



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE Guayaquil

Facultad de: Ingenierías

Carrera: Ingeniería en Sistemas mención: informática para la gestión

Tesis previa a la obtención del título de: ingeniero de sistemas

Título: ESTUDIO DE FACTIBILIDAD, DESARROLLO E IMPLEMENTACION DE UN SISTEMA INTEGRADO DE GESTION EDUCATIVA APLICABLE A CUALQUIER NIVEL DE EDUCACION (SIGA).
MODULOS DE GESTIÓN DE CONVIVENCIA Y RÉGIMEN ACADÉMICO Y
MODULO EVALUACIÓN Y GESTION DE CALIFICACIONES

Nombre del autor/a (s): Eliana Escobar

Dalton Hernández

Nombre del director: Ing. Nelson Mora

Ciudad y año: Guayaquil, 2012

Declaratoria de Responsabilidad

Las investigaciones, definiciones expuestas, análisis realizados así como las conclusiones del proyecto en mención, han sido plasmadas con plena y exclusiva responsabilidad de los autores, y el desarrollo intelectual le pertenece a la Universidad Politécnica Salesiana.

Guayaquil, 30 Julio del 2012

f.....

Eliana Escobar Estacio

f

Dalton Hernández García

Dedicatoria

A Dios porque ha estado conmigo a cada paso que doy, dándome fortaleza para continuar, brindándome la oportunidad y la dicha de la vida y los medios necesarios para continuar.

A mis padres, por darme una carrera para mi futuro, porque creyeron en mí, por sus dignos ejemplos de superación y entrega, formando en mí el anhelo de triunfo.

A mi segunda madre Inés Estacio, por la paciencia para impulsarme siempre en los momentos más difíciles de mi carrera, por sus consejos y regaños, ya que sin ellos no hubiera podido llegar.

A mis hermanos y primos, por su apoyo constante, por su confianza y porque no dudaron que llegaría al final de la meta.

A mis amigos que sin esperar nada a cambio aportaron su granito de arena para alcanzar esta meta.

A mis profesores por su ejemplo de profesionalismo que nunca olvidare.

Eliana

Dedicatoria

A mis apreciados y amados Padres, quienes con su esfuerzo y dedicación supieron y pudieron brindarme el apoyo necesario para llegar a donde estoy, quienes me inculcaron importantes valores y un profundo respeto e igualdad para quienes me rodean, me enseñaron lo importante que es trabajar por lo que se quiere, pues sencillamente la satisfacción del logro es proporcional al esfuerzo.

A mis catedráticos, quienes contribuyeron de una forma indispensable con su valioso conocimiento, y claro con los consejos académicos-profesionales que aportaban y aportan sed de éxito y cumplimiento de nuevos retos, es claro entender que no hay imposibles, no mientras se quiera, no mientras logremos nuestro propósito, no mientras nos apoyemos como lo hemos hecho hasta ahora, no mientras no dejemos de ser quienes somos.

A mis hermanos y familia quienes con su voz de aliento y apoyo me brindaron fuerza, gracias a su convicción y por creer en mí, porque a pesar de la distancia siempre están cerca.

A todos mis compañeros, primero por brindarme su amistad, por transmitir su alegría, por compartir conmigo esta carrera, espero seguir compartiendo con ustedes muchos retos y logros.

Por último pero no menos importante, quiero agradecer a mi buen amigo y compañero, el Ing. Néstor Mestanza quien considero pieza clave del éxito de mi proyecto, ya que gracias al asesoramiento y apoyo incondicional, podemos contar con un proyecto de calidad.

Dalton

Agradecimiento

Agradecemos a nuestros padres y familia por su apoyo incondicional, por creer en nosotros, por comprendernos y por su incomparable amor, queremos manifestar que ha sido, es y será recíproco.

A la Universidad Politécnica Salesiana sede Guayaquil, prestigiosa institución conocida por formar profesionales con valor humanitario de excelencia, estamos muy satisfechos con el aporte brindado por la institución para con nosotros y es un gusto poder ser reconocido como politécnico salesiano.

De manera especial al Ing. Nelson Mora, por el importante asesoramiento y colaboración en nuestro proyecto, por su gran predisposición y cooperación, pues siempre ha estado disponible ante cualquier inquietud o inconveniente por mínimo que este fuera.

Eliana Escobar Estacio
Dalton Hernández García

Introducción

Hoy en día son numerosas y casi la mayoría de las instituciones educativas sean estas del sector público o privado, cuentan con herramientas para el registro de datos del cliente, es decir, para estudiantes y sus representantes, sean estos de educación primaria, secundaria y tercer nivel; Sin embargo en la mayoría de casos los sistemas de información están “Fragmentados”, sistemas diseñados por diferentes proveedores, sin seguir un estándar, procedimiento o políticas de seguridad que garanticen y reflejen la completa interoperabilidad de los módulos, disminuyendo la productividad de la compañía.

El hecho de contar con sistemas fragmentados, es que al no contemplar el crecimiento de la institución con la adecuada proyección, se puede tener problemas de operatividad de gran magnitud, en una entidad educativa normalmente se puede observar departamentos como: Matriculación, Colecturía, Médico, Calificación, Dobe, (Psicología), etc. No necesariamente cada departamento maneja una herramienta que permita salvaguardar información de forma consistente y si tienen algún tipo de herramienta lo más probable es que no esté integrada con las demás y estén basadas en un proceso particular, excluyendo cierta información, con este resultado lo que logramos tener es información fragmentada, aislada y lo más probable inconsistente.

El proyecto presentado fue planteado basado en un análisis justamente para satisfacer esta necesidad y poder brindar solución eficiente e integra ante los requerimientos de los usuarios, como referencia hemos tomado un modelo en particular como lo es un sistema ERP (Enterprise Resource Planning). Los beneficios obtenidos al contar con una herramienta que cuente con un alcance similar son numerosas pero sobre todo eficientes, por ejemplo:

- ✓ Base de Datos Centralizada.
- ✓ No importa la cantidad de módulos desarrollados, todos son y representan uno solo.

La tendencia en soluciones informáticas está basada en brindar soluciones sectoriales, personalizadas para satisfacer necesidades en particular.

Se garantiza la integración de todos los módulos consolidando y, presentando una sola solución.

Optimización de Recursos y Procesos empresariales.

Esta solución será desarrollada con el fin de consolidar la gestión académica y poder posicionarla en el mercado como una herramienta eficiente que garantiza el ahorro de costo de operatividad, garantizar consistencia en los datos de forma inmediata, y poder acceder de forma sencilla mediante un explorador web.

El proyecto está orientado hacia el sector educativo y podrá ser acoplado, personalizado, según las necesidades del cliente.

ÍNDICE INICIAL

Tema	Pag.
Declaratoria de Responsabilidad	II
Dedicatoria	III
Dedicatoria	IV
Agradecimiento	V
Introducción	VI

INDICE DE CONTENIDO

CAPÍTULO 1	16
Diseño de la Investigación	16
1.1 Antecedentes de la Investigación	16
1.2 Problema de la investigación	17
1.2.1 Planteamiento del problema	17
1.2.2 Formulación del problema de investigación	18
1.2.3 Sistematización del problema de investigación	18
1.3 Objetivo de Investigación	19
1.3.1 Objetivo general	19
1.3.2 Objetivos específicos	19
1.4 Justificación de la investigación	20
1.5 Marco de referencia de la investigación	21
1.5.1 Marco teórico	21
1.5.1.1 Patrón de Diseño MVC	21
1.5.1.2 Hibernate	22
1.5.1.3 Apache Struts	28
1.5.1.4 Extjs	28
1.5.1.5 Netbeans	30
1.5.1.6 ¿Por qué utilizar el lenguaje JSP en vez de ASP?	33
1.5.1.7 Ajax	34
1.5.1.8 Postgres	42
1.5.1.9 Funciones	44
1.5.2.1 Modelo relacional de datos	45
1.5.2.2 Esquema	46
1.5.2.3 Instancias	46

1.5.2.4 Base de datos relacional	46
1.5.2.5 La Entrevista	48
1.5.2.6 El Entrevistador	48
1.5.2.7 El entrevistado	49
1.6 Marco conceptual	55
1.7 Formulación de la Hipótesis y Variables	61
1.7.1 Hipótesis general	61
1.7.2 Hipótesis particulares	61
1.7.3 Matriz Causa – Efecto	62
1.7.4 Variables	64
1.8 Aspectos Metodológicos de la Investigación	65
CAPÍTULO 2	69
2. ANÁLISIS, PRESENTACIÓN DE RESULTADOS Y DIAGNÓSTICOS	69
2.1 Análisis de la situación actual	69
2.1.1 Biografía San Antonio María Claret	69
2.1.2 Reseña Histórica de la unidad educativa claretiana	70
2.1.3 Antecedentes	71
2.1.4 Misión	72
2.1.5 Visión	73
2.1.6 Objetivos de la Unidad Educativa Claretiana	74
2.1.7 Justificación	75
2.2 Estructura Organizacional	78
2.2.1 Organigrama	78
2.2.2 Diagnóstico	79
2.2.3 Análisis Costo/ Calidad en la Implementación	82
2.2.4 Estudio Tecnológico	84
CAPÍTULO 3	86
3. PROPUESTA DE CREACIÓN	86
3.1 Modelo de Procesos	86
3.1.1 Procedimiento de listado, ingreso, y selección de dimensiones	86
3.1.2 Registros de Atrasos	87
3.1.3 Registro de Faltas	88
3.1.4 Control de Justificación de Faltas	89
3.1.5 Registro de Indisciplina de Estudiante	91

3.1.6 Ingreso de Pruebas de Admisión	91
3.1.7 Ingreso Categoría Calificación	96
3.1.8 Modificación de Nota Parcial	98
3.1.9 Registro de Materias a Evaluar en Examen de Grado	99
3.2.1 Ingreso de Paracadémico	101
3.3 Diagrama de Clase	103
3.3.1 Diagrama de Calificación	103
3.3.2 Diagrama de Supervisión y Tutoría	104
3.3.3 Diagrama de Promedios de Grado	105
3.4 Diagrama de Casos de Uso	106
3.4.1 Calificación Parcial	106
3.4.2 Control de Asistencia	108
3.4.3 Gestión de Conflicto de Convivencia	110
3.4.4 Tutoría	112
3.5 Diagrama de Secuencia	114
3.5.1 Diagrama de calificación de un determinado periodo y parcial	114
3.5.2 Diagrama de generación de promedio de grado	115
3.5.3 Diagrama de tutoría	116
3.6 Diagrama de Estado	117
3.6.1 Diagrama de calificación parcial de un determinado período	117
3.6.2 Diagrama de promedio para obtener nota de grado	118
3.6.3 Diagrama de Tutoría	119
3.7 Modelo de Procesos	120
3.7.1 Modelo de Calificación Parcial	120
3.7.2 Modelo de Tutoría	121
3.7.3 Modelo de Promedio de Nota de Grado	122
3.8 Modelo Entidad Relación	123
3.8.1 Diagrama de Calificación	123
3.8.2 Diagrama de Promedio de Grado	124
3.8.3 Diagrama de Supervisión y Tutoría	125
3.9 Diccionario de Datos	126
3.10 Funciones y Trigger	137
3.10.1 Función “publico.insertarpromedioexamengrado()”	137
3.10.2 Job	139

3.11 Diagrama Jerárquico de la aplicación.	147
3.12 Manual de Usuario	152
3.12.1 Pantalla de Inicio	152
3.12.2 Mantenimientos	153
3.12.3 Ingresos	153
3.12.4 Mantenimiento Calificaciones	154
3.13 Conclusiones	159
3.14 Recomendaciones	160
3.15 Bibliografía	161
3.16.1 Encuesta para la estudio de factibilidad de un sistema integrado de gestión educativa.	164
3.16.2 Instalación del motor de base de datos postgre-sql en windows	167
3.16.3 Mapeo de las tablas de la base de datos	171

INDICE DE TABLAS

Tabla	Pag.
Tabla #1: Archivo de configuración Hibernate.....	27
Tabla #2: Cabecera necesarias para integrar las interfaces de ExtJS.....	30
Tabla #3: Código de ejemplo de ASP	33
Tabla #4: Ejemplo de Código jsp y ajax de calificación por periodo.	41
Tabla #5: Cuadro diferencial de procesamiento de gestión de calificación.....	54
Tabla #6: Matriz causa efecto	63
Tabla #7: Organigrama de la institución claretiana.	78
Tabla #7: Cuadro descriptivo del estudio realizado en las instituciones.	83
Tabla #8: Cuadro descriptivo del caso de uso calificación parcial.	107
Tabla #9: Cuadro descriptivo del caso de uso de control de asistencia.	109
Tabla #10: Cuadro descriptivo del caso de uso de gestión de convivencia.	111
Tabla #11: Cuadro descriptivo del caso de uso de tutoría.	113
Tabla #12: Diccionario de datos de la tabla ‘aca_atraso’	126
Tabla #13: Diccionario de datos de la tabla ‘aca_calificación_categoria’.....	127
Tabla #14: Diccionario de datos de la tabla ‘aca_calificación_parcial’	128
Tabla #15: Diccionario de datos de la tabla ‘aca_categoria_calificacion’.....	128
Tabla #16: Diccionario de datos de la tabla ‘aca_control_faltas’	129
Tabla #17: Diccionario de datos de la tabla ‘aca_ficha_conducta’.....	130
Tabla #18: Diccionario de datos de la tabla ‘aca_materia_examen_grado’	130
Tabla #19: Diccionario de datos de la tabla ‘aca_paracademico’.....	131
Tabla #20: Diccionario de datos de la tabla ‘aca_promedio_anio_lectivo’.....	131
Tabla #21: Diccionario de datos de la tabla ‘aca_promedio_grado’.....	132
Tabla #22: Diccionario de datos de la tabla ‘aca_promedio_periodo’	133
Tabla #23: Diccionario de datos de la tabla ‘aca_rango_supletorio’	133
Tabla #24: Diccionario de datos de la tabla ‘aca_supletorio_registro’.....	134
Tabla #25: Diccionario de datos de la tabla ‘aca_tipo_paracademico’.....	134
Tabla #26: Diccionario de datos de la tabla ‘aca_alumnos_tutores’.....	135
Tabla #27: Diccionario de datos de la tabla ‘aca_libretas’	135
Tabla #28: Diccionario de datos de la tabla ‘aca_calificación_examen_grado’.....	136
Tabla #29: Ejemplo de función para insertar el promedio examen de grado.....	138
Tabla #30: Muestra el nivel de mantenimiento académico.....	148

Tabla #31: Muestra el nivel de mantenimiento de categorías y calificaciones.....	149
Tabla #32: Muestra el nivel de paracadémico y rango supletorio.	149
Tabla #33: Muestra opciones para el mantenimiento de disciplina y atrasos.	150
Tabla #34: Muestra opciones de mantenimiento faltas y ficha conducta.	150
Tabla #35: Muestra el nivel para el proceso académico y tutoría.....	151
Tabla #36: Opciones para la opción de asignaturas examen de grado.....	151

INDICE FIGURAS

Figura	Pag.
Figura #1: Esquema descriptivo de funcionamiento de Ajax	36
Figura #2: Tablas correspondiente a la base de datos siga, módulo calificación.....	47
Figura #3: Decisión de implementar un sistema ERP.....	81
Figura #4: Costo Calidad de Implementación.....	81
Figura #5: Representación de objetivos que buscan las instituciones.....	82
Figuras #6: Diagrama de Procedimiento de Ingreso, listado y selección de dimensiones.....	86
Figura #7: Diagrama de registro de atrasos por alumno	87
Figura #8: Diagrama de registro de faltas por alumno.....	88
Figura #9: Diagrama de justificación de faltas por alumno	89
Figura #10: Diagrama de indisciplina de faltas por alumno, materia y curso	90
Figura #10: Diagrama de indisciplina de faltas por alumno, materia y curso	91
Figura #11: Diagrama de ingreso de pruebas de admisión, materia, curso y especialidad	92
Figura #12: Actualización, Ingreso, modificación de Preguntas, alternativas y Respuestas de Pruebas de Admisión	93
Figura #13: Programación de Pruebas de Admisión.....	94
Figura #14: Programación de fechas de activación y desactivación de pruebas de admisión	95
Figura #15: Diagrama de ingreso de una categoría calificación.	96
Figura #16: Registro de calificación por categoría.	97
Figura #17: Modificación de una nota parcial realizada por secretaría.	98
Figura #18: Registro y selección de las materias a evaluar para examen de grado por especialidad	99
Figura #19: Registro de calificación de examen de grado.	100
Figura #20: Registrar el tipo de paracadémico y el curso a utilizar.....	101
Figura #21: Registrar la calificación de paracadémico.	102
Figura #22: Diagrama de clase correspondiente a calificación categoría, calificación parcial, promedios, supletorio.	103
Figura #23: Diagrama de clase de tutoría, atraso, faltas y conducta.....	104
Figura #24: Diagrama de clase de tutoría, atraso, faltas y conducta.....	105
Figura # 25: Diagrama de caso de uso de calificación parcial.	106

