ingresar como invitado al sistema y crear una lista	No se debe permitir el acceso a la creación de listas	OK
Ingresar como invitado al sistema y crear una estadística	No se debe permitir el acceso a la creación de estadísticas	OK
Ingresar como administrador al sistema y eliminarse a sí mismo	No se debe permitir eliminarse así mismo.	OK

Tabla 64. Caso de prueba 12: Login

CASO DE PRUEBA N°13.

Módulo Relacionado		Seguridad	
Prueba Realizada	Resultado Esperado	Estado de la Prueba	
Ingresar a alguna funcionalidad del sistema introduciendo la URL directamente en el navegador	No se debe permitir el acceso. Se debe destinar a la pantalla de inicio	OK	

Autenticarse en el sistema e ingresar a una funcionalidad no permitida mediante el ingreso directo de la URL en el navegador	No se debe permitir el acceso. Se debe desplegar un mensaje indicando que no se puede acceder	OK
--	---	----

Tabla 65. Caso de prueba 13: Seguridad

CASO DE PRUEBA N°14.		
Módulo Relacionado	Listas y Reportes	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Ingresar datos válidos y comprobar la consistencia mediante una lista	La lista debe ser consistente y debe reflejar correctamente los datos ingresados	OK
Ingresar datos válidos y comprobar la consistencia mediante una estadística	La estadística debe ser consistente y debe reflejar correctamente los datos ingresados.	OK
Crear estadística y verificar el gráfico	El gráfico desplegado debe ser consistente con los datos entregados	OK
Verificar el formato del gráfico	El formato del gráfico debe ser de barras	OK

Tabla 66. Caso de prueba 14: Listas y Reportes

CASO DE PRUEBA N° 15.

Caso de Prueba N° 15		
Módulo Relacionado	Notas y Actas	
Prueba Realizada	Resultado Esperado	Estado de la Prueba
Cerrar el acta de una asignatura de un programa en una fecha de dicción específica.	El acta se debe cerrar correctamente	OK
Cerrar el acta de una asignatura de un programa en una fecha de dicción específica y luego agregarle notas a dicha asignatura	No se debe permitir modificar una asignatura una vez cerrada el acta	OK
Ingresar como profesor e ingresar notas en una asignatura que es dictada por otra persona	No se debe permitir modificar las notas de las asignaturas pertenecientes a otros profesores	OK
Ingresar como profesor y cerrar el acta de una asignatura dictada por otra persona	No se debe permitir cerrar actas de asignaturas que no sean propias	OK

Tabla 67. Caso de prueba 15: Notas y Actas

Checklist de Consideraciones mínimas

ITEM

Filtros
Filtros deben buscar por contenido de la palabra (Ejemplo: Filtro “au” encuentra el nombre Claudia)
Los filtros de texto no deberían ser sensibles a los cambios entre mayúsculas y minúsculas.
Cada vez que se cargue una página con filtros, éstos deben desplegarse sin datos (valor nulo).
Visualización de los datos
Los nombres de campos deben estar en español y correctamente escritos
Barra de título debe dejar claro cuál es la página que está siendo visualizada
Consistencia entre el nombre del menú y la página a la cual se ingresa
Los botones de edición deben tener nombres consistentes e iguales en todo el sistema: Ej.: “Agregar”, “Editar” y “Eliminar”
Visualización de información en tablas (completitud de la información a ser vista por el usuario)
Validar que se ocupen los espacios en su totalidad en las grillas (que no ocupen más líneas de las necesarias)
Edición de Datos
Información en combos escrita correctamente
Los combobox deben cargar la información al momento de cargarse la página