Figura #26: Diagrama de caso de uso de control de asistencia.....	108
Figura # 27: Diagrama de caso de uso de gestión de convivencia.....	110
Figura #28: Diagrama de caso de uso para tutoría.....	112
Figura #29: Diagrama de secuencia para calificación docente.....	114
Figura #30: Diagrama de secuencia para la generación de promedio.....	115
Figura #31: Diagrama de secuencia para tutoría.....	116
Figura #32: Diagrama de estado para la calificación parcial de un período.....	117
Figura #33: Diagrama de estado para la generación de promedio de grado.....	118
Figura #34: Diagrama de estado para tutoría.....	119
Figura #35: Modelo de Proceso para Calificación Parcial.....	120
Figura #36: Modelo de Tutoría para Alumnos.....	121
Figura #37: Modelo de Promedio de Nota de Grado.....	122
Figura #38: Modelo de entidad relación del proceso de calificación.....	123
Figura #39: Diagrama entidad relación de promedio de grado.....	124
Figura #40: Diagrama entidad relación de supervisión y tutoría.....	125
Figura #41: Archivos necesarios para creación y configuración de un job.....	139
Figura #42: Ejecución de consulta para la base de datos.....	140
Figura #43: Muestra la pantalla inicial cuando se abre PgAdmin III.....	141
Figura #44: Ventana que muestra los parámetros de configuración de un job.....	142
Figura #45: Muestra los parámetros de fecha de un job.....	143
Figura #46: Muestra la ventana para creación de un nuevo paso.....	145
Figura #47: Se especifica el código a ejecutar el nuevo paso.....	146
Figura #48: Muestra la pantalla inicial del sistema y autenticación.....	152
Figura #49: Muestra opciones de mantenimiento para todos los procesos.....	153
Figura #50: Muestra opciones para ingreso de una categoría de calificación.....	153
Figura #51: Muestra opciones de mantenimiento para calificación docente.....	154
Figura #52: Muestra la lista de alumnos disponibles a calificar.....	155
Figura #53: Imagen que muestra calificaciones correspondientes a una materia...	156
Figura #54: Imagen que muestra el mantenimiento de calificación por secretaría..	156
Figura #55: Imagen de las opciones para secretaría para modificación o consulta.	157
Figura #56: Muestra las calificaciones por curso y por periodo o parcial.....	158
Figura #57: Libreta de calificaciones correspondiente a un alumno.....	158
Figura #58: Instalador de postgresql.....	167
Figura #58: Primer ventana que durante la instalación de PostgreSQL.....	167

Figura #59: Imagen que muestra la ruta donde se instalará PostgreSQL.	168
Figura #60: Nos solicita el usuario y contraseña por defecto.	169
Figura #61: Nos muestra la ventana con el puerto predeterminado de instalación..	169
Figura #62: Muestra el progreso y la finalización de instalación de Postgres.	170
Figura #63: Muestra la opción de conexión de la base de datos con la aplicación..	171
Figura #64: Imagen que muestra la conexión realizada con la base de datos.....	172
Figura #65: Imagen que muestra el primer paso para la creación de mapeo.	172
Figura #66: Creando mapeo parte 1.	173
Figura #67: Creando mapeo parte 2.	174
Figura #68: Creando mapeo parte 3.	174
Figura #69: Creando mapeo parte 4.	175
Figura #70: Creando mapeo parte 5.	176
Figura # 71: Creando mapeo parte 6.	176
Figura #72: Creando mapeo parte 7.	177

CAPÍTULO 1

Diseño de la Investigación

1.1 Antecedentes de la Investigación

La unidad educativa Claretiana, institución Educativa Particular Mixta, se destaca por ser formadora de buenos estudiantes, dedicados y comprometidos con su carrera y conscientes de la realidad social.

La institución cuenta con distintas herramientas básicas para registrar los procesos de matriculación, calificación, actualmente no posee una herramienta automatizada para el seguimiento, control, registro y direccionamiento adecuado en cuanto a calificación. Esta situación es común en la mayoría de las instituciones educativas reflejando la subutilización del recurso tecnológico forzando la carga operativa, ya que el registro de datos para los procesos que no están automatizados tendría que ser manual, así podemos observar en algunos casos el uso frecuente de plantillas preimpresas. Hay que tener en cuenta que el trabajo puede variar dependiendo de la cantidad de alumnos con que cuente la institución o del tipo de informe que se requiera.

Las herramientas tecnológicas están diseñadas para automatizar procesos, y un plus para el proyecto lo justifica la integración de los módulos existentes, consolidando información y logrando establecer un estándar de trabajo en un ambiente Web, facilitando el acceso desde cualquier estación de trabajo.

1.2 Problema de la investigación

1.2.1 Planteamiento del problema

El proceso actual asignado a personal de la institución educativa Claretiana, demanda bastante tiempo y representa una tarea tediosa que normalmente demanda una gran carga operativa para el personal de la empresa e implica la recolección de datos en métodos comunes que no garantizan la integridad de los datos.

Para la institución en mención los procesos como secretaria, compras, infraestructura, contabilidad, biblioteca no están automatizados, mientras que las herramientas que posee para calificación y matriculación se mantienen aislados, lo que significa que para poder realizar un informe que involucre algún tipo de relación entre los procesos mencionados y algún otro proceso tendría que depurar la información de forma manual, luego de ello otro inconveniente sería presentar el informe ya que este también tendría que realizarse de forma manual, un detalle importante que hay que recalcar en esta situación es que si bien es cierto se puede contar con más talento humano para la realización de las tareas al menos no nos garantiza que toda la información sea consistente.

Para el problema anterior una solución para optimizar recursos y aumentar productividad sería automatizar e integrar los departamentos para que puedan realizar su trabajo de forma eficiente, evitando que la base de datos utilizada trabaje de forma descentralizada, por ende ahorraría la inversión en obtener algún tipo de herramienta de un tercero para la generación de informes, en consecuencia demandaría menor participación y esfuerzo de recurso humano, se evita contratar algunos proveedores por la diversidad de soluciones necesarias, se ahorra en gran magnitud la inversión en mantenimiento y en capacitación del personal, es decir, usuarios de los módulos y soporte.

1.2.2 Formulación del problema de investigación

Determinar cómo afecta la falta de una herramienta estándar e integrada al resto de procesos, con mayor disponibilidad para el manejo de datos y el registro de los mismos en cuanto al proceso de calificación, asistencia, y gestión de convivencia.

1.2.3 Sistematización del problema de investigación

¿En qué incide la forma de acceso en cuanto al proceso de registro, modificación y consulta de calificación?

¿En qué medida afecta que la herramienta utilizada para los procesos de calificación funcione de forma descentralizada?

¿Qué incidencia tiene en la duración el llevar el procesamiento de calificaciones, gestión de convivencia y asistencia de la forma actual?

El registro de calificación en el sistema a través de una sola persona incide en la cantidad de tiempo necesario para realizar el registro.

¿Qué impacto ecológico tiene el uso de recursos como papel a gran volumen para el registro de calificación, convivencia y asistencia?

¿Qué beneficio existe al almacenar la información correspondiente de registro de calificaciones en un sistema y no en un repositorio físico?

¿Qué ventajas existe en tener la información de forma digital a tenerla de forma física?

¿Cuál es el costo de llevar a cabo el proceso de respaldo manual versus el respaldo automático?

1.3 Objetivo de Investigación

1.3.1 Objetivo general

Identificar los problemas y posibles causas que originan que los procesos de calificación, gestión de convivencia y asistencia no se desarrollen de una forma óptima.

1.3.2 Objetivos específicos

Identificar en que magnitud afecta la forma de acceso a los procesos de registro, modificación y consulta de calificación.

Determinar las desventajas que se tienen como resultado cuando se utiliza una herramienta descentralizada.

Determinar la incidencia que tiene en la duración el llevar el procesamiento de información de la manera actual.

Identificar las causas que justifiquen que la distribución de trabajo brinda mayor productividad.

Determinar el impacto negativo que afecta ecológicamente el uso de papel a gran volumen.

Determinar si es más seguro la metodología de almacenamiento en una base de datos que en archivadores.

Validar las ventajas y beneficios de tener la información digitalizada versus física.

Determinar que es más costoso en cuanto a realizar respaldos de la información, en un sistema manual y otro automatizado.

1.4 Justificación de la investigación

El problema que se detecta al momento de trabajar con soluciones informáticas independientes o aisladas, es que genera la demanda de tiempo y talento humano para la elaboración de informes o depuración de datos, hay que tener en cuenta que el esfuerzo requerido depende del tiempo que se tiene para finalizar y la cantidad de información que se tenga que utilizar para elaborar el informe, otro punto es que exista la garantía que la información presentada sea 100% real y consistente. Adicional a esto está la parte financiera ya que se invierte en soluciones aisladas sin garantizar el retorno de la inversión, para el tema de mantenimiento se debe contar con los distintos proveedores quienes son los responsables de realizar modificaciones, actualizaciones de las aplicaciones, si los vemos de forma general hemos detectado la falta de control y centralización para los planes estratégicos de la organización.

La investigación que se está realizando tiene como objetivo, diseñar, desarrollar e implementar una solución que se encuentre alineada con el plan estratégico de la institución académica y permita tener el control de forma centralizada de la información con alta disponibilidad y que permita el retorno de la inversión, además de tener pleno control sobre el proceso de calificaciones, asistencia y supervisión docente y que se integre con el resto de procesos de cualquier institución educativa acorde a necesidades específicas.

El proyecto se plantea posicionarlo en el mercado, teniendo como target las instituciones educativas de cualquier nivel, la idea es que cuenten con los beneficios que ofrece la herramienta, incrementar la productividad y facilitar el acceso desde cualquier lugar de la empresa ya que el ingreso web ofrece mayor cobertura sin necesidad de instalar alguna aplicación que permita el acceso o conectividad.

El proyecto estará orientado netamente a las instituciones educativas por su flexibilidad y escalabilidad puede ser implementado según la necesidad de las mismas.

1.5 Marco de referencia de la investigación

1.5.1 Marco teórico

1.5.1.1 Patrón de Diseño MVC

¿Qué es MVC?

El patrón de arquitectura de Software Modelo Vista Controlador, tiene como función separar los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes diferentes. El modelo es el gestor de la base de datos y la Lógica del Negocio, el controlador es el encargado de recibir los datos de entrada desde la vista.

Modelo: Esta es la representación específica de la información con la cual el sistema opera, el modelo se limita a lo relativo de la vista y su controlador facilitando las presentaciones visuales complejas. El sistema también puede operar con más datos que no tengan relación con la presentación, haciendo uso integrado de otras lógicas de negocio y de datos afines con el sistema modelado.

Vista: Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Este responde a eventos, usualmente acciones del usuario, e invoca peticiones al modelo y, probablemente, a la vista.

Muchos de los sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos: en líneas generales del MVC corresponde al modelo. La unión entre capa de presentación y capa de negocio conocido en el paradigma de la Programación por capas representaría la integración entre Vista y su correspondiente Controlador de eventos y acceso a datos, MVC no pretende discriminar entre capa de negocio y capa de presentación pero si pretende separar la capa visual gráfica de su correspondiente programación y acceso a datos, algo que mejora el desarrollo y mantenimiento de la Vista y el Controlador en paralelo, ya que ambos cumplen ciclos de vida muy distintos entre sí.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace, etc.). El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.

El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.

El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se reflejan los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice. Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.

La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

De la Torre Llorente Cesar (Microsoft), Zorrilla Castro Unai (Plain Concepts), Calvarro Nelson Javier (Microsoft), Miguel Ángel Ramos Barroso (Microsoft).

“Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0”

Editorial; Krasis Press. Edición I.

1.5.1.2 Hibernate

Hibernate es una herramienta que realiza el mapping entre el mundo orientado a objetos de las aplicaciones y el mundo entidad-relación de las bases de datos en entornos Java. El término utilizado es ORM (object/relational mapping) y consiste en

la técnica de realizar la transición de una representación de los datos de un modelo relacional a un modelo orientado a objetos y viceversa.

Hibernate no solo realiza esta transformación sino que nos proporciona capacidades para la obtención y almacenamiento de datos de la base de datos que nos reducen el tiempo de desarrollo.

Ejemplo: Archivo de configuración Hibernate que se genera cuando se trabaja con framework Hibernate.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate
Configuration DTD 3.0//EN" "http://hibernate.sourceforge.net/hibernate-
configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property
name="hibernate.dialect">org.hibernate.dialect.PostgreSQLDialect</property>
    <property
name="hibernate.connection.driver_class">org.postgresql.Driver</property>
    <property
name="hibernate.connection.url">jdbc:postgresql://localhost:5432/siga</property>
    <property name="hibernate.connection.username">sistema_siga</property>
    <property name="hibernate.connection.password">siga</property>
    <property name="hibernate.show_sql">>true</property>
    <property name="hibernate.format_sql">>true</property>
    <mapping resource="mapeos/AcaGrupoEntidad.hbm.xml"/>
    <mapping resource="mapeos/AcaTipoHorario.hbm.xml"/>
    <mapping resource="mapeos/AcaPeriodo.hbm.xml"/>
    <mapping resource="mapeos/BibReferencia.hbm.xml"/>
    <mapping resource="mapeos/PubDireccion.hbm.xml"/>
    <mapping resource="mapeos/AcaAnioLectivo.hbm.xml"/>
    <mapping resource="mapeos/ComOrdenCompra.hbm.xml"/>
    <mapping resource="mapeos/PubEstado.hbm.xml"/>
    <mapping resource="mapeos/AcaUnidadDidactica.hbm.xml"/>
    <mapping resource="mapeos/AcaAsignatura.hbm.xml"/>
```



```
<mapping resource="mapeos/AcaAsignaturaEmpleado.hbm.xml"/>
<mapping resource="mapeos/DobPregunta.hbm.xml"/>
<mapping resource="mapeos/DobPlanificacionPrueba.hbm.xml"/>
<mapping resource="mapeos/PubCiudad.hbm.xml"/>
<mapping resource="mapeos/AcaCurso.hbm.xml"/>
<mapping resource="mapeos/SimEfectoSimulacro.hbm.xml"/>
<mapping resource="mapeos/AcaPromedioPeriodo.hbm.xml"/>
<mapping resource="mapeos/PubTelefono.hbm.xml"/>
<mapping resource="mapeos/AcaCapacitacion.hbm.xml"/>
<mapping resource="mapeos/AcaHorarioDetalle.hbm.xml"/>
<mapping resource="mapeos/ComActividadProveedor.hbm.xml"/>
<mapping resource="mapeos/ComDetalleActividad.hbm.xml"/>
<mapping resource="mapeos/BibAutor.hbm.xml"/>
<mapping resource="mapeos/AcaEspecialidad.hbm.xml"/>
<mapping resource="mapeos/InvMovimiento.hbm.xml"/>
<mapping resource="mapeos/SimSimulacro.hbm.xml"/>
<mapping resource="mapeos/MedTipoMalestar.hbm.xml"/>
<mapping resource="mapeos/SegPrivilegio.hbm.xml"/>
<mapping resource="mapeos/PubTipoEntidad.hbm.xml"/>
<mapping resource="mapeos/ComUnidadMedida.hbm.xml"/>
<mapping resource="mapeos/AcaAsistencia.hbm.xml"/>
<mapping resource="mapeos/AcaModalidad.hbm.xml"/>
<mapping resource="mapeos/AcaDia.hbm.xml"/>
<mapping resource="mapeos/AcaPromedioGradoGlobal.hbm.xml"/>
<mapping resource="mapeos/BibAlquilerDetalle.hbm.xml"/>
<mapping resource="mapeos/PubTipoDireccion.hbm.xml"/>
<mapping resource="mapeos/DobAlternativa.hbm.xml"/>
<mapping resource="mapeos/AcaTipoAula.hbm.xml"/>
<mapping resource="mapeos/DobResultado.hbm.xml"/>
<mapping resource="mapeos/AcaTipoExamen.hbm.xml"/>
<mapping resource="mapeos/PubTipoTelefono.hbm.xml"/>
<mapping resource="mapeos/AcaValoracionPromedio.hbm.xml"/>
<mapping resource="mapeos/AcaPlanAnualCurricular.hbm.xml"/>
```

```
<mapping resource="mapeos/DobRangoPuntaje.hbm.xml"/>
<mapping resource="mapeos/PubEstadoCivil.hbm.xml"/>
<mapping resource="mapeos/RolCargo.hbm.xml"/>
<mapping resource="mapeos/PubProfesion.hbm.xml"/>
<mapping resource="mapeos/AcaPromedioGrado.hbm.xml"/>
<mapping resource="mapeos/AcaPromedioAnioLectivo.hbm.xml"/>
<mapping resource="mapeos/AcaFichaConducta.hbm.xml"/>
<mapping resource="mapeos/MedConsulta.hbm.xml"/>
<mapping resource="mapeos/AcaAulaCursoParalelo.hbm.xml"/>
<mapping resource="mapeos/SegGrupoUsuario.hbm.xml"/>
<mapping resource="mapeos/InvItemTipo.hbm.xml"/>
<mapping resource="mapeos/ComRegistroFacturaItem.hbm.xml"/>
<mapping resource="mapeos/ComSolicitudCompraItem.hbm.xml"/>
<mapping resource="mapeos/AcaJornada.hbm.xml"/>
<mapping resource="mapeos/InvItemDetalle.hbm.xml"/>
<mapping resource="mapeos/InvRazonMovimiento.hbm.xml"/>
<mapping resource="mapeos/DobPruebaAlumno.hbm.xml"/>
<mapping resource="mapeos/DobAlternativaAngth.hbm.xml"/>
<mapping resource="mapeos/AcaEstructuraAcademica.hbm.xml"/>
<mapping resource="mapeos/InvInventario.hbm.xml"/>
<mapping resource="mapeos/BibLibroXAutor.hbm.xml"/>
<mapping resource="mapeos/RolCargoEmpleado.hbm.xml"/>
<mapping resource="mapeos/AcaCiclo.hbm.xml"/>
<mapping resource="mapeos/AcaCalificacionParcial.hbm.xml"/>
<mapping resource="mapeos/AcaPlanUnidadDidactica.hbm.xml"/>
<mapping resource="mapeos/AcaTemaUnidad.hbm.xml"/>
<mapping resource="mapeos/AcaTipoCapacitacion.hbm.xml"/>
<mapping resource="mapeos/SimTipoCatastrofe.hbm.xml"/>
<mapping resource="mapeos/AcaAlumno.hbm.xml"/>
<mapping resource="mapeos/ComContactoProveedor.hbm.xml"/>
<mapping resource="mapeos/InvMovimientoTipo.hbm.xml"/>
<mapping resource="mapeos/AcaMateriaExamenGrado.hbm.xml"/>
<mapping resource="mapeos/DobFactorMg.hbm.xml"/>
```

```
<mapping resource="mapeos/AcaSeccion.hbm.xml"/>
<mapping resource="mapeos/AcaCategoriaCalificacion.hbm.xml"/>
<mapping resource="mapeos/InvInventarioDetalle.hbm.xml"/>
<mapping resource="mapeos/DobTipoPrueba.hbm.xml"/>
<mapping resource="mapeos/AcaSupletorioRegistro.hbm.xml"/>
<mapping resource="mapeos/ComProveedor.hbm.xml"/>
<mapping resource="mapeos/SegOpcionesModulo.hbm.xml"/>
<mapping resource="mapeos/AcaAula.hbm.xml"/>
<mapping resource="mapeos/ComRegistroFactura.hbm.xml"/>
<mapping resource="mapeos/AcaRangoSupletorio.hbm.xml"/>
<mapping resource="mapeos/DobRangoEdad.hbm.xml"/>
<mapping resource="mapeos/BibPercha.hbm.xml"/>
<mapping resource="mapeos/AcaExamenCalificacion.hbm.xml"/>
<mapping resource="mapeos/PubIdentificacion.hbm.xml"/>
<mapping resource="mapeos/DobApreciacion.hbm.xml"/>
<mapping resource="mapeos/InvItemCategoria.hbm.xml"/>
<mapping resource="mapeos/SimPlanEvacuacion.hbm.xml"/>
<mapping resource="mapeos/PubSexo.hbm.xml"/>
<mapping resource="mapeos/AcaEntidad.hbm.xml"/>
<mapping resource="mapeos/InvItem.hbm.xml"/>
<mapping resource="mapeos/AcaActividadEnsenanzaAprendizaje.hbm.xml"/>
<mapping resource="mapeos/RolArea.hbm.xml"/>
<mapping resource="mapeos/AcaSumaCalificacionCategoria.hbm.xml"/>
<mapping resource="mapeos/RolDepartamento.hbm.xml"/>
<mapping resource="mapeos/BibAlquilerCabe.hbm.xml"/>
<mapping resource="mapeos/InvMarca.hbm.xml"/>
<mapping resource="mapeos/PubPersona.hbm.xml"/>
<mapping resource="mapeos/DobPrueba.hbm.xml"/>
<mapping resource="mapeos/DobPreguntaFactor.hbm.xml"/>
<mapping resource="mapeos/AcaHorarioCabecera.hbm.xml"/>
<mapping resource="mapeos/PubTelefonoEmpresa.hbm.xml"/>
<mapping resource="mapeos/AcaParacademico.hbm.xml"/>
<mapping resource="mapeos/PubPais.hbm.xml"/>
```

```

<mapping resource="mapeos/ComOrdenCompraItem.hbm.xml"/>
<mapping resource="mapeos/BibPerchaCategoria.hbm.xml"/>
<mapping resource="mapeos/AcaActividad.hbm.xml"/>
<mapping resource="mapeos/PubTipoIdentificacion.hbm.xml"/>
<mapping resource="mapeos/AcaParcial.hbm.xml"/>
<mapping resource="mapeos/AcaAtraso.hbm.xml"/>
<mapping resource="mapeos/SegUsuario.hbm.xml"/>
<mapping resource="mapeos/RolEmpleado.hbm.xml"/>
<mapping resource="mapeos/AcaDetalleCategorias.hbm.xml"/>
<mapping resource="mapeos/BibLibroXCategoria.hbm.xml"/>
<mapping resource="mapeos/DobPreguntaAlternativa.hbm.xml"/>
<mapping resource="mapeos/AcaSistemaCalificacion.hbm.xml"/>
<mapping resource="mapeos/AcaAlumnoCurso.hbm.xml"/>
<mapping resource="mapeos/DobPruebaPregunta.hbm.xml"/>
<mapping resource="mapeos/DobRangoRaven.hbm.xml"/>
<mapping resource="mapeos/PubProvincia.hbm.xml"/>
<mapping resource="mapeos/SegPrivilegioModulo.hbm.xml"/>
<mapping resource="mapeos/SegModulo.hbm.xml"/>
<mapping resource="mapeos/DobDetalleResolucion.hbm.xml"/>
<mapping
resource="mapeos/AcaRegistroCalificacionDetalleCategoria.hbm.xml"/>
  <mapping resource="mapeos/AcaTipoParacademico.hbm.xml"/>
  <mapping resource="mapeos/MedReceta.hbm.xml"/>
  <mapping resource="mapeos/ComSolicitudCompra.hbm.xml"/>
  <mapping resource="mapeos/BibLibro.hbm.xml"/>
  <mapping resource="mapeos/BibCategoria.hbm.xml"/>

</session-factory>
</hibernate-configuration>

```

Tabla #1: Archivo de configuración Hibernate.

Fuente: Los autores

1.5.1.3 Apache Struts

Struts es una herramienta de soporte para el desarrollo de aplicaciones Web bajo el patrón MVC bajo la plataforma Java EE (Java Enterprise Edition). Struts se desarrollaba como parte del proyecto Jakarta de la Apache Software Foundation, pero actualmente es un proyecto independiente conocido como Apache Struts.

Struts permite reducir el tiempo de desarrollo. Su carácter de "software libre" y su compatibilidad con todas las plataformas en las que Java Enterprise esté disponible lo convierten en una herramienta altamente disponible.

Con la versión 2 del framework se introdujeron algunas mejoras sobre la primera versión, de cara a simplificar las tareas más comunes en el desarrollo de aplicaciones web, así como mejorar su integración con AJAX, etc.

Struts se basa en el patrón de arquitectura de software Modelo-Vista-Controlador (MVC) el cual se utiliza ampliamente y es considerado de gran solidez. De acuerdo con este Framework, el procesamiento se separa en tres secciones diferenciadas llamadas el modelo, las vistas y el controlador.

Fuente: http://es.wikipedia.org/wiki/Apache_Struts.

1.5.1.4 Extjs

Es una librería JavaScript que permite construir aplicaciones complejas en Internet.

Esta librería incluye:

- Componentes UI del alto performance y personalizables.
- Modelo de componentes extensibles.
- Un API fácil de usar.
- Licencias Open source y comerciales.

Antes de poder entrar a examinar ExtJS primero tenemos que hablar sobre RIA, acrónimo de Rich Internet Applications (Aplicaciones Ricas en Internet). Lo que RIA intenta proveer es aquello de lo que siempre ha adolecido la web, una experiencia de usuario muy parecida o igual a la que se tiene en las aplicaciones de escritorio.

Las aplicaciones web tradicionales tienen problemas como la recarga continua de las páginas cada vez que el usuario pide nuevo contenido, o la poca capacidad multimedia, para lo cual se han hecho necesarios plug-ins externos.

Junto con el reto de llevar la experiencia RIA a los usuarios comenzó el debate sobre cuál sería el mejor modo de atacar el problema. La historia de los últimos años nos ha traído diversas tecnologías, basadas en Flash (Adobe), Java (Sun), Silverlight (MS). Todas muy interesantes, pero con la desventaja de necesitar algún tipo de extensión en los navegadores que podría no estar presente. Ha sido esta limitante lo que le ha dado la victoria (al menos por el momento) al casi dejado de lado Javascript y la “nueva” tecnología conocida como AJAX.

ExtJS encaja dentro de este esquema como un motor que permite crear aplicaciones RIA mediante Javascript. Si enmarcamos a ExtJS dentro del desarrollo RIA, éste sería el render de la aplicación que controla el cliente y que se encarga de enviar y obtener información del servicio.

Una de las grandes ventajas de utilizar ExtJS es que nos permite crear aplicaciones complejas utilizando componentes predefinidos así como un manejador de layouts similar al que provee Java Swing, gracias a esto provee una experiencia consistente sobre cualquier navegador, evitando el tedioso problema de validar que el código escrito funcione bien en cada uno (Firefox, IE, Safari, etc.).

Además la ventana flotante que provee ExtJS es excelente por la forma en la que funciona. Al moverla o redimensionarla solo se dibujan los bordes haciendo que el movimiento sea fluido lo cual le da una ventaja tremenda frente a otros.

Usar un motor de render como ExtJS nos permite tener además estos beneficios: Existe un balance entre Cliente – Servidor. La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.

Comunicación asíncrona. En este tipo de aplicación el motor de render puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se de cuenta.

Eficiencia de la red. El tráfico de red puede disminuir al permitir que la aplicación

elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico.

Ejemplo: Referencias que se deben usar para agregar a cada página web donde se desee utilizar componentes ExtJS.

```
1 <link rel="stylesheet" type="text/css" href="ext/resources/css/ext-  
all.css" />  
  
2 <script type="text/javascript" src="ext/ext-all.js"></script>  
  
3 <script type="text/javascript" src="ext/adapter/ext/ext-  
base.js"></script>  
  
4 <script type="text/javascript" src="ext/ext-all.js"></script>
```

Tabla #2: Cabecera necesarias para integrar las interfaces de ExtJS.

Fuente:<http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>.

1.5.1.5 Netbeans

¿Qué es Netbeans?

Netbeans es un entorno de desarrollo integrado (IDE) que permite editar programas en java, compilarlos, ejecutarlos, depurarlos, construir rápidamente el interfaz gráfico de una aplicación eligiendo los componentes de una paleta, etc.

http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Fundamentos_de_Programacion_Parte_2.pdf

Características Fundamentales de Netbeans

- EL IDE de Netbeans es gratuito y de código abierto para desarrolladores de software.
- En esta versión tienes al alcance de tu mano todas las herramientas necesarias para crear aplicaciones profesionales para entornos de escritorio, empresa, web y móviles, ya sea en C/C++, Java e incluso Ruby. El IDE ha sido desarrollado para distintas plataformas como Linux, MacOS X, Solaris y también Windows.
- Proceso simplificado de instalación que permite la fácil instalación y configuración según las necesidades del usuario.

Ventajas de Netbeans

Es un entorno de desarrollo, Netbeans fue diseñado principalmente para desarrollar código Java, adicionalmente existe un número importante de módulos para extender el IDE, cabe indicar que es un producto libre y gratis para uso general.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados módulos. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

Fuente: <http://es.wikipedia.org/wiki/NetBeans>

Modularidad. Todas las funciones del IDE son provistas por módulos. Cada módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. NetBeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permitiéndole al usuario comenzar a trabajar inmediatamente.

Código de Ejemplo de JSP

```
%--
Document : seleccionCursosParcial
Created on : 11-abr-2012, 22:47:26
Author : Eliana
--%>

<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<jsp:include page="../extjs/ext-include.jsp"/>
<script type="text/javascript" src="js/SeleccionCursosWindow.ui.js"></script>
<script type="text/javascript" src="js/SeleccionCursosWindow.js"></script>
<script type="text/javascript" src="js/dsAsignaturaEmpleado.js"></script>

<script type="text/javascript"
src="js/calificacionParcialWindow.ui.js"></script>
<script type="text/javascript" src="js/calificacionParcialWindow.js"></script>
<script type="text/javascript" src="js/dsPeriodoActivo.js"></script>
<script type="text/javascript" src="js/dsParcialActivo.js"></script>
<script type="text/javascript" src="js/dsCalificacionParcial.js"></script>
<script type="text/javascript" src="../disciplina/js/dsCurso.js"></script>
<script type="text/javascript"
src="../disciplina/js/dsCursoAsignatura.js"></script>
<script type="text/javascript" src="js/dsParalelos.js"></script>
</head>
<body>
<script language="JavaScript">
Ext.onReady(function() {
```