Los datos de los combos siempre deben desplegarse ordenados.
Se valida el ingreso de información en los campos que son obligatorios
Al ingresar a una nueva página de edición de datos el cursor se posiciona en el primer campo. Tabuladores deben estar ordenados.
En ningún editor de texto se deben considerar los caracteres en blanco del principio o final de un texto al momento de grabar el campo en la base de datos. Ej.: Si se ingresa el texto: (5 caracteres blanco) + Manzana + (5 caracteres blanco), sólo se debe grabar la descripción “Manzana”, sin considerar los blancos al inicio y al final del texto.
Siempre se debe exigir el ingreso de datos obligatorios, a través de mensajes de error que especifiquen el campo que se debe ingresar.
Generales
Evitar errores de ortografía (incluidos los acentos de palabras con minúsculas) en los labels de todas las páginas.
El botón salir (que cierra la sesión) debe siempre estar disponible
El botón “Volver” siempre debe regresar a la página inmediatamente anterior.
Gráficos
Las leyendas deben ser consistentes y representativas, con respecto a los valores desplegados
Deben tener un título representativo
Nombres de los ejes deben ser consistentes

Se deben indicar las unidades de medida, cuando corresponde

CONCLUSIONES

En primer lugar, es necesario destacar la importancia que tiene la comunicación entre los miembros del equipo de desarrollo. El entendimiento entre las partes permite trabajar en forma confiada unos con otros y ayuda a realizar las tareas de la mejor manera posible, logrando consistencia en el trabajo realizado y optimización de los tiempos y esfuerzos.

Por otra parte, la comunicación con el cliente ayuda en todos los propósitos del proyecto. Permite entender al máximo lo que el cliente desea plantear en los requerimientos, ayuda a encontrar errores anticipados (por ejemplo, estar haciendo algo que no se quiere o que es distinto a lo que se pidió), permitiendo desarrollar de la mejor forma posible la solución que necesita o requiere el cliente.

Una vez que se conocen los requerimientos del cliente, se puede obtener una solución que satisfaga estos, pero, puede que esta solución no sea mejor, ya que es necesario conocer primero como trabaja la organización en la cual el cliente se desempeña, ya sea los flujos e importancia de la información, las tomas de decisiones y todo lo relevante que pueda llevar a tomar una mejor decisión con respecto a la solución que se le quiere dar a los requerimientos.

Otro punto a considerar, es que no sólo es el cliente quien hará uso de la herramienta construida, en muchos casos es posible que el cliente no llegue a utilizarla nunca, por lo que se hace necesario considerar las opiniones y puntos de vista de los principales usuarios que tendrá la aplicación. Con esto, se puede aclarar mucho más que es lo que el sistema debe realizar y ayuda también a reconocer la mejor forma de construirlo.

Referente al desarrollo del proyecto, se puede decir también que la construcción de un sistema de software no es sólo un asunto de programación como se podría pensar en algún momento si no se tuvieran los conocimientos apropiados. Para llevar a cabo un software de calidad, es absolutamente necesario hacer un buen análisis, comprender el problema, conocer al cliente y los usuarios. Para esto se ha pasado por distintas etapas, para poder garantizar un buen

estudio, buena comprensión del problema y por ende la realización de la mejor solución posible.

La organización del equipo de desarrollo es importante, los estándares de codificación permiten trabajar separados pero como si estuviera trabajando una sola persona. Esto da más libertad a los desarrolladores, los cuales pueden saber (o tener noción) de qué manera se está realizando el trabajo y así no marcar diferencia en lo que respecta a la programación. Esto permite una integración más sencilla de los módulos y del sistema en general. Por otra parte, la organización permite ir avanzando de acuerdo a lo establecido a la carta gantt, lo que permite establecer los tiempos de trabajo y el cumplimiento del principal objetivo que es la construcción de este sistema.

REFERENCIAS

Bibliografía

[PRESS] INGENIERIA DEL SOFTWARE: UN ENFOQUE PRÁCTICO (5ª ED.) McGraw Hill, 2002. Roger Pressman.

[LARMAN] UML Y PATRONES (2ª ED.) Prentice-Hall, 2003. Craig Larman.