```
Ext.QuickTips.init();
var cmp10 = new SeleccionCursosWindow({
    renderTo: Ext.getBody()
});
cmp10.show();
});
</script>
</body>
</html>
```

Tabla #3: Código de ejemplo de ASP

Fuente: Los autores

1.5.1.6 ¿Por qué utilizar el lenguaje JSP en vez de ASP?

JSP Y ASP permiten realizar un trabajo similar, sin embargo si analizamos los tipos de herramientas para una solución compleja habría que tener en cuenta lo siguiente:

- En cuanto a plataforma, el uso de JSP tiene ventajas ya que sigue la filosofía Java, “escribe una vez ejecuta donde sea”.
- JSP se lo puede utilizar en servidores web más comunes, mientras que ASP solo tiene soporte nativo con IIS y Personal Web Server.
- El API de JSP se beneficia por la extensa comunidad que contribuye con el uso y producción con esta herramienta, mientras que ASP trabaja de forma interna y reservada.
- Los componentes JSP son reusables, utilizan distintas plataformas como (Unix y Windows).
- Java es un lenguaje más potente y escalable que los lenguajes Scripts.
- Java ayuda a proteger el sistema contra caídas, mientras que ASP es más propenso a sufrirlas.
- Java es un lenguaje estructurado y es más fácil de construir y mantenimientos grandes como aplicaciones modulares.

- La tecnología JSP es abierta y multiplataforma, los servidores web, plataformas y otros componentes pueden ser fácilmente actualizados o cambiados sin que afecte a las aplicaciones basadas en dicha tecnología.

Conclusión: Hemos podido apreciar la mayoría de beneficios que ofrece trabajar con JSP, sin embargo hay que tener en cuenta que ASP es más sencillo de aprender por lo que es más intuitivo para el usuario y no requiere de un experto para la implementación.

1.5.1.7 Ajax

¿Qué es Ajax?

Ajax es una técnica de desarrollo web para crear aplicaciones interactivas, estas aplicaciones se ejecutan del lado del cliente, es decir, en el navegador de los usuarios, mientras se mantiene la conexión asíncrona con el servidor en segundo plano. De esta forma el usuario puede contar con usabilidad, interactividad y velocidad, ya que el usuario puede trabajar en la interfaz sin necesidad de actualizarla o recargarla.

Ajax es una tecnología asíncrona, teniendo en cuenta que la solicitud para el servidor se hace en segundo plano sin necesidad que la operación interfiera con la interfaz que el usuario manipula. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. Para este caso no es necesario que el contenido asíncrono este con formato XML.

Tecnologías que incluyen AJAX

Ajax también se puede definir como una combinación de cuatro tecnologías existentes:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario.

- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Características de Ajax

- Las páginas con AJAX son más difíciles de desarrollar que las páginas estáticas.
- Las páginas creadas dinámicamente mediante peticiones sucesivas AJAX, no son registradas de forma automática en el historial del navegador, así que haciendo clic en el botón de "volver" del navegador, el usuario no será devuelto a un estado anterior de la página, en cambio puede volver a la última página que visitó. Soluciones incluyen el uso de IFrames invisible para desencadenar cambios en el historial del navegador y el cambio de la porción de anclaje de la dirección (después de un #).
- Los motores de búsquedas no entienden JavaScript. La información en la página dinámica no se almacena en los registros del buscador.
- Hay problemas usando Ajax entre nombres de dominios. Eso es una función de seguridad.
- El sitio con Ajax usa más recursos en el servidor. Recomendación: sólo usar las peticiones necesarias en Ajax, no desarrollar todo el sitio en AJAX. Con esto garantizamos menos recursos del servidor.
- Es posible que páginas con Ajax no puedan funcionar en teléfonos móviles, PDA u otros aparatos. Ajax no es compatible con el software para ciegos u otras discapacidades.

Fuente: <http://es.wikipedia.org/wiki/AJA>



Figura #1: Esquema descriptivo de funcionamiento de Ajax

Fuente: Eguíluz Pérez Javier, Junio 2008 "Introducción a Ajax".

<http://www.librosweb.es/ajax/index.html>

CalificacionCategPeriodo.jsp

Código:

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <jsp:include page="../extjs/ext-include.jsp"/>

    <script type="text/javascript" src="js/calificacionCategPeriodoViewPort.ui.js"></script>
    <script type="text/javascript" src="js/calificacionCategPeriodoViewPort.js"></script>
    <script type="text/javascript" src="js/dsPeriodo.js"></script>
    <script type="text/javascript" src="js/dsAsignaturaEmpleado.js"></script>
    <title>JSP Page</title>
  </head>
  <body>
    <script language="JavaScript">
      Ext.onReady(function() {
        Ext.QuickTips.init();

        var cmp = new calificacionCategPeriodoViewPort({
          renderTo: Ext.getBody()
        });
        cmp.show();
      });
    </script>
  </body>
```

```

<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <jsp:include page="../extjs/ext-include.jsp"/>

    <script type="text/javascript" src="js/calificacionCategPeriodoViewPort.ui.js"></script>
    <script type="text/javascript" src="js/calificacionCategPeriodoViewPort.js"></script>
    <script type="text/javascript" src="js/dsPeriodo.js"></script>
    <script type="text/javascript" src="js/dsAsignaturaEmpleado.js"></script>
    <title>JSP Page</title>
  </head>
  <body>
    <script language="JavaScript">
      Ext.onReady(function() {
        Ext.QuickTips.init();

        var cmp = new calificacionCategPeriodoViewPort({
          renderTo: Ext.getBody()
        });
        cmp.show();
      });
    </script>
  </body>

```

</html>index.jsp

Código:

```
<%--
  Document : index
  Created on : 24/03/2011, 09:52:30 PM
  Author : Eliana & Dalton
--%>

<%
String contexto = request.getContextPath();
%>

<% @page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">

<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <link rel="shortcut icon" href="<%=contexto%>/icons/16x16/star.png" >
    <script language="Javascript">

      function principal()
      {
        aWidth =screen.availWidth - 10;
        aHeight=screen.availHeight - 30;

        ruta = "<%=contexto%>/general/Principal.jsp";

        abrirVentana(ruta,"Principal", 'toolbar=no,status=yes,scrollbars=no,resizable=yes',aWidth,aH
eight,true');
      }

      function abrirVentana(pagina,target,caracteristicas, ancho, alto, centrar) {
        if(window.screen)if(centrar)if(centrar=="true"){
          var left = (screen.width-ancho)/2;
          var top = (screen.height-alto)/2;
          caracteristicas+=(caracteristicas!="")?',' ':'';
          caracteristicas+=',left='+left+',top='+top;
        }
      }
    </script>
  </head>
</html>
```



```
Var                                ventana                                =
window.open(pagina,target,caracteristicas+((caracteristicas!='')?',';')+ 'width='+ancho+'height
='+alto);
    return ventana;
}
</script>
</head>
<body>
    <script language="Javascript">
        principal();
    </script>
</body>
```

Código Ajax:

```
function createRequestObject(){
var req;
if(window.XMLHttpRequest){

req = new XMLHttpRequest();
}
else if(window.ActiveXObject){
//For IE 5+
req = new ActiveXObject("Microsoft.XMLHTTP");
}
else{
}

return req;
}
//Make the XMLHttpRequest Object
var http = createRequestObject();

function sendRequest(method, url){
if(method == 'get' || method == 'GET'){
http.open(method,url);
http.onreadystatechange = handleResponse;
http.send(null);
}
}

function handleResponse(){
if(http.readyState == 4 && http.status == 200){
var response = http.responseText;
if(response){
document.getElementById("ajax_res").innerHTML = response;}}
}
```

Tabla #4: Ejemplo de Código jsp y ajax de calificación por periodo.

Fuente: <http://www.forodejava.com/showthread.php/286-Ejemplo-de-ajax-con-jsp>

1.5.1.8 Postgres

¿Qué es Postgres?

PostgreSQL es un sistema de base de datos relacional, libre y orientado a objetos, publicado bajo licencia BSD, al igual que otros proyectos y lenguajes por ser código abierto no es manejado por una empresa sino más bien por una comunidad sin fines de lucro, dicha comunidad es denominada PGDG (Postgres SQL Development Group).

Entre sus características podemos resaltar las siguientes:

- Alta Concurrencia, esto se debe al uso de un sistema llamado MVCC (Acceso Concurrente Multiversion), PostgreSQL permite que mientras un proceso escribe en una tabla, otros acceden a la misma tabla sin necesidad de bloqueo, cada usuario tiene acceso de forma consistente de lo último a lo que se hizo un commit.

- PostgreSQL provee nativamente soporte para:
 - Números de precisión arbitraria.
 - Texto de largo ilimitado.
 - Figuras geométricas (con una variedad de funciones asociadas).
 - Direcciones IP (IPv4 e IPv6).
 - Bloques de direcciones estilo CIDR.
 - Direcciones MAC.
 - Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

- Claves ajenas también denominadas Llaves ajenas o Claves Foráneas (foreign keys).
- Disparadores (triggers): Un disparador o trigger se define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de

la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:

- El nombre del disparador o trigger
 - El momento en que el disparador debe arrancar
 - El evento del disparador deberá activarse sobre...
 - La tabla donde el disparador se activará
 - La frecuencia de la ejecución
 - La función que podría ser llamada
 - Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores (triggers).
-
- Vistas.
 - Integridad transaccional.
 - Herencia de tablas.
 - Tipos de datos y operaciones geométricas.
 - Soporte para transacciones distribuidas. Permite a PostgreSQL integrarse en un sistema distribuido formado por varios recursos (p.ej, una base de datos PostgreSQL, otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción global es el resultado del éxito de las transacciones locales.

1.5.1.9 Funciones

Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Los disparadores (triggers en inglés) son funciones enlazadas a operaciones sobre los datos.

Algunos de los lenguajes que se pueden usar son los siguientes:

Un lenguaje propio llamado PL/PgSQL (similar al PL/SQL de oracle).

- ✓ C.
- ✓ C++.
- ✓ Java PL/Java web.
- ✓ PL/Perl.
- ✓ plPHP.
- ✓ PL/Python.
- ✓ PL/Ruby.
- ✓ PL/sh.
- ✓ PL/Tcl.
- ✓ PL/Scheme.

Lenguaje para aplicaciones estadísticas R por medio de PL/R.

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta (query en inglés).

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como "procedimientos almacenados" (stored procedures en inglés).

Fuente: <http://es.wikipedia.org/wiki/PostgreSQL>

1.5.2.1 Modelo relacional de datos

El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente. Tras ser postuladas sus bases en 1970 por Edgar Frank Codd, de los laboratorios IBM en San José (California), no tardó en consolidarse como un nuevo paradigma en los modelos de base de datos.

Su idea fundamental es el uso de (relaciones). Estas relaciones podrían considerarse en forma lógica como conjuntos de datos llamados (tuplas). Pese a que ésta es la teoría de las bases de datos relacionales creadas por Edgar Frank Codd, la mayoría de las veces se conceptualiza de una manera más fácil de imaginar, esto es, pensando en cada relación como si fuese una tabla que está compuesta por registros (cada fila de la tabla sería un registro o tupla), y columnas (también llamadas campos).

Descripción

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que éstos se almacenen no tiene relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario el cual no requiere de mayor conocimiento y experiencia. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. **Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.**

Para manipular la información utilizamos un lenguaje relacional, actualmente se cuenta con dos lenguajes formales el Álgebra relacional y el Cálculo relacional. El Álgebra relacional permite describir la forma de realizar una consulta, en cambio, el Cálculo relacional sólo indica lo que se desea devolver.

1.5.2.2 Esquema

Un esquema es la definición de una estructura (generalmente relaciones o tablas de una base de datos), es decir, determina la identidad de la relación y que tipo de información podrá ser almacenada dentro de ella; en otras palabras, el esquema son los metadatos de la relación. Todo esquema constará de:

- Nombre de la relación (su identificador).
- Nombre de los atributos (o campos) de la relación y sus dominios; el dominio de un atributo o campo define los valores permitidos para el mismo, es equivalente al tipo de dato por ejemplo: character, integer, date, string, etc.

1.5.2.3 Instancias

Una instancia de manera formal es la aplicación de un esquema a un conjunto finito de datos. En palabras no tan técnicas, se puede definir como el contenido de una tabla en un momento dado, pero también es válido referirnos a una instancia cuando trabajamos o mostramos únicamente un subconjunto de la información contenida en una relación o tabla, como por ejemplo:

- Ciertos caracteres y números (una sola columna de una sola fila).
- Algunas o todas las filas con todas o algunas columnas.
- Cada fila es una tupla. El número de filas es llamado cardinalidad.
- El número de columnas es llamado aridad o grado.

1.5.2.4 Base de datos relacional

Una base de datos relacional es un conjunto de una o más tablas estructuradas en registros (líneas) y campos (columnas), que se vinculan entre sí por un campo en común, en ambos casos posee las mismas características como por ejemplo el nombre de campo, tipo y longitud; a este campo generalmente se le denomina ID, identificador o clave. A esta manera de construir bases de datos se le denomina modelo relacional.

Hablando estrictamente el término se refiere a una colección específica de datos pero a menudo se le usa, en forma errónea como sinónimo del software usado para gestionar esa colección de datos. Ese software se conoce como SGBD (sistema gestor de base de datos) relacional o RDBMS (del inglés relational database management system).

Las bases de datos relacionales pasan por un proceso al que se le conoce como normalización de una base de datos, el cual es entendido como el proceso necesario para que una base de datos sea utilizada de manera óptima.

Fuente: http://es.wikipedia.org/wiki/Modelo_relacional

publico.aca_atraso	publico.aca_control_faltas
id_parcial: integer	id_control_faltas: serial
id_atraso: serial	id_alumno: integer
id_estado: integer	id_asignatura: integer
id_alumno: integer	id_aula_curso_paralelo: integer
id_asignatura: integer	id_estado: integer
id_aula_curso_paralelo: integer	id_parcial: integer
id_estructura_academica: integer	id_periodo: integer
fecha: date	id_estructura_academica: integer
atraso: boolean	observacion: text
observacion: text	falta: boolean
atraso_pkey	falta_justificada: boolean
	fecha: date
	aca_control_faltas_pkey

publico.aca_ficha_conducta
id_parcial: integer
id_ficha_conducta: serial
id_estado: integer
id_alumno: integer
id_asignatura: integer
id_aula_curso_paralelo: integer
id_periodo: integer
id_estructura_academica: integer
descripcion: text
fecha: date
ficha_conducta_pkey

Figura #2: Tablas correspondiente a la base de datos siga, módulo calificación

Fuente: Los autores

1.5.2.5 La Entrevista

Una entrevista es un diálogo entablado entre dos o más personas: el entrevistador o entrevistadores que interrogan y el o los entrevistados que contestan. Se trata de una técnica o instrumento empleado para diversos motivos, investigación, medicina, selección de personal. Una entrevista es un diálogo interesado, con un acuerdo previo y unos intereses y expectativas por ambas partes. También la entrevista puede significar mucho para otras personas ya que pueden ayudar a conocer personas de máxima importancia. El diccionario de la real academia española define la palabra Entrevista como: la conversación que tiene como finalidad la obtención de información.

En una entrevista intervienen básicamente dos personas: El entrevistador y el entrevistado. El primero, además de tomar la iniciativa de la conversación, plantea mediante preguntas específicas cada tema de su interés y decide en qué momento el tema ha cumplido sus objetivos. El entrevistado facilita información sobre sí mismo, su experiencia o el tema en cuestión.

1.5.2.6 El Entrevistador

El entrevistador debe tener disposición y paciencia para así llevar a cabo con éxito una entrevista.

Debe hablar de manera clara, precisa y en voz entendible.

Es conveniente que el entrevistador hable en voz regulable ya que puede ayudar a mejorar de alguna u otra manera el resultado de la entrevista.

Debe tener buena y clara letra para que sean entendibles las respuestas del entrevistado.

Las preguntas debe hacerlas de manera natural para que el entrevistado responda con sinceridad y se sienta a gusto entrevistándose.

Las preguntas deben ser precisas y deben ser sencillas (cortas), deben ser exactas a lo que se quiere preguntar, y adecuadas al nivel educativo del entrevistado, y la entrevista debe ser en el menor tiempo posible, o depende de la disponibilidad de tiempo de ambas partes.

El entrevistador debe estar seguro de lo que desea preguntar, si es necesario anotar las preguntas antes de comenzar la entrevista.

El entrevistador debe estar preparado para realizar preguntas improvisadas a partir de las respuestas

El entrevistador debe tener información sobre la persona la cual entrevistará.

1.5.2.7 El entrevistado

Uno de los requisitos para que haya una entrevista es que el sujeto entrevistado esté de acuerdo. Esto ocurre normalmente porque tiene algún interés en ser entrevistado.

Las seis razones principales que puede tener son las siguientes:

- Por su propio ego: por el deseo de aparecer en los medios o que se escuche su opinión.
- Por publicidad: sobre todo los políticos y entre ellos los famosos, que dependen de la publicidad para influir en la opinión pública o para continuar en la brecha. Consideran las entrevistas como publicidad gratuita.
- Por dinero: aunque según muchos códigos deontológicos los periodistas no deberían pagar a los entrevistados es una práctica frecuente en medios sensacionalistas o amarillistas. Algunos famosos llegan a tener grandes ingresos por ello.
- Para ayudar al periodista: algunos sujetos se dejan entrevistar simplemente para ayudar al periodista en el desempeño de su trabajo.
- Por gratitud: El entrevistador logra que el entrevistado acceda a contestar las preguntas sin oponer resistencia mediante un diálogo introductorio predefinido.

La entrevista como instrumento de investigación ha sido utilizada de forma ambiciosa por antropólogos, sociólogos, psicólogos, politólogos o economistas. Es

por ello que gran parte de los datos con que cuentan las ciencias sociales proceden de las entrevistas. Los científicos sociales dependen de ellas para obtener información sobre los fenómenos investigados y comprobar así sus teorías e hipótesis.

Fuente: http://es.wikipedia.org/wiki/Entrevista_period%C3%ADstica

En función de la técnica de interrogar

Los tipos principales de entrevista, desde este punto de vista son:

Entrevista estandarizada: es una especie de cuestionario oral. El entrevistador lee las preguntas y anota las respuestas que obtiene. Se suele utilizar en los procesos de selección de búsqueda pasiva, cuando hay muchos candidatos.

Entrevista libre: es como una charla de carácter informal, que permite la obtención de datos subjetivos acerca del candidato.

Entrevista mixta: no se sirve de un cuestionario fijo, ni plantea unas preguntas prefijadas, sino que sigue una especie de guía de entrevista en las que se detallan, no las preguntas sino las áreas que han de ser exploradas.

En función del clima de la entrevista

Entrevista normal o relajada: se busca conseguir un clima, sereno, tranquilo y de confianza.

Entrevista dura: parte del principio de que en condiciones de tensión afloran más fácilmente los rasgos ocultos de la personalidad. El entrevistador utiliza distintas técnicas para inducir **tensión en el candidato**. Se utilizan para puestos que requieren un **fuerte control emocional**.

Fuente: <http://desarrollo-profesional.universia.es/mercado-laboral/proceso-seleccion/entrevistas/tipos-entrevistas/>

1.5.2.8 Sistema Integrado de Gestión Académica

¿Qué es un sistema integrado de gestión académica?

Es una solución global e integrada que facilita el desempeño operacional y productividad de cualquier institución educativa basada en la transparencia y automatización de procesos, los mismos que han sido desarrollados en base a los requerimientos del usuario.

Adicionalmente el usuario cuenta con la facilidad de poder realizar cualquier transacción de forma normal sin tener que depender de algún otro proceso para poder integrar la información, uno de los principales beneficios es que ofrece personalización según la necesidad del cliente, el tipo de personalización es aplicable a procesos como: Matriculación, Calificación, Planificación Académica, DOBE, etc. Todo esto garantiza un mejor control interno, y permite cumplir con la planificación estratégica de la compañía según el tiempo establecido.

Hay que tener en cuenta que la automatización e integración de los diferentes módulos disminuye la carga operativa para el recurso humano, ofrece un mejor desempeño y garantiza la integridad de los datos.

¿Qué es evaluación y gestión de calificaciones?

La Evaluación Y Gestión de Calificaciones es donde el docente registra el desempeño del alumno, allí ingresa las notas correspondientes al parcial, exámenes, paracadémico, supletorio. Nos puede generar un performance del estudiante de una materia en especial. Todo el proceso mencionado se lo puede aplicar dependiendo de la modalidad de estudio, es decir, bimestre, trimestre, quinquimestre; Jornada: Matutino, Vespertino, Nocturno; Modalidad: Presencial a distancia.

Gestión de Calificaciones elimina el cálculo manual de notas, mitigando el riesgo de almacenar datos inconsistentes, disminuyendo reclamos de calificaciones, permite que la información esté disponible para quien la necesite.

¿Qué es gestión de convivencia y régimen académico?

La gestión de convivencia es un registro de asistencia al alumno, de donde se podrá realizar justificaciones, control necesario para poder informar de forma oportuna al representante o direccionarlo al DOBE (Dpto. Bienestar Estudiantil).

Adicional también se puede registrar la conducta del alumno, logrando tener un histórico por alumno, curso, materia, dicha información nos permitirá conocer el alumno que cometió una falta grave, así como buscar una solución oportuna para dicho caso o en su defecto cumplió a cabalidad con las normas de la institución y logro honores.

Objetivo de gestión de calificaciones

Tiene como principal objetivo el registro de calificaciones del alumnado, calificado por Año Lectivo, Jornada, modalidad, Curso, Especialización, según los parámetros que requiera la institución. La información obtenida es de utilidad para realizar seguimiento sobre el desempeño de un alumno y para poder determinar posibles obstáculos de aprendizaje según el método de enseñanza, falencias por parte del profesor, etc.

Los docentes deben ingresar las notas mediante el acceso disponible que se otorgue ya que el departamento de secretaría debe recibir la información a través del sistema para proceder a la elaboración de libretas y generar reportes para la junta de curso.

La información que se ingrese formará parte de una gran base de conocimiento para analizar el comportamiento del estudiante, la intención no es solamente poder detectar el origen de un posible problema, es también brindar información relevante para poder corregirlo a tiempo.

Objetivo de convivencia y régimen académico

Registrar datos relacionados al comportamiento, es decir, conducta del alumno, y lograr tomar una decisión a tiempo para que la dirección correspondiente pueda hacer frente al caso y resolverlo. En este caso lo más recomendable es que el Departamento DOBE en conjunto con los representantes trabaje en talleres para poder contribuir y ser parte activa del desarrollo de sus hijos

Características de gestión de calificaciones, convivencias y régimen académico.

1.- Consistencia

Tanto las calificaciones de las materias, como las de disciplina y faltas se registran, garantizando así la disponibilidad de la información y veracidad.

2.- Seguridad

El acceso por parte de cada usuario para el uso del software cuenta con su respectivo logon, este paso no se puede obviar pues dependiendo de la categoría de usuario tendrá acceso a ciertos modulo o funciones y a otros no.

3.- Única

Por ser un sistema automatizado e integrado, la información que fluye no es repetitiva, lo cual garantiza el punto 1, consistencia de datos.

4.- Acoplable

Dependiendo de la necesidad del cliente y del entorno, los módulos se acoplan para fácilmente trabajar según la disposición, teniendo un alcance variado.

5.- Accesible

Por contar con una interfaz agradable, llamativa e intuitiva, produce que los usuarios tengan más facilidad para acceder a las bondades que brinda la herramienta.

6.- Estabilidad

El sistema está diseñado con herramientas que soportan una carga operativa progresiva sin necesidad de contar con algún tipo de interrupción, caída o tiempo de espera que perjudique el proceso normal.

Diferencias entre procesos manuales y automáticos de gestión de calificación.

SISTEMA DE CALIFICACION TRADICIONAL	SISTEMA DE CALIFICACION AUTOMATIZADA
Utiliza gran cantidad de recursos, en cuanto a papel y carpetas, así como espacio para bodegaje.	Las transacciones se almacenan en el Disco Duro del servidor en donde reside la base de datos.
Se tiene que invertir gran cantidad de tiempo para realizar el proceso de calificación, sin considerar el tiempo que toma que secretaria depure y clasifique la calificación para presentarla.	La función de ingreso y registro de calificaciones se las hace de forma directa en el sistema facilitando el uso de la data para otro proceso/cálculo para el mismo u otro departamento.
El tiempo que demanda obtener el resultado esperado depende de la cantidad de alumno que se tenga que calificar (Cantidad de Transacciones).	Se obtienen resultados estadísticos inmediatos.
Debido a que el ingreso de calificaciones por parte del profesor se realiza de forma manual se puede incurrir en pérdida del medio que justifica las calificaciones (Hoja de Registro, lecciones, exámenes, etc.)	Se garantiza la consistencia y veracidad de la información ya que reside en una base solida con los parámetros de seguridad correspondiente.

Tabla #5: Cuadro diferencial de procesamiento de gestión de calificación.

Fuente: Los autores

1.6 Marco conceptual

AJAX: siglas en inglés de Asynchronous JavaScript And XML (JavaScript asíncrono y XML).

Biblioteca: es un conjunto de subprogramas utilizados para desarrollar software. Contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular.

BSD: siglas en inglés Berkeley Software Distribution. Es una licencia de software libre que tiene menos restricciones en comparación con otras como la **GPL** (*General Public License*) estando muy cercana al dominio público. La licencia BSD al contrario que la GPL permite el uso del código fuente en software no libre.

Clase: Declaración o abstracción de objetos, lo que significa, que una clase es la definición de un objeto. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase.

CGI: Interfaz de entrada común (en inglés *Common Gateway Interface*) es una tecnología de la World Wide Web que permite a un cliente (navegador web) solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa cuyo resultado final de la ejecución son objetos MIME. Las aplicaciones que se ejecutan en el servidor reciben el nombre de CGI.

Consulta: Información a obtener de acuerdo a criterios específicos.

CSS: Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

Diccionario de datos (DD): conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa, incluyendo nombre, descripción, alias, contenido y organización.

Dimensión: Categoría que agrupa a las distintas preguntas a considerarse en una evaluación docente.

DOM: *Document Object Model* (traducido al español, pero *Modelo en Objetos para la representación de Documentos* o también *Modelo de Objetos del Documento*). Interfaz de programación de aplicaciones que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos.

Etiqueta: marca con tipo que delimita una región en los lenguajes basados en XML.

ExtJS: Es una biblioteca de JavaScript para el desarrollo de aplicaciones web interactivas usando tecnologías como AJAX, DHTML y DOM. Fue desarrollada por Sencha.

Framework: estructura conceptual y tecnológica de soporte definida, normalmente con artefactos o módulos de software concretos, en base a la cual otro proyecto de software puede ser organizado y desarrollado.

Hibernate: es una herramienta de Mapeo objeto-relacional (ORM) para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Hiperenlace: (*también llamado enlace, vínculo, o hipervínculo*). Elemento de un documento electrónico que hace referencia a otro recurso, por ejemplo, otro documento o un punto específico del mismo o de otro documento.

Hipertexto: nombre que recibe el texto que en la pantalla de un dispositivo electrónico conduce a otro texto relacionado. La forma más habitual de hipertexto en documentos es la de hipervínculos o referencias cruzadas automáticas que van a otros documentos.

HTML: siglas de **HyperText Markup Language** (*Lenguaje de Marcado de Hipertexto*), es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes.

HTTP: HyperText Transfer Protocol o **HTTP** (en español *protocolo de transferencia de hipertexto*). Protocolo usado en cada transacción de la World Wide Web (WWW). HTTP define la sintaxis y la semántica que utilizan los elementos de software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

HQL: El HQL (Hibernate Query Language) es un lenguaje de interrogación. En el mundo relacional disponemos del SQL (Structured Query Language) que nos permite obtener información haciendo preguntas basadas en las tablas y sus columnas. El equivalente en el mundo objetual es el HQL, que nos permite hacer preguntas basadas en los objetos y sus propiedades.

Interfaz de usuario: medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, y comprende todos los puntos de contacto entre el usuario y el equipo, normalmente suelen ser fáciles de entender y fáciles de accionar.

Instancia: la palabra **Instancia** significa: Solicitud o Insistencia. Una **instancia** de un programa es una copia de una versión ejecutable del programa que ha sido escrito en la memoria del computador.

Javascript: lenguaje de scripting basado en objetos, utilizado para acceder a objetos en aplicaciones. Principalmente, se utiliza integrado en un navegador web permitiendo el desarrollo de interfaces de usuario mejoradas y páginas web dinámicas.

Jquery: biblioteca de Javascript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

Lógica de control: conjunto de operaciones lógicas y estructuras de control que determinan el orden de ejecución de las instrucciones de un programa.

Mantenimiento: Operación mediante la cual se realiza ingreso de datos nuevos, y la modificación, actualización o eliminación de datos existentes.

Modelo Relacional El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicados y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

Modularidad: Capacidad que tiene un sistema de ser estudiado, visto o entendido como la unión de varias partes que interactúan entre sí y que trabajan para alcanzar un objetivo común, realizando cada una de ellas una tarea necesaria para la consecución de dicho objetivo.

Multiplataforma: Término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de software, que puedan funcionar en diversas plataformas.

MVC (Modelo Vista Controlador). Estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

Objeto: Unidad que en tiempo de ejecución realiza las tareas de un programa. También a un nivel más básico se define como la instancia de una clase.

Plataforma: Arquitectura de computador o de procesador.

Netbeans: Es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender

el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

POO: Siglas de Programación Orientada a Objetos. Está basado en varias técnicas, incluyendo herencia, abstracción, polimorfismo y encapsulamiento.

PostgreSQL: Sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD (*Berkeley Software Distribution*).

Pregunta: Interrogantes que serán tomadas en cuenta y mostradas en un formulario y planteadas a los estudiantes en la realización de una evaluación docente.

Proceso: (en nuestro Sistema SEDUES) Operación mediante la cual se planifica, activa y desarrolla una evaluación docente.

Programa: es un conjunto de instrucciones que una vez ejecutadas realizarán una o varias tareas en una computadora.

Query: cadena de consulta, este término generalmente se utiliza para hacer referencia a una interacción con una base de datos. Es la parte de una URL que contiene los datos que deben pasar a aplicaciones web como los programas CGI.

Reporte: representación de resultados.

Servidor: computadora que, formando parte de una red, provee servicios a otras computadoras denominadas clientes.

Servidor web: programa que está diseñado para transferir hipertextos, páginas web o páginas HTML (HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música. El programa implementa el *protocolo HTTP (HyperText Transfer Protocol)* que pertenece a la capa de aplicación del modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

Script: (*archivo de órdenes o archivo de procesamiento por lotes*). Programa usualmente simple, que generalmente se almacena en un archivo de texto plano. Los script son casi siempre interpretados, pero no todo programa interpretado es considerado un script. El uso habitual de los scripts es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario. Por este uso es frecuente que los shells sean a la vez intérpretes de este tipo de programas.

Trigger: (o disparador) en una base de datos, es un procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación de inserción (INSERT), actualización (UPDATE) o borrado (DELETE).

URL: (Uniform Resource Locator - Localizador Uniforme de Recursos). Forma de organizar la información en la web.

Dirección que permite acceder a un archivo o recurso como páginas html, php, asp, o archivos gif, jpg, etc. Se trata de una cadena de caracteres que identifica cada recurso disponible en la WWW.

XML: siglas en inglés de *Extensible Markup Language* (lenguaje de marcas extensible). Tecnología que permite compartir los datos con los que se trabaja a todos los niveles, por todas las aplicaciones y soportes. Permite al programador y los soportes dedicar sus esfuerzos a las tareas importantes cuando trabaja con los datos, ya que algunas tareas tediosas como la validación de estos o el recorrido de las estructuras corre a cargo del lenguaje y está especificado por el estándar, de modo que el programador no tiene que preocuparse por ello.

WWW: World Wide Web, (*Red Global Mundial o "Red de Amplitud Mundial"*). Sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet. Con un navegador web, un usuario visualiza sitios web compuestos de páginas web que pueden contener texto, imágenes, videos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

1.7 Formulación de la Hipótesis y Variables

1.7.1 Hipótesis general

La falta de una herramienta estándar e integrada al resto de procesos, con mayor disponibilidad para el manejo de datos y el registro de los mismos en cuanto al proceso de calificación, asistencia, gestión de convivencia, origina que dichos procesos no se desarrollen de una forma óptima.

1.7.2 Hipótesis particulares

El procesamiento actual de la información se realiza de forma local, es decir, la disponibilidad se limita a los equipos donde está instalada la aplicación.

La forma en que se procesan los datos puede mejorar si se logra estandarizar e integrar la herramienta de calificación con el resto de departamentos.

La duración del procesamiento actual de información alarga los tiempos de repuesta para los requerimientos recibidos.

El registro de calificaciones en papel pre-impreso y luego la entrega de este reporte a secretaría para su respectivo ingreso en el sistema amplía el tiempo necesario.

Dependiendo de la cantidad de alumnos o ítems a calificar por alumno, ya sea por calificación, convivencia, asistencia. Se genera gran demanda por el uso de papel.

El gestor de base de datos ofrece consistencia sobre la información que se trabaja o la cual se ingresa.

El hecho de tener la información digitalizada brinda disponibilidad y consistencia.

El proceso de respaldo de forma automatizada viene incluido como una opción en la herramienta de gestor de la base de datos, mientras que el respaldo manual tendría que hacerse con alguna otra herramienta/infraestructura.