Internet

[WEB1]

<http://academicos.cualtos.udg.mx/Informatica/Ceneval2003/TutorialdeanalisisyDiseñodesistemas.doc>

[WEB2]

http://www.tenstep.es/01_Publico/0010_ContentoPrincipal/Castellano/0.0.1.2_Direccion_de_Proyectos_vs_Ciclo_de_Vida_del_proyecto.htm

[WEB3] <http://www.monografias.com/trabajos5/desof/desof.shtml>

[WEB4] <http://www.cs.buap.mx/~jpalacio/gimf/esp-form.html>

[WEB5] <http://research.nianet.org/~munoz/espanol.html>

[WEB6] <http://members.fortunecity.es/odi39/modelo.htm>

[WEB7] <http://barranquillo.ucaldas.edu.co>

[WEB8] <http://www.microsoft.com/latam/office/visio/prodinfo/overview.mspx>

[WEB9] http://es.wikipedia.org/wiki/Microsoft_SQL_Server

[WEB10] <http://odfwebdesign.blogspot.com/2006/07/jsp.html>

[WEB11] <http://www.monografias.com/trabajos16/lenguaje-modelado-unificado/lenguaje-modeladounificado.shtml#PROCESO>

[WEB12] http://es.wikipedia.org/wiki/Proceso_Unificado

[WEB13] http://dspace.icesi.edu.co/dspace/bitstream/item/399/1/rcastro_estructura-bas-puds.pdf

[WEB14] <http://www.sparxsystems.com.ar/resources/tutorial/uml-tutorial.html>

[WEB15] http://es.wikipedia.org/wiki/Lenguaje_Unificado_de_Modelado

[WEB16]

http://www.elprofesionaldelainformacion.com/contenidos/1999/septiembre/xml_orientado_a_objetos.html

[WEB17] http://es.wikipedia.org/wiki/Caso_de_uso

ANEXO A: INTERFACES DEL SISTEMA

Acceso al sistema



SAPP
Sistema Administrador de
Programas de Postítulo

Bienvenido

Por Favor Ingrese Rut y Clave de Acceso

Rut: -

Clave:

Ingresar

Login

Mantenedor Alumnos



- [Inicio](#)
- [Mantenedores](#)
- [Alumnos](#)
- [Programa-Asignaturas](#)
- [Profesores](#)
- [Usuarios](#)
- [Tareas](#)
- [Buscadores](#)
- [Reportes](#)
- [Cambiar Clave Acceso](#)
- [Salir](#)

Mantenedor De Alumnos
Ingreso de Datos:

Foto:	
	<input type="button" value="Examinar..."/>
Rut:	<input type="text" value=" - ."/>
Nombres:	<input type="text"/>
Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
Fecha Nacimiento:	<input type="text" value="/ /"/> 1997
Sexo:	<input type="radio"/> Masculino <input type="radio"/> Femenino
País:	Seleccione país
Dirección:	<input type="text"/>
Ciudad:	<input type="text"/>
Teléfono:	<input type="text"/>
Email:	<input type="text"/>
Programa:	Seleccione Programa
<input type="button" value="Grabar"/>	
Ingreso De Títulos:	
Tipo:	Abogacia/Derechos/Leyes
Nombre:	<input type="text"/>
Universidad Obtención:	<input type="text"/>
País:	Argentina
Año:	<input type="text"/> <input type="button" value="Agregar"/>

Alumnos Existentes Para El Programa <input type="text" value="Seleccione Programa"/>			
Rut	Apellidos	Nombres	Opciones

Ingresar Alumnos

Modificar Alumno



- [Inicio](#)
- [Mantenedores](#)
- [Alumno](#)
- [Programa-Asignatura](#)
- [Profesores](#)
- [Usuarios](#)
- [Temas](#)
- [Buscadores](#)
- [Reportes](#)
- [Cambiar Clave Acceso](#)
- [Salir](#)

Mantenedor De Alumnos

Ingreso de Datos:

Foto:	
	