1.7.3 Matriz Causa – Efecto

Problema general	Objetivo general	Hipótesis general
¿Qué aspectos están incidiendo para que el proceso de calificación, gestión de convivencia y asistencia no sean óptimos?	Identificar los problemas y posibles causas que originan que los procesos de calificación, gestión de convivencia y asistencia no se desarrollen de una forma óptima.	La falta de una herramienta estándar e integrada al resto de procesos, con mayor disponibilidad para el manejo de datos y el registro de los mismos en cuanto al proceso de calificación, asistencia, gestión de convivencia, origina que dichos procesos no se desarrollen de una forma óptima.
Problemas específicos	Objetivos específicos	Hipótesis particulares
¿En qué incide la forma de acceso en cuanto al proceso de registro, modificación y consulta de calificación?	Identificar en que magnitud afecta la forma de acceso a los procesos de registro, modificación y consulta de calificación.	El procesamiento actual de la información se realiza de forma local, es decir, la disponibilidad se limita a los equipos donde está instalada la aplicación.
¿En qué medida afecta que la herramienta utilizada para los procesos de calificación funcione de forma descentralizada?	Determinar las desventajas que se tienen como resultado cuando se utiliza una herramienta descentralizada.	La forma en que se procesan los datos puede mejorar si se logra estandarizar e integrar la herramienta de calificación con el resto de departamentos.
¿Qué incidencia tiene en la duración el llevar el procesamiento de calificaciones, gestión de convivencia y asistencia de la forma actual?	Determinar la incidencia que tiene en la duración el llevar el procesamiento de información de la manera actual.	La duración del procesamiento actual de información alarga los tiempos de respuesta para los requerimientos recibidos.
El registro de calificación en el sistema a través de una sola persona incide en la cantidad de tiempo necesario para realizar el registro.	Identificar las causas que justifiquen que la distribución de trabajo brinda mayor productividad.	El registro de calificaciones en papel pre-impreso y luego la entrega de este reporte a secretaría para su respectivo ingreso en el sistema amplia el tiempo necesario.

¿Qué impacto ecológico tiene el uso de recursos como papel a gran volumen para el registro de calificación, convivencia y asistencia?	Determinar el impacto negativo que afecta ecológicamente el uso de papel a gran volumen.	Dependiendo de la cantidad de alumnos o ítems a calificar por alumno, ya sea por calificación, convivencia, asistencia. Se genera gran demanda por el uso de papel.
¿Qué beneficio existe al almacenar la información correspondiente de registro de calificaciones en un sistema y no en un repositorio físico?	Determinar si es más seguro la metodología de almacenamiento en una base de datos que en archivadores.	El gestor de base de datos ofrece consistencia sobre la información que se trabaja o la cual se ingresa.
¿Qué ventajas existe en tener la información de forma digital a tenerla de forma física?	Validar las ventajas y beneficios de tener la información digitalizada versus física.	El hecho de tener la información digitalizada brinda disponibilidad y consistencia.
¿Cuál es el costo de llevar a cabo el proceso de respaldo manual versus el respaldo automático?	Determinar que es más costoso en cuanto a realizar respaldos de la información en un sistema manual y otro automatizado.	El proceso de respaldo de forma automatizada viene incluido como una opción en la herramienta de gestor de la base de datos, mientras que el respaldo manual tendría que hacerse con alguna otra herramienta/infraestructura.
¿Hay compatibilidad entre un sistema automatizado con diferentes plataformas de sistemas operativos?	Validar si un sistema automatizado brinda amplia funciones de compatibilidad en varios sistemas operativos.	Contar con una herramienta automatizada brinda ciertos beneficios entre ellos la compatibilidad que garantiza funcionalidad en entornos comunes.

Tabla #6: Matriz causa efecto

Fuente: Los autores

1.7.4 Variables

Una variable es un valor que esta propenso al cambio, por lo general son datos numéricos aunque también pueden ser valores de otro tipo como nombre, esto valores pueden cambiar de una forma delimitada.

En el campo de la investigación existen dos tipos de variables y las mismas se han identificado en nuestra solución propuesta de la siguiente forma:

Variable dependiente:

- Tiempo de procesamiento de los datos y accesibilidad a la herramienta.
- Promedios por período.
- Promedio anual.
- Justificación por Falta o Atraso.
- Promedio de Grado.

Variable independiente:

- Proceso de registro de calificación.
- Calificaciones correspondientes a un período.
- Calificaciones de promedio por período.
- Registro manual de un atraso o una falta para un alumno.
- Registro de Exámenes de grado.

1.8 Aspectos Metodológicos de la Investigación

La metodología con la cual se trabaja en el proyecto es la de campo, pues demanda el proceso de relacionarse en el entorno donde surge la necesidad por parte del usuario, el investigador obtiene las respuestas de las fuentes más relevante y analizadas ya sea tomada de forma individual o grupal.

1.8.1 Tipo de Estudio

Tipos de Investigación

Los tipos de investigación aplicados son: de campo, explicativa, no experimental y transversal.

- **Investigación de campo:** Nos permitió la observación en contacto directo con el objeto de estudio, y el acopio de testimonios que permitan confrontar la teoría con la práctica en la búsqueda de la verdad objetiva.
- **Investigación explicativa:** Nos permitió centrarnos en determinar los orígenes o las causas de un determinado conjunto de fenómenos, donde el objetivo es conocer por que suceden ciertos hechos a través de la delimitación de las relaciones causales existentes o, al menos, de las condiciones en que ellas producen. Este es el tipo de investigación que mas profundiza nuestro conocimiento de la realidad, por qué nos explica la razón, el por qué de las cosas, y es por lo tanto más complejo y delicado pues el riesgo de cometer errores aumenta considerablemente.
- **Tipo de investigación no experimental:** Es no experimental porque los sujetos son observados en su contexto natural, en su realidad cotidiana tal y como ellos se comportan naturalmente, ya que las variables independientes ya han ocurrido y el investigador no tiene control directo sobre dichas variables, no puede influir sobre ellas porque ya sucedieron, al igual que sus efectos.

1.8.2 Método de investigación

Para la ejecución del presente proyecto utilizaré el método inductivo – deductivo.

Así mismo aplicaré las siguientes técnicas de investigación: Observación directa, entrevista, y cuestionario.

1.8.2.1 Método inductivo – deductivo

Inducción: Es un método que parte de lo particular hacia un todo, algo general, este método permite generar hipótesis en base a lo estudiado.

Inducir ha permitido generalizar los estudios en distintos campos como: Física, Química, biología, etc. Las ciencias mencionadas se basan en el estudio u observación de un fenómeno en particular permitiendo moldear procedimientos para aplicarlos de forma general.

Deducción: Es un método que parte de lo general hacia algo particular, luego somete. Pese a que el razonamiento deductivo es una maravillosa herramienta del conocimiento científico, si el avance de la ciencia se diera sólo en función de él, éste sería muy pequeño. Esto se debe a que nuestra experiencia como humanos es limitada, depende de nuestros sentidos y de nuestra memoria.

Centrándonos en el deductivismo, se trata de un procedimiento que consiste en desarrollar una teoría empezando por formular sus puntos de partida o hipótesis básicas y deduciendo luego sus consecuencias con la ayuda de las subyacentes teorías formales. Sus partidarios señalan que toda explicación verdaderamente científica tendrá la misma estructura lógica, estará basada en una ley universal, junto a ésta, aparecen una serie de condicionantes iniciales o premisas, de las cuales se deducen las afirmaciones sobre el fenómeno que se quiere explicar.

La inducción y la deducción no son formas diferentes de razonamiento, son formas de inferencia.

Cuando usamos simultáneamente los métodos de inferencia inductiva y deductiva para buscar la solución de un problema científico decimos que estamos empleando el método inductivo–deductivo, cuyas reglas básicas de operación son:

- a) Observar cómo se asocian ciertos fenómenos, aparentemente ajenos entre sí.
- b) Por medio del razonamiento inductivo, intentar descubrir el denominador común (ley o principios) que los asocia a todos.
- c) Tomando como punto de partida este denominador común (por inducción), generar un conjunto de hipótesis referidas a los fenómenos diferentes, de los que se partió inicialmente.
- d) Planteadas las hipótesis, deducir sus consecuencias con respecto a los fenómenos considerados.
- e) Hacer investigaciones (teóricas o experimentales) para observar si las consecuencias de las hipótesis son verificadas por los hechos.

Grupo Planeación Estratégica, 2009. “Método Inductivo y Deductivo”.

<http://planeacionestrategica.blogspot.es/1236115440>

1.8.3 Fuentes y técnicas para la recolección de información

Para el proyecto en mención haré uso de la entrevista como método de recolección de información, y la misma que va dirigida a personal administrativo y docentes de la institución:

- Padre Enrique Aponte, Director Administrativo.
- Ec. Rosa Bermeo, Secretaria.
- Ingeniero Alvaro Rivera, Docente.

Los métodos de recolección para la información se basan en la entrevista con preguntas abiertas o cerradas al departamento de secretaria y alguno docentes cuyo factor común es el uso de recurso de papel para el registro y control de notas.

La entrevista

En el marco teórico consta la funcionalidad de la entrevista, para recordar podemos mencionar que es un dialogo establecido por dos partes que son: Entrevistado y Entrevistador, toda entrevista tiene un propósito específico de ahí parte la importancia de su uso para distintos campos como: periodismo, investigación, proyectos sociales, y para este caso tesis.

Para lo antes mencionado existen técnicas que facilitan la recolección de información al momento de realizar la entrevista. Más detalles acerca de esta técnica para recolección de información, se encuentran en el **marco teórico** (*sección 1.5.2.5*).

1.9 Resultados Esperados

El resultado que se espera obtener al desarrollar e implementar el proyecto en el cual se automatiza y se integran los módulos de calificación, de gestión de supervivencia y de asistencia, es de optimizar el trabajo, mejorar la productividad, el acceso para los usuarios, en este caso docentes y alumnos que ingresen al módulo correspondiente con el debido acceso, este método garantiza el funcionamiento y accesibilidad de una manera interactiva e intuitiva.

Adicionalmente se planea obtener un gran impacto en el ahorro de recurso renovable como el papel, ya que el registro de calificaciones, asistencia y evaluación se hará en la web, de esta forma la información será filtrada y depurada para la generación de reportes.

CAPÍTULO 2

2. ANÁLISIS, PRESENTACIÓN DE RESULTADOS Y DIAGNÓSTICOS

2.1 Análisis de la situación actual

2.1.1 Biografía San Antonio María Claret

San Antonio María Claret y Clara (Sallent, 23 de diciembre de 1807 - Abadía de Fontfroide, 24 de octubre de 1870), Español, Arzobispo de Santiago de Cuba y confesor de la reina Isabel II de Borbón. Fundador de la Congregación religiosa católica de los Misioneros Hijos del Inmaculado Corazón de María (Misioneros Claretianos) (Cordis Mariae Filii -C.M.F.-) el 16 de Julio de 1849 y, con la Venerable María Antonia París de San Pedro, de la Congregación de las Religiosas de María Inmaculada Misioneras Claretianas el 27 de agosto de 1855. Es patrono de estas congregaciones, los claretianos, co-patrón de la Diócesis de Canarias¹ y de los tejedores.

Quinto hijo de Juan Claret y de Josefa Clara, es bautizado como Antonio, y solamente más tarde, en su ordenación episcopal, incluirá el nombre de María en el suyo por devoción a la madre de Jesucristo. San Antonio María Claret iba con frecuencia con su hermana Rosa a llevarle flores a una capilla. Trabajó en los telares de su padre desde los doce años. Ya con diecisiete, su padre lo envía a la escuela Comercial de La Lonja en Barcelona para apoyar su carrera como industrial textil. A pesar de su prometedor futuro, después de cuatro años regresa a Sallent, su pueblo natal, e ingresa en el seminario de Vic con 22 años. Su primera intención es hacerse cartujo, pero no la llevará a cabo. Se ordenó finalmente sacerdote en 1835 en Solsona. En 1839 viaja a Roma con la intención de ingresar en la Propaganda Fide (Propagación de la Fe) y prepararse para convertirse en misionero, pero un año después regresa a España por motivos de salud.

De nuevo en Cataluña, se le confía la parroquia de Viladrau. Viajará mucho por las tierras catalanas entre 1843 y 1847 hasta que es enviado por el obispado a Canarias. En 1848 fundó la Librería Religiosa, a través de la cual intenta luchar a favor de la fidelidad a la religión católica dentro del país. En 1849 regresa a Vic, donde funda la Congregación de los Misioneros Hijos del Inmaculado Corazón de María junto

con los sacerdotes Esteve Sala, Josep Xifré, Manuel Vilaró, Domenec Fábregas y Jaume Clotet y casi inmediatamente recibe la notificación de su nombramiento como arzobispo de Santiago de Cuba, el 20 de mayo de 1850, a donde se traslada un año después. Fue consagrado el 6 de octubre de 1850 por Lluçia Casadevall Duran, obispo de Vic, actuando como co-consagrantes Josep Domenec Costa i Borràs, obispo de Barcelona, y Florencio Costa y Montón, obispo de Gerona.

En Cuba, con la madre María Antonia París, fundó el Inmaculado Corazon de María en 1855. En 1856 sufre un atentado durante una visita a Holguín y un año más tarde vuelve a España al ser nombrado confesor de la reina Isabel II. La reina lo elige también como protector de la iglesia y del hospital de Montserrat de Madrid, y en 1859 Presidente de El Escorial. Además ostentará el título, un tanto honorífico, de arzobispo de Trajanópolis in partibus infidelium (en país de infieles).

Con motivo de la revolución de 1868 tiene que exiliarse a París junto con la reina, donde funda las Conferencias de la Sagrada Familia.

En 1869 participa en la preparación del Concilio Vaticano I, en el que interviene defendiendo la infalibilidad pontificia. Posteriormente se traslada a la comunidad que sus misioneros tienen en Prades (Francia), pero tendrá que refugiarse en la abadía de Fontfroide al ser perseguido por motivo de sus vínculos políticos con la corte de Isabel II. Allí fallece a los 63 años, el 24 de octubre de 1870. Sus restos mortales se trasladaron a Vic en 1897. Es beatificado por Pío XI el 25 de febrero de 1934 y canonizado por Pío XII el 7 de mayo de 1950.

Fuente: http://es.wikipedia.org/wiki/Antonio_Mar%C3%ADa_Claret

2.1.2 Reseña Histórica de la unidad educativa claretiana

La historia de los misioneros claretianos se remonta al año 1955 cuando Mons. César Mosquera pidió al Superior General de la Congregación Claretiana, que se hiciera una fundación de la comunidad religiosa en este país. Con esta petición se abre para los Misioneros Claretianos un abanico de posibilidades para aportar en la construcción de una sociedad más justa e igualitaria.

La educación católica es prioridad de la obra claretiana que desde hace más de medio siglo permanece dentro de los ecuatorianos. La labor educativa desempeñada por los misioneros claretianos se ve reflejada a lo largo de estos años en diferentes etapas, a saber:

✓ **Escuela Espíritu Santo:**

Mediante acuerdo Ministerial N° 21 Se inicia en el año 1956 la Escuela Espíritu Santo, con el N° 93 y funcionaba en las calles Venezuela y Octava. Conto con seis grados de instrucción primaria.

✓ **Colegio Cristo del Consuelo:**

Mediante acuerdo Ministerial No. 529 en 1.961 se inicia el Colegio “Santísimo Cristo del Consuelo” con el primer, segundo y tercer curso del ciclo básico el cual funcionó hasta 1.980.

El Colegio Cristo del Consuelo contaba con tres cursos, el número de estudiantes no sobrepasaban los cien y se laboraba por las tardes; con el pasar del tiempo, se vio necesario que la jornada de trabajo fuera por la mañana debido al peligro nocturno. El colegio fue cerrado temporalmente en el año de 1988.

Es importante recordar que desde el año 1961 hasta nuestros días, la Escuela siempre ha permanecido funcionando sin ninguna interrupción, además con el cierre del colegio en el año 1988 bajo acuerdo ministerial N° 48 la Escuela deja de ser de Varones y pasa a ser Mixta.

2.1.3 Antecedentes

En el 2002 al observar las necesidades de la zona, los Misioneros Claretianos contando con la aprobación de las autoridades educativas respectivas, solicitan la reapertura del colegio y cambia la denominación del Santísimo Cristo del Consuelo a COLEGIO CLARETIANO mediante acuerdo Ministerial No. 047 y 709, ofertando los niveles Primaria, Secundaria y Bachillerato Técnico en Comercio y Administración con las especializaciones: Informática y Contabilidad. Viendo que

escuela y colegio se extendió se decidió constituirlo en una Unidad Educativa, con el nombre Claretiana, Según acuerdo 149-AJ.

Para el año 2004 el colegio ya contaba con la segunda planta para empezar a utilizarse, aunque no estaban en perfectas condiciones, con el tiempo se le hicieron los arreglos respectivos. Para entonces el colegio ya contaba con los cursos que van desde el ciclo básico hasta el segundo año de bachillerato, es decir, el cuarto de propedéutico y el segundo año de bachillerato en las especializaciones de Informática y Contabilidad; es importante recordar que la primera promoción como Unidad Educativa se realiza en el año 2005. Actualmente (2010) se están haciendo los trámites pertinentes para contar con la nueva modalidad de Química y biología (Quibio).

Es importante resaltar que durante más de cincuenta años de labores educativas al servicio de la vida, seguimos haciendo realidad el proyecto de nuestro patrono “San Antonio María Claret”.

2.1.4 Misión

Somos una institución educativa con pedagogía innovadora, reflexiva, crítica y formadora de personas que brinda oportunidad de involucrarse en un proceso que permite crecer en valores humanos, para llegar a desarrollarse en el campo laboral o en la continuación de estudios superiores.

Proyectamos nuestra misión hacia un futuro de calidad en el servicio evangelizador, desde la educación. Esto significa abrir espacios para ir configurando una propuesta alternativa con un modelo antropológico cristiano, el mismo que nos reclama el Dios de la Vida. Los educadores Claretianos realizamos la misión evangelizadora desde las propuestas de pastoral educativa realizadas en misión compartida al estilo de Claret.

Optamos por una educación:

- Abierta a la vida, su defensa y su dignificación, sensible a la dinámica de cambio de nuestras comunidades y capaz de respuesta a las necesidades prioritarias que plantea la realidad, generadora de esperanza, que garantice el

compromiso de parte de los agentes educativos como servicio a la comunidad y al entorno.

- Que garantice la vivencia de los valores del evangelio, la capacidad de gestión, el trabajo en equipo y la organización.

2.1.5 Visión

Presentar una propuesta educativa que permita propiciar y potenciar factores de calidad humana en búsqueda de la formación de niños y jóvenes con valores, conciencia crítica y reflexiva que le permita desenvolverse en todos los ámbitos y en la transformación de su entorno.

Como Institución hemos tenido una Visión global de la realidad y nuestra principal preocupación ha sido saber cómo responder a las urgencias y necesidades de la niñez, la juventud y la familia.

Nuestra propuesta educativa busca responder desde el evangelio a los retos que nos plantea esta realidad, donde el Espíritu del Señor Jesús está presente y nos desafía a soñar, a concretar y a vivir la opción por construir una Comunidad Educativa según el evangelio y al estilo de Claret. Por tanto los educadores Claretianos, soñamos y pensamos en un proyecto evangelizador con rasgos propios que respondan a las necesidades y urgencias del momento actual.

2.1.6 Objetivos de la Unidad Educativa Claretiana

- Una plataforma y base sólida, para la transformación de la persona y de la sociedad para cooperación y aportación con valor humanitario.
- Misionera, sensible a lo urgente, oportuno y eficaz, para acompañar los procesos de formación y maduración de la fe.
- Realizada en misión compartida y con una propuesta clara de formación para los agentes educativos.
- Una educación con sentido y proyección misionera, siempre en camino. Inculturada, que sea abierta a la vida y sensible socialmente, con espacios de sensibilización y práctica solidaria ante las necesidades de las personas.
- Con una mirada crítica de la realidad, con un enfoque y una pedagogía propios y con una propuesta clara de pastoral educativa.
- Comprometida con la vida y la dignificación de la persona.
- Con proyectos alternativos.
- Con procesos de gestión que respondan a las urgencias del contexto y del entorno.

Fuente: <http://www.uec.edu.ec/>

2.1.7 Justificación

Desde hace algunos años atrás las organizaciones comenzaron a necesitar sistemas que apoyaran la gestión empresarial, que integren procesos del negocio, promuevan procesos operativos y sirvieran de soporte para procesos claves para la parte administrativa.

Actualmente se exige que las empresas cuenten con la agilidad, flexibilidad y capacidad para adaptarse a nuevos entornos de una manera eficaz y espontánea. Estas características serán adoptadas solamente, gracias a la tecnología, con la cual, las empresas podrán obtener ventajas competitivas, con el debido soporte y entrenamiento para poder usarlas.

Hoy en día contamos con una economía global y altamente competitiva, que busca posicionarse en el mercado con recursos tecnológicos innovadores que le permitan optimizar y ganar productividad, por lo cual es indispensable que los administradores de las empresas desarrollen estrategias que les ayuden a satisfacer las necesidades de clientes quienes son cada vez más exigentes; anticipándose a sus requerimientos y dándoles un trato personalizado a cada uno de ellos. Actualmente, las compañías buscan implementar sistemas que manejen todas las áreas del negocio, es decir, tener un control estándar y alineado con el objetivo de la empresa. Se ha investigado nuevas herramientas tecnológicas para poder optimizar los procesos operativos para así ahorrar costos y ser más eficientes, lo que tiene como consecuencia un mejor posicionamiento y la atracción o bien conservación de clientes.

Hoy en día contamos con herramientas para cubrir necesidades globales y personalizadas, teniendo como pilar el uso de las Tecnologías de Información y Comunicación, con base en el impacto que ha demostrado su aprovechamiento principalmente como parte del sector privado, desde el uso de sistemas para la administración interna de las organizaciones con el uso del Enterprise Resource Planning (ERP), con el cual se ha conseguido aprovechar mejor los recursos y obtener grandes ahorros en costo.

La herramienta a nivel de ERP (Planeación de Recursos Empresariales) es una manera de utilizar la información a través de la organización de forma proactiva en las diferentes áreas tales como compras, administración de inventario, manufactura y suministros, control financiero, administración de talento humano, logística y distribución, ventas, marketing y manejo de relaciones con clientes. Su finalidad primordial consiste en unir de forma íntegra estos elementos, y proporcionarle al usuario una manera estándar de acceder y utilizar la información en el momento que se necesite, entonces hablamos de disponibilidad como característica, consistencia e integridad de la información como ventaja.

Con un sistema integrado a través de interfaces estandarizadas se logra romper las barreras de información entre los diferentes sistemas y departamentos y mantener comunicación de forma unificada.

Los tipos de soluciones ERP frecuentemente implementadas:

Propietario.-Los sistemas propietarios son aquellos que requieren del pago de una licencia para poder ser utilizados.

Opensource.- Una buena alternativa a los ERP propietario son los sistemas ERP Opensource o de software libre. Aunque se tiende a pensar: “si es libre es gratis”, esto no es del todo cierto. Las empresas desarrolladoras de este tipo de sistemas suelen tener una comunidad de “partners” que ofrecen servicios de implantación, configuración, parametrización y formación de usuarios en sus aplicaciones ERP. Los proyectos Opensource se basan en la entrega y garantía de libertades al usuario final. El Software Libre es el que da:

Modalidad SaaS.-La nueva tendencia en el mercado de sistemas ERP es el software como servicio o SaaS. Se trata de un modelo de entrega de software donde la compañía que implanta proporciona mantenimiento, operación técnica diaria, y la ayuda para el software proporcionado a su cliente. El cliente tiene el sistema hospedado en la compañía de IT.

En Ecuador las principales empresas que implementan soluciones ERP son las industrias manufactureras las cuales debido a su gran número de procesos y transacciones se ven en la necesidad de contar con un sistema integral.

El actual Gobierno reconoce los beneficios que brinda el uso de Tecnologías de Información y Comunicaciones como factor clave para el desarrollo del país, para ello existe un incremento en el presupuesto del estado destinado a mejoras en el sector educativo que seguramente contara con un buen soporte tecnológico.

En nuestro país dentro del sector educativo son pocas las instituciones que cuentan con sistemas integrados, por lo general las particulares. Las instituciones mayormente poseen sistemas independientes por lo cual no se aprecia la información en un flujo continuo.

Las instituciones educativas poseen diversas áreas o departamentos, en donde cada uno posee sus propios procesos, los cuales son efectuados mediante pequeños sistemas poco eficientes y en muchas ocasiones de manera manual. Esto demanda mayor cantidad de tiempo y esfuerzo humano.

Con la implementación del Sistema Integrado de Gestión Académica brindaremos ayuda tecnológica a las instituciones para que estas se anticipen y se adapten a las necesidades de cambio que demanden, todo esto se planea hacer en un tiempo mínimo con el fin de disminuir el impacto y aumentar competitividad.

El Sistema Integrado de Gestión Académica dispondrá de módulos específicos para cubrir las exigencias de cada una de las áreas funcionales de la institución, de tal manera que se cree un flujo de trabajo entre los distintos usuarios. Este flujo permite evitar tareas repetitivas, y mejorará la comunicación en tiempo real entre todas las áreas que integran la entidad educativa.

2.2 Estructura Organizacional

2.2.1 Organigrama

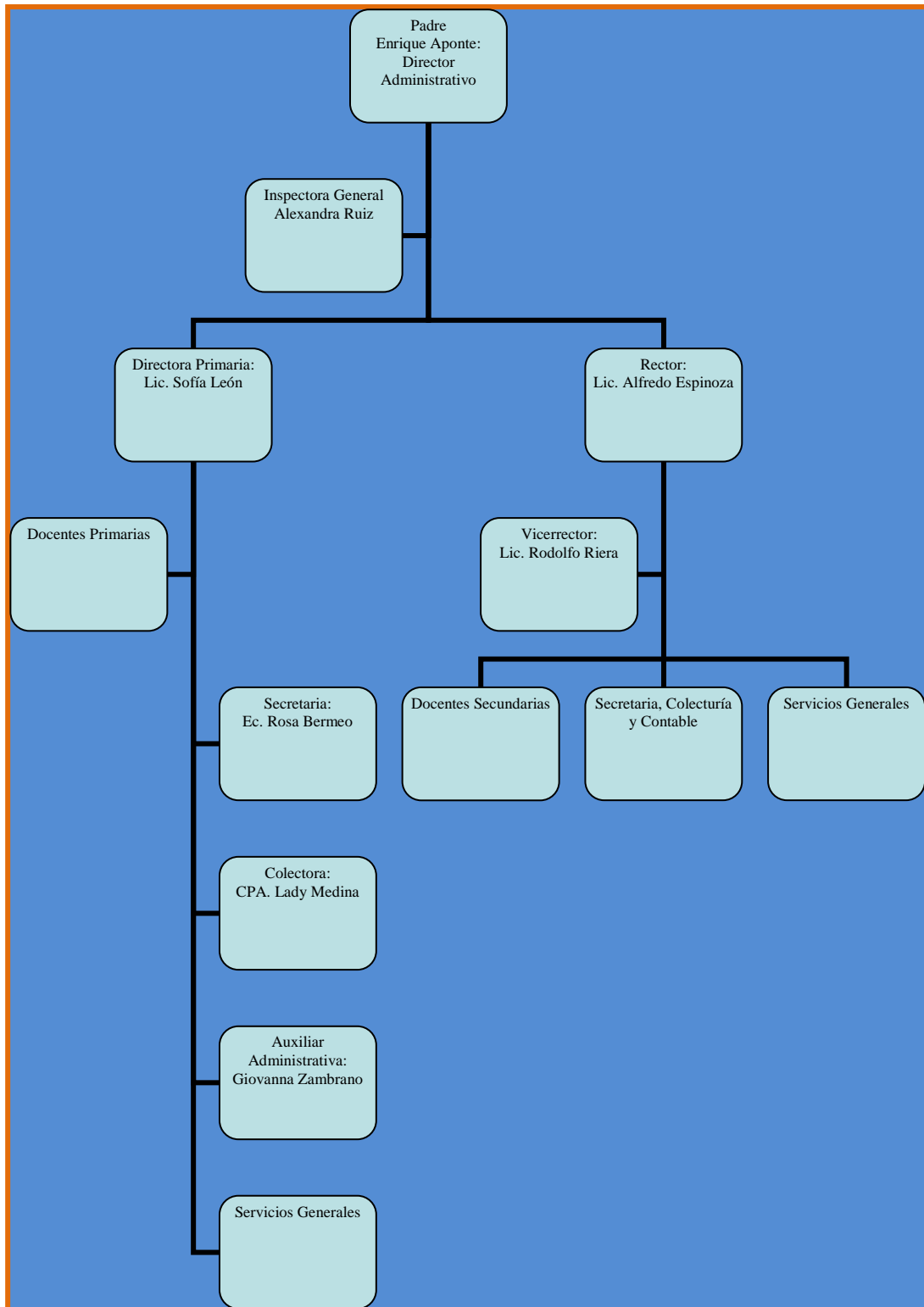


Tabla #7: Organigrama de la institución claretiana.

Fuente: Los autores.

2.2.2 Diagnóstico

En el trabajo realizado de investigación hemos logrado obtener información relevante mediante método de investigación de campo y la entrevista (1.5.2.5).

Hemos realizado la investigación en el lugar donde se origina la necesidad y requerimientos por parte del cliente o usuario, es decir, hemos realizado una investigación de campo en la unidad Educativa Claretiana, el beneficio de realizar este tipo de investigación es que tenemos la percepción directa de las necesidades por parte del usuario para poder mejorar su productividad, así mismo podemos apreciar la infraestructura tecnológica que poseen para poder dimensionar la solución que se propone con el fin que el impacto (Resistencia Usuario) e inversión sean mínimos. Hay que tener en cuenta que la solución planteada cuenta con licenciamiento libre y es multiplataforma.

Realizamos una investigación no experimental porque todo fue observado en su contexto natural, en su realidad cotidiana, ya que la variable independiente que obtuvimos como resultado, el tipo de manipulación de datos (*ver sección 1.7.4*), ya ha ocurrido y nosotros como investigadores no podemos ejercer un efecto de control directo sobre dicha variable, no podemos tampoco influir sobre ella porque ya sucedió al igual que sus efectos, por ejemplo, la demora en la entrega de resultados e informes estadísticos. Adicional a ello contamos con información recopilada de vital importancia para garantizar la funcionalidad de nuestro proyecto.

De una población de cerca de 300 Unidades Educativas, las cuales aun no cuentan con un sistema de gestión académica, pero desean implementarlo, se extrajo una pequeña muestra formada por 30 instituciones educativas entre ellas:

- ✓ Unidad Educativa Fisco misional DOMINGO SAVIO
- ✓ Unidad Educativa ESPIRITU SANTO
- ✓ Unidad Educativa CLARETIANA
- ✓ Escuela Particular femenina LA PRESENTACION
- ✓ Unidad Educativa LICEO CRISTIANO
- ✓ Unidad Educativa FEDERICO GONZALEZ SUAREZ
- ✓ Colegio Particular Mixto CAYETANO TARRUELL
- ✓ Colegio Fiscal Mixto ATI II PILLAHUASO
- ✓ Colegio Particular Mixto GUILLERMO ROHDE
- ✓ Colegio Nacional femenino AURORA ESTRADA
- ✓ Colegio Nacional femenino GUAYAQUIL
- ✓ Instituto Tecnológico SIMON BOLIVAR
- ✓ ESCUELA SUPERIOR POLITECNICA DEL LITORAL
- ✓ UNIVERSIDAD POLITECNICA SALESIANA
- ✓ UNIVERSIDAD DE GUAYAQUIL
- ✓ UNIVERSIDAD LAICA VICENTE ROCAFUERTE
- ✓ UNIVERSIDAD DEL PACIFICO
- ✓ Unidad Educativa SAGRADOS CORAZONES
- ✓ Colegio Fiscal VICENTE ROCAFUERTE
- ✓ Unidad Educativa AGUILAS DE CRISTO
- ✓ Colegio Particular de Señoritas NARCISA DE JESUS MARTILLO MORAN
- ✓ Unidad Educativa SANTIAGO DE LAS PRADERAS

De las cuales representa el 10% de la población solo en la Provincia del Guayas, el criterio de elección fue “conveniencia”.

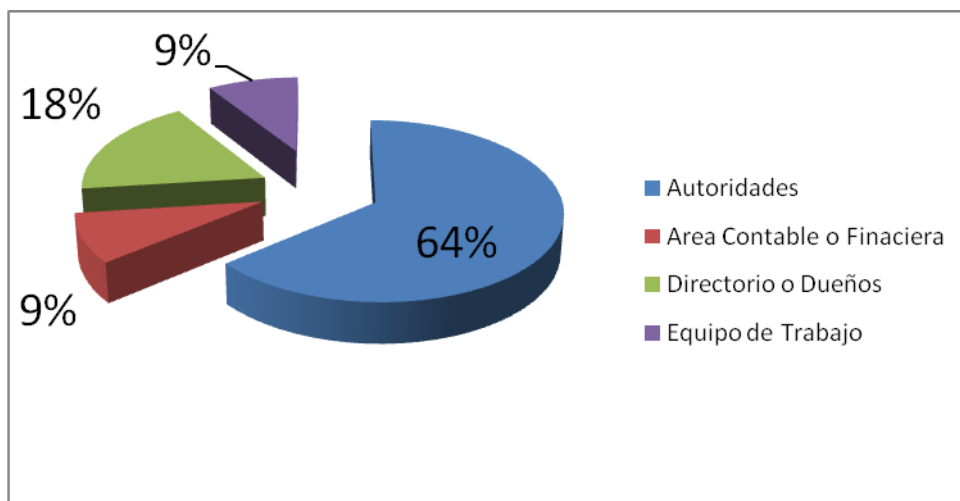


Figura #3: Decisión de implementar un sistema ERP

Fuente: Los autores.