<input type="button" value="Examinar..."/>	
Rut:	14229577 - 8
Nombre:	Fernando
Apellido Paterno:	Muñoz
Apellido Materno:	Yañez
Fecha Nacimiento:	26 / 09 / 1997
Sexo:	<input checked="" type="radio"/> Masculino <input type="radio"/> Femenino
País:	Chile
Dirección:	Bata Carrusel 873
Ciudad:	Viña del Mar
Teléfono:	2544004
Email:	xepishedo@hotmail.com
Programa:	Postítulo en trabajo social

Ingreso De Titular

Tipo:	Abogado/Derecho/leyes
Nombre:	
Universidad Obtención:	
País:	Alemania
Año:	<input type="button" value="Agregar"/>

Alumnos Existentes Para El Programa Postítulo en trabajo social

Rut	Apellidos	Nombre	Opciones
-----	-----------	--------	----------

Mantenedor Profesores



[Inicio](#)

[Mantenedores](#)

[Alumno](#)

[Programa-Asignaturas](#)

[Profesores](#)

[Usuarios](#)

[Tareas](#)

[Bases de Datos](#)

[Reportes](#)

[Cambiar Clave Acceso](#)

[Salir](#)

Mantenedor De Profesores

Ingreso de Datos:

Rut	<input type="text" value=""/> - <input type="text" value=""/>
Nombres	<input type="text"/>
Apellido Paterno	<input type="text"/>
Apellido Materno	<input type="text"/>
Dirección	<input type="text"/>
País	seleccione País <input type="button" value="v"/>
Ciudad	<input type="text"/>
<input type="button" value="Grabar"/>	

Profesores Existentes

Rut	Apellidos	Nombres			
1111111-1	Cofre Zúñiga	Juan Esteban	Modificar	Eliminar	Asignaturas
3333333-5	Perez Flores	José Domingo	Modificar	Eliminar	Asignaturas
4444444-4	Estroff Ortiz	Cristian Javier	Modificar	Eliminar	Asignaturas
54386150-2	Rozas Ruiz	María Josefina	Modificar	Eliminar	Asignaturas
5555555-5	Mosquera Cabalido	Miguel Enrique	Modificar	Eliminar	Asignaturas
6666666-6	Espinoza Vasquez	Oswaldo Anderson	Modificar	Eliminar	Asignaturas
7777777-7	Devidel Murgado	Andrea Piedad	Modificar	Eliminar	Asignaturas

Ingresar Profesor



[Inicio](#)

[Mantenedores](#)

[Alumno](#)

[Programa-Asignaturas](#)

[Profesores](#)

[Usuarios](#)

[Tareas](#)

[Buscadores](#)

[Reportes](#)

[Cambiar Clave Acceso](#)

[Salir](#)

Mantenedor De Profesores
Ingreso de Datos:

Rut	3333333 - 3
Nombre	Jose Domingo
Apellido Paterno	Perez
Apellido Materno	Pezoa
Direccion	Los boscos 111
País	Chile
Ciudad	Valparaiso
<input type="button" value="Grabar"/>	

Profesores Existentes

Rut	Apellidos	Nombre	Modificar	Eliminar	Asignaturas
1111111-1	Cofre Zafra	Juan Esteban	Modificar	Eliminar	Asignaturas
3333333-3	Perez Pezoa	Jose Domingo	Modificar	Eliminar	Asignaturas
4444444-4	Echeverría	Cristian Javier	Modificar	Eliminar	Asignaturas
54386150-2	Rozas Ruiz	Maria Josefina	Modificar	Eliminar	Asignaturas
5555555-5	Monaga Cabaldo	Miguel Enrique	Modificar	Eliminar	Asignaturas
6666666-6	Espinoza Vasquez	Oswaldo Andres	Modificar	Eliminar	Asignaturas
7777777-7	Devicel Murgado	Andrea Fedria	Modificar	Eliminar	Asignaturas

Modificar Profesor

Vincular Asignatura a Profesor

Vincular Profesor a Asignaturas

Profesor:
Rut: 33333333-3
Nombre: Jose Domingo Perez Pezoa

Programa:
Diccionario:

Asignaturas Vinculadas

Código Programa	Programa	Sigla Asignatura	Asignatura	Nº Diccion Programa	Opciones
001	Postítulo en retardo mental	FRM01	Psicopedagogía de las Necesidades	1	
001	Postítulo en retardo mental	FRM02	Educación Especial	1	
001	Postítulo en retardo mental	FRM03	Psicopedagogía del Retardo Mental	1	

Mantenedor Programas





- [Inicio](#)
- [Mantenedores](#)
- [Alumno](#)
- [Programa-Asignaturas](#)
- [Profesores](#)
- [Usuarios](#)
- [Tareas](#)
- [Buscadores](#)
- [Reportes](#)
- [Cambiar Clave Acceso](#)
- [Salir](#)

Mantenedor De Programas/Asignaturas

Ingreso de Datos:

Código Programa:	<input type="text"/>
Nombre:	<input type="text"/>
Descripción:	<input type="text"/>
Facultad:	Seleccione Facultad <input type="button" value="v"/>
Escuela:	<input type="button" value="v"/>
Duración (Hr):	<input type="text"/>
Encargado:	<input type="text"/>
Decreto:	<input type="text"/>
Tipo Período:	Trimestral <input type="button" value="v"/>
Cantidad Períodos:	1 <input type="button" value="v"/>

Programas Existentes

Codigo Programa	Nombre	Tipo Período	Cantidad Períodos			
001	Postítulo en estudio mental	Trimestral	3	Asignatura	Cursos	Canción
002	Postítulo en Dirección de Empresas	Trimestral	4	Asignatura	Módulo	Activa
003	Postítulo en Tratamiento de la Comunicación y del Lenguaje	Trimestral	3	Asignatura	Módulo	Activa
004	Postítulo en Ingeniería de Alimentos Varial	Trimestral	5	Asignatura	Módulo	Activa
005	Postítulo en Composición Musical	Trimestral	3	Asignatura	Cursos	Canción

Ingresar Programas



- [Inicio](#)
- [Mantenedores](#)
- [Alumno](#)
- [Programa-Asignaturas](#)
- [Profesores](#)
- [Usuarios](#)
- [Tareas](#)
- [Buscadores](#)
- [Reportes](#)
- [Cambiar Clave Acceso](#)

[Salir](#)

Mantenedor De Programas/Asignaturas	
Ingreso de Datos:	
Código Programa:	982
Nombre:	Postítulo en Dirección d
Descripción:	
Fecha Creación:	06/17/2007
Facultad:	Facultad de Ciencias Económicas y Administrativas
Escuela:	Escuela de Comercio
Duración (Hrs):	600
Encargado:	Enrique Almose Garcia
Decreto:	OCC001954-4
Tipo Período:	Trimestral
Cantidad Períodos:	4
Vigencia:	En Proceso
Grabar	

Programas Existentes						
Código Programa	Nombre	Tipo Período	Cantidad Períodos			
001	Postítulo en estudio mental	Trimestral	3	Asignaturas	Cursos	Cancels
002	Postítulo en Dirección de Empresas	Trimestral	4	Asignaturas	Modificar	Activa
003	Postítulo en Tratamiento de la Comunicación y del Lenguaje	Trimestral	3	Asignaturas	Modificar	Activa
004	Postítulo en Ingeniería de Alimentos Versal	Trimestral	5	Asignaturas	Modificar	Activa
005	Postítulo en Composición Musical	Trimestral	3	Asignaturas	Cursos	Cancels

Modificar Programa

Asignaturas de un Programa

Vincular Asignaturas a Programas

Código Programa:	002	
Nombre Programa:	Postítulo en Dirección de Empresas	
Sigla	Nombre Asignatura	
Periodo	Numero:1	Ingresar Asignatura a este Periodo
PDE01	Dirección de Empresas	Eliminar
PDE02	Modelos Cuantitativos para la Gestión	Eliminar
PDE03	Comportamiento Humano en Organizaciones	Eliminar
Periodo	Numero:2	Ingresar Asignatura a este Periodo
PDE04	Análisis de Empresas	Eliminar
PDE05	Contabilidad de Gestión	Eliminar
PDE06	Análisis Microeconómico	Eliminar
PDE07	Análisis Macroeconómico	Eliminar
Periodo	Numero:3	Ingresar Asignatura a este Periodo
PDE08	Dirección Financiera	Eliminar
PDE09	Dirección de Recursos Humanos	Eliminar
PDE10	Dirección de Marketing	Eliminar
Periodo	Numero:4	Ingresar Asignatura a este Periodo
PDE11	Dirección Estratégica	Eliminar
PDE12	Estrategias Funcionales	Eliminar
PDE13	Evaluación y Dirección de Proyectos	Eliminar

[Salir](#)



Código Programa:	001
Nombre Programa:	Postítulo en retardo mental
Descripción:	El Diplomado en Psicopedagogía del Retardo Mental, es un programa de postítulo que tiene por objeto perfeccionar y complementar la competencia profesional de quienes se desempeñan o tienen
Escuela:	23
Facultad:	6
Duración (Horas):	315
Encargado:	Maria Luz Merillo Quereu
Decreto:	12.34.6257-17
Tipo Períodos:	Trimestral
Cantidad Períodos:	3
Estado:	Vigente

Mantenimiento de diccionarios

[Crear Una Nueva Dicción](#)

Lista de diccionarios

Selección de dicción

Detalle dicción

Período:	Estado:	Opciones:
----------	---------	-----------

Versiones de un Programa

Mantenedor Usuarios



[Inicio](#)

[Mantenedores](#)

[Alumnos](#)

[Programa Asignaturas](#)

[Profesores](#)

[Usuarios](#)

[Tareas](#)

[Buscadores](#)

[Reportes](#)

[Cambiar Clave Acceso](#)

[Salir](#)

Mantenedor De Usuarios	
Ingreso de Datos:	
Rut	<input type="text"/>
Clave de Acceso:	<input type="text"/>
Reingrese Clave de Acceso:	<input type="text"/>
Nombre:	<input type="text"/>
Apellido Paterno:	<input type="text"/>
Apellido Materno:	<input type="text"/>
Perfil:	Administrador <input type="text"/>
<input type="button" value="Grabar"/>	

Filtro Tipos De Usuario:

Usuarios del Sistema				
Rut	Apellidos	Nombre	Perfil	
42399233-6	Fuella Ortega	Jon Marcos	Administrador	Eliminar Modificar
73734674-5	Cabrera Cifuentes	Cristian	Invitado	Eliminar Modificar

Ingresar Usuario



SAPP
Sistema Administrador de
Programas de Postítulo



[Inicio](#)

[Mantenedores](#)

[Alumnos](#)

[Programa Asignaturas](#)

[Profesores](#)

[Usuarios](#)

[Tareas](#)

[Buscadores](#)

[Reportes](#)

[Cambiar Clave Acceso](#)

[Salir](#)

Mantenedor De Usuarios

Ingreso de Datos:

Rut	42399233 - 6
Clave de Acceso:	*****
Reingrese Clave de Acceso:	*****
Nombre:	Jan Marcos
Apellido Paterno:	Pachla
Apellido Materno:	Ortega
Perfil:	Administrador

Filtro Tipos De Usuario: Todos

Usuarios del Sistema

Rut	Apellidos	Nombre	Perfil	
42399233-6	Pachla Ortega	Jan Marcos	Administrador	Permisos Modificar Eliminar
73734674-5	Cabrera Cifuentes	Christian	Invitado	Modificar Eliminar

Modificar Usuario



SAPP
Sistema Administrador de
Programas de Postítulo

Permisos de Usuario

Ficha Usuario

Rut	42399233-6
Nombre:	Jan Marcos
Apellidos:	Pachla Ortega
Perfil:	Usuario administrador

Permisos

Programas	<input checked="" type="checkbox"/> Ingreso	<input type="checkbox"/> Modificación	<input type="checkbox"/> Eliminación
Asignaturas	<input checked="" type="checkbox"/> Ingreso	<input type="checkbox"/> Eliminación	
Alumnos	<input checked="" type="checkbox"/> Ingreso	<input type="checkbox"/> Modificación	<input type="checkbox"/> Eliminación
Profesores	<input checked="" type="checkbox"/> Ingreso	<input type="checkbox"/> Modificación	<input type="checkbox"/> Eliminación
Usuarios	<input checked="" type="checkbox"/> Ingreso	<input type="checkbox"/> Modificación	<input type="checkbox"/> Eliminación <input type="checkbox"/> Permisos

Asignar Permisos a Usuario

Buscadores



The screenshot shows the SAPP (Sistema Administrador de Programas de Postítulo) interface. On the left is a navigation menu with links: Inicio, Mantenedores, Alumno, Programa-Asignatura, Profesores, Usuarios, Tareas, Buscadores, Reportes, and Cambiar Clave Acceso. At the bottom of the menu is a 'Salir' button. The main content area features the SAPP logo and a search section titled 'Buscadores'. Under 'Buscar:', there are radio buttons for 'Alumno' (selected), 'Profesor', 'Usuario', 'Programa', and 'Asignatura'. Below this is a sub-section 'Buscar alumno por:' with radio buttons for 'Rut', 'Apellidos Nombres' (selected), and 'Programas - Apellidos Nombres'. The 'Ingreso Apellidos y Nombres' form has two input fields: 'Apellidos' and 'Nombres', both highlighted in yellow, and a 'Buscar' button below them.

Buscar Alumnos por Nombre

Buscar Alumnos por Rut



This screenshot is identical to the one above, showing the SAPP search interface. The difference is in the search criteria: under 'Buscar alumno por:', the 'Rut' radio button is selected instead of 'Apellidos Nombres'. Consequently, the 'Ingreso Rut' form has a single input field for 'Rut' highlighted in yellow, with a hyphen and a second empty space for the second part of the number, and a 'Buscar' button below it.

The screenshot shows the SAPP web interface. On the left is a navigation menu with links: Inicio, Mantenedores, Alumno, Programa-Asignatura, Profesores, Usuarios, Tareas, Buscadores, Reportes, Cambiar Clave Acceso, and Salir. The main content area is titled "Buscadores" and contains a search form. The "Buscar:" section has radio buttons for "Alumno", "Profesor", "Usuario", "Programa", and "Asignatura", with "Asignatura" selected. Below this, the "Buscar asignatura por:" section has radio buttons for "Sigla", "Nombre Asignatura", and "Programa", with "Sigla" selected. The input field is labeled "Ingrese Sigla Asignatura" and contains the text "Sigla". A "Buscar" button is at the bottom of the form.

Buscar Asignatura por Sigla

Buscar Profesor por Programa

The screenshot shows the SAPP web interface. On the left is a navigation menu with links: Inicio, Mantenedores, Alumno, Programa-Asignatura, Profesores, Usuarios, Tareas, Buscadores, Reportes, Cambiar Clave Acceso, and Salir. The main content area is titled "Buscadores" and contains a search form. The "Buscar:" section has radio buttons for "Alumno", "Profesor", "Usuario", "Programa", and "Asignatura", with "Profesor" selected. Below this, the "Buscar profesor por:" section has radio buttons for "Rut", "Apellidos Nombres", and "Programas - Apellidos Nombres", with "Programas - Apellidos Nombres" selected. The input field is labeled "Ingrese Apellidos y Nombres:" and contains a dropdown menu for "Seleccione Programa" and two text input fields for "Apellidos" and "Nombres". A "Buscar" button is at the bottom of the form.

The screenshot shows the SAPP web interface. At the top right is the SAPP logo with the text "Sistema Administrador de Programas de Postítulo". On the left is a navigation menu with links: Inicio, Mantenedores, Alumno, Programa-Asignatura, Profesores, Usuarios, Tareas, Buscadores, Reportes, and Cambiar Clave Acceso. Below the menu is a "Salir" button. The main content area features a "Buscadores" section with a "Buscar:" dropdown menu containing options: Alumno, Profesor, Usuario, Programa (selected), and Asignatura. Below this is a "Buscar programa por:" section with radio buttons for Código Programa, Nombre Programa, Facultad, and Facultad - Escuela (selected). A second form titled "Seleccione Facultad y escuela" contains dropdown menus for Facultad (Todas) and Escuela (Todos), a text input for Nombre, and a "Buscar" button.

Buscar Programa por Escuela

Buscar Programa por Nombre

This screenshot is similar to the first one, showing the SAPP web interface. The "Buscar:" dropdown menu is set to "Programa". In the "Buscar programa por:" section, the "Nombre Programa" radio button is selected. The "Seleccione Facultad y escuela" form is now titled "Ingreso Nombre Programa" and only contains a text input field for "Nombre" and a "Buscar" button.

Buscar Usuario por Perfil

Listas

Crear Lista de Alumnos

SAPP
Sistema Administrador de
Programas de Postítulo



[Inicio](#)
[Mantenedores](#)
[Alumno](#)
[Programa-Asignatura](#)
[Profesores](#)
[Usuarios](#)
[Tareas](#)
[Buscadores](#)
[Reportes](#)
[Cambiar Clave Acceso](#)
[Salir](#)

Tipos de Informes
 Listas Estadísticas

Listas
Tipo de Listas Pública Privada

Listas de:
 Alumno Profesor Programa Asignatura

Buscar programa por:
 Programas en proceso, vigentes, no vigentes Por tipo de periodo Cantidad de Periodos Facultad/Escuela

Seleccione estado de los programas
Estados: En Proceso Vigente No Vigente

Crear Lista de Programas

Lista de Alumnos por Programa

Listado de alumnos por Programa

Código Programa: 001

Nombre programa: Postítulo en retardo mental

Rut	Apellidos Nombres
14229577-6	Arce Astorga María Bernarda
47853189-3	Barahona Cárcamo Isaias Ignacio
48607730-1	Fernandez Fernandez Daniel Enrique
50507414-9	Medina Sanhueza Carolina Alejandra
7692676-k	Montenegro Aranguiz Cristina Andrea
98304861-7	Muñoz Yañez Fernando

Estadísticas

SAPP
Sistema Administrador de
Programas de Postítulo

Tipos de Informes
 Listas Estadísticas

Estadísticas
 Tipo de Estadísticas Pública Privada

Estadísticas de:
 Alumno Profesor Programa Asignatura

Sexo: Masculino Femenino Ambos

País de procedencia: Todos

Ciudad: Todos

Que hayan: Cursado Aprobado Reprobado Egresados Indiferente

Hayan cursado programa: Seleccione Programa

Asignatura: Todos

En Facultad: Todos

En Escuela: Todos

Posean Teléfono: Si No Indiferente

Posean Email: Si No Indiferente

Entre los años: 2008

Generar

Crear Estadísticas Alumno

SAPP
Sistema Administrador de
Programas de Postítulo

Tipos de Informes
 Listas Estadísticas

Estadísticas
 Tipo de Estadísticas Pública Privada

Estadísticas de:
 Alumno Profesor Programa Asignatura

Estado: Creados En Proceso Activos Cancelados Indiferente

Cantidad Periodos: Entre | y |
 Trimestres Semestres Años Indiferente

Cantidad duración (hrs): Entre | y |

En Facultad: Todos

En Escuela: Todos

Cantidad de alumnos aprobados/reprobados: Entre | Y |
 Cursados Aprobados Reprobados Indiferente

Entre los años: 2008

Generar

Crear Estadísticas

Programa Estadística de Programas



Estadística Programas

Periodos entre: - Y - Tipo Periodo: Indiferente
Facultad: Todas Estado: Indiferente
Escuela: Todas Duración (hrs) entre: - Y -
Cantidad de alumnos: Indiferente entre - Y -

Año	Cantidad
2006	0
2007	5