En el gráfico anterior se puede observar que la decisión de implementar un sistema como el nuestro, en el 64% de las Instituciones Educativas analizadas recae sobre las Autoridades; en un 9% sobre el área contable o financiera; en un 18% sobre el directorio o los Dueños; y el 9% restante sobre el equipo de trabajo, formado por un integrante de cada departamento o área de la Institución Educativa.

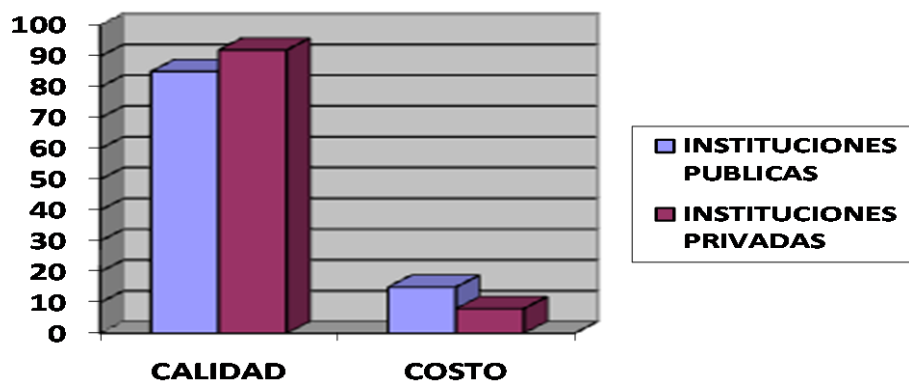


Figura #4: Costo Calidad de Implementación.

Fuente: Los autores.

2.2.3 Análisis Costo/ Calidad en la Implementación

En el grafico anterior se observa que la ventaja económica al momento de implementar la solución propuesta es una ventaja competitiva clara sin alterar la calidad del producto.

Las herramientas utilizadas garantizan estabilidad y flexibilidad adaptándose a plataformas previamente existentes.

Tanto el sector privado como el sector público están apuntando a utilizar herramientas tecnológicas que mejoren su productividad y que aseguren la recuperación de la inversión en poco tiempo.

Grupos Objetivos para la adquisición de este tipo de Software.

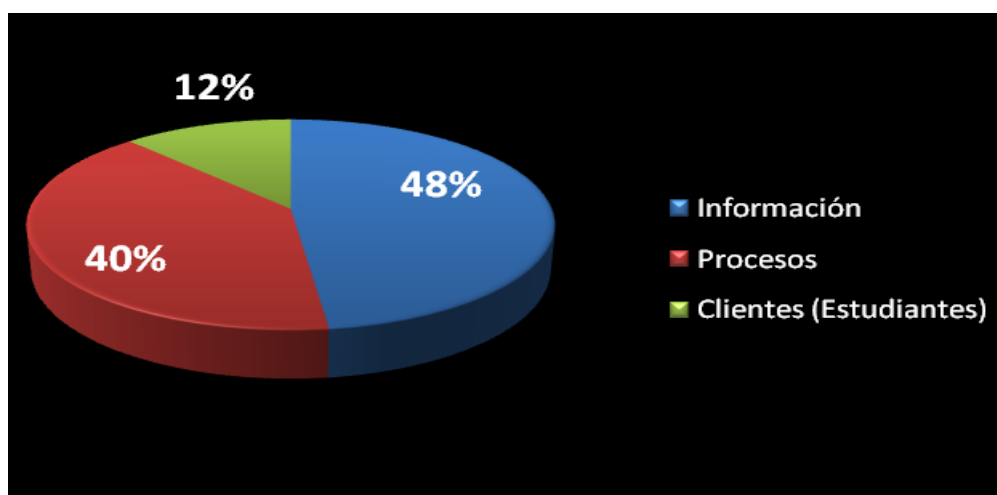


Figura #5: Representación de objetivos que buscan las instituciones.

Fuente: Los autores.

- ✓ En este grafico se distinguen tres grupos objetivos, que todas las Instituciones interesadas en este tipo de Software pretenden lograr después de su implementación, estos grupos son:

1. **Objetivos al nivel de toma de decisiones:** Entendida esta como una administración eficiente de la información: Contable, Financiera y de sus operaciones, (Acceso a información confiable, precisa, oportuna,

posibilidad de compartir información, integridad de los datos y mejorar el proceso de toma de decisiones).

2. **Objetivos al nivel de procesos:** Optimización, reducciones de tiempo y de costo y mejorar productividad.
3. **Objetivos al nivel de sus clientes:** que vendrían a ser los estudiantes quienes demandan eficiencia en las operaciones de las Instituciones y por quienes las entidades educativas mejoran su infraestructura física, organizacional y tecnológica.

✓ En base a los resultados obtenidos por medio de las encuestas realizadas a cada una de las Instituciones, se realizó un análisis más detallado por Grupo Objetivo, mediante la siguiente tabla mostraremos un resultado más detallado.

Razones para adquirir el software	Instituciones educativas encuestadas	Porcentaje que representa
Disponer y acceder a información confiable, precisa y oportuna	5	16.67%
Posibilidad de compartir información entre todas las áreas o departamentos de la Institución	4	13.33%
Eliminación de datos y operaciones innecesarias	2	6.67%
Mejorar el proceso de la toma de decisiones	9	30%
Mejorar el servicio al cliente mediante la optimización de los procesos de la Institución	10	33.33%

Tabla #7: Cuadro descriptivo del estudio realizado en las instituciones.

Fuente: Los autores

- ✓ Los beneficios que esperan la mayoría de las Instituciones a las que entrevistamos es que la integración de la información de los diferentes procesos de la Institución permita aumentar la confianza en los mismos, facilitar el acceso a la bases de datos centralizadas, facilitando el control de las áreas, mejorando el proceso de toma de decisiones, esto, se traducirá en mejorar la productividad al reducir sus costos operativos, sus tiempos de entrega y así aumentando la competitividad de la misma.

2.2.4 Estudio Tecnológico

En la investigación realizada, hemos podido obtener información con la entrevista realizada a la siguiente persona:

- Sr. Alvaro Rivera, Soporte Sistemas y Docente de la Unidad Educativa Claretiana.

En base a la información recopilada con la ayuda del Sr. Rivera hemos podido determinar y plantear las herramientas a utilizar para desarrollar el sistema.

A continuación se mostrarán los distintos criterios del Coordinador de Sistemas de la Unidad Educativa Claretiana, de acuerdo a la entrevista que se le realizó:

✓ Herramienta de Programación

Se escogió Netbeans por la facilidad de integración y administración en cuanto a la arquitectura y diseño de software, modularidad, ya que es la herramienta adecuada para proyectos de gran magnitud y alcance. Adicional a ello la licencia es Open source.

Esta herramienta de programación es multiplataforma, es decir, no requiere de un sistema operativo en especial para ejecutarse, dando facilidad para que pueda ejecutarse sobre el entorno Window, Linux, ext.

✓ **Motor de Base de Datos**

El motor de la base datos que se utilizara de acuerdo a la propuesta realizada y a la respectiva decisión del coordinador de sistemas es Postgres SQL, ya que es un potente gestor de base de datos con amplia cobertura y funcionalidad y al igual que la herramienta de programación también es Open Source.

✓ **Infraestructura Requerida**

Debido a que los sistemas a utilizar no demandan una infraestructura en especial para su respectivo funcionamiento, los servidores deben considerarse con infraestructura estándar, es decir, no se requiere de alguna tecnología en especial.

Respecto a las licencias de los sistemas operativos, pues queda en potestad de la institución que usará la herramienta ya que como mencionamos anteriormente la herramienta de programación y el gestor de la base de datos son multiplataforma. El desempeño de las funciones que brinda la aplicación no se ven afectados por los detalles mencionados.

El análisis de los resultados de las preguntas realizadas al Coordinador de Sistemas de la Unidad Claretiana, permiten justificar el uso de herramientas de diseño propuestas.

De acuerdo a lo que precisó la persona entrevistada, obtuvimos lo siguiente:

Con estos resultados podemos justificar el uso de las herramientas propuestas para realizar el diseño e implementación de nuestro proyecto, para que así se ejecuten en cualquier tipo de servidor, sea éste Windows o Linux. Estas herramientas son:

- Netbeans: Lenguaje de programación.
- PostgreSQL: Manejador de bases de datos.

CAPÍTULO 3

3. PROPUESTA DE CREACIÓN

3.1 Modelo de Procesos

3.1.1 Procedimiento de listado, ingreso, y selección de dimensiones



Figuras #6: Diagrama de Procedimiento de Ingreso, listado y selección de dimensiones.

Fuente: Los autores.

3.1.2 Registros de Atrasos



Figura #7: Diagrama de registro de atrasos por alumno

Fuente: Los autores.

3.1.3 Registro de Faltas

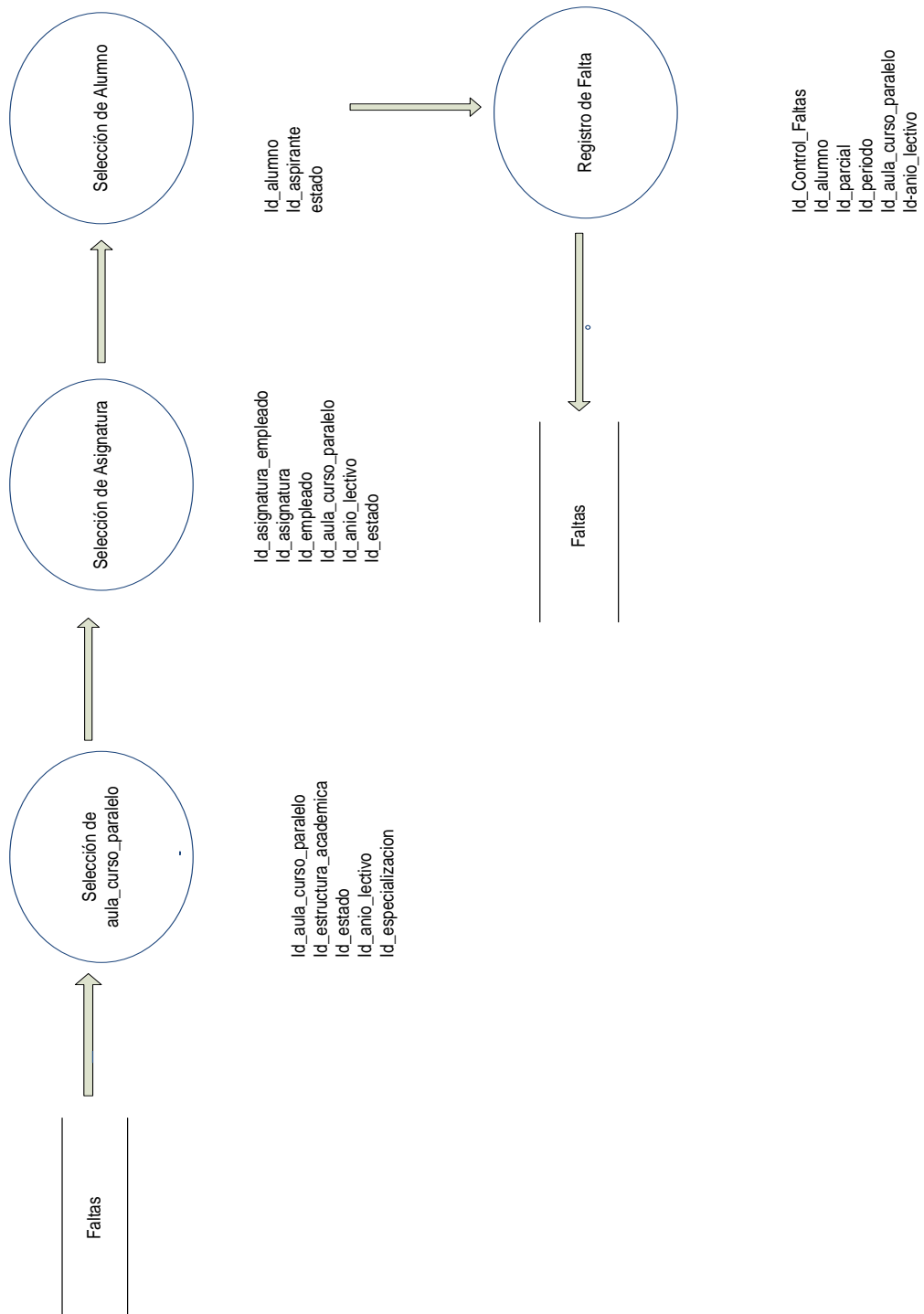


Figura #8: Diagrama de registro de faltas por alumno

Fuente: Los autores.

3.1.4 Control de Justificación de Faltas

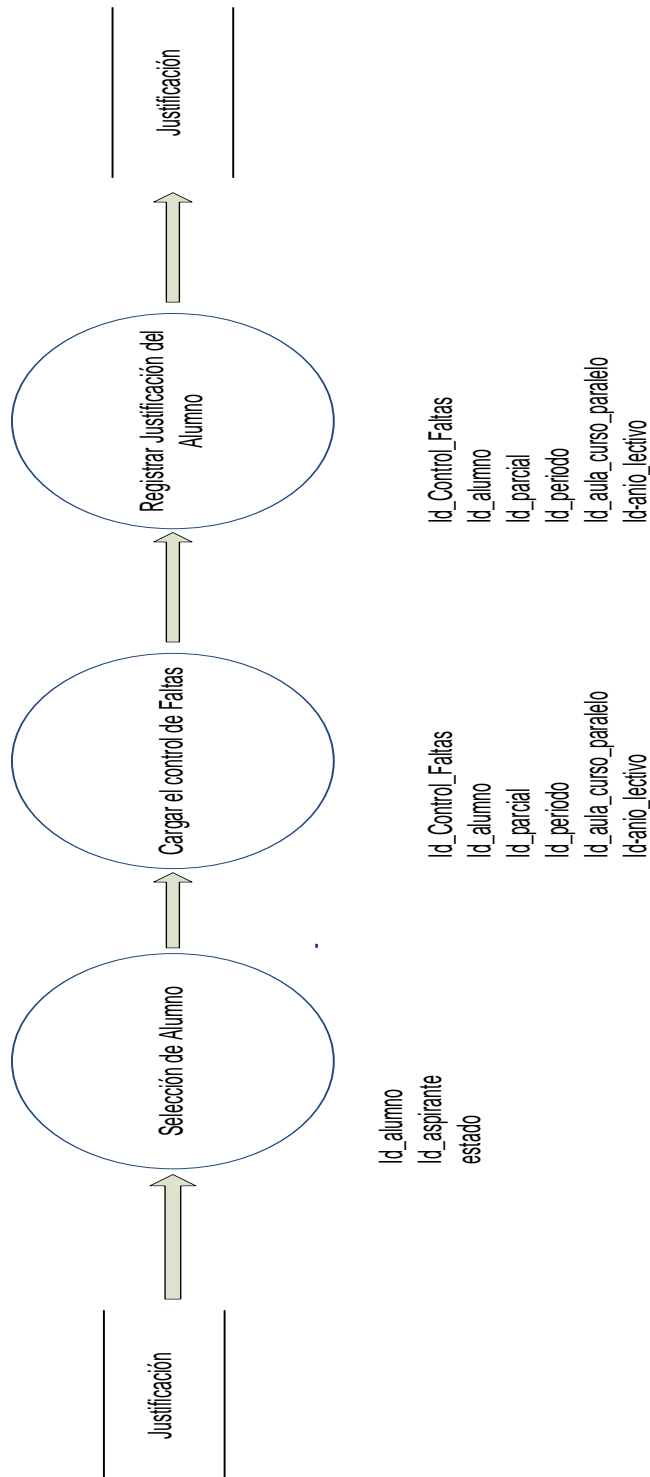


Figura #9: Diagrama de justificación de faltas por alumno

Fuente: Los autores.

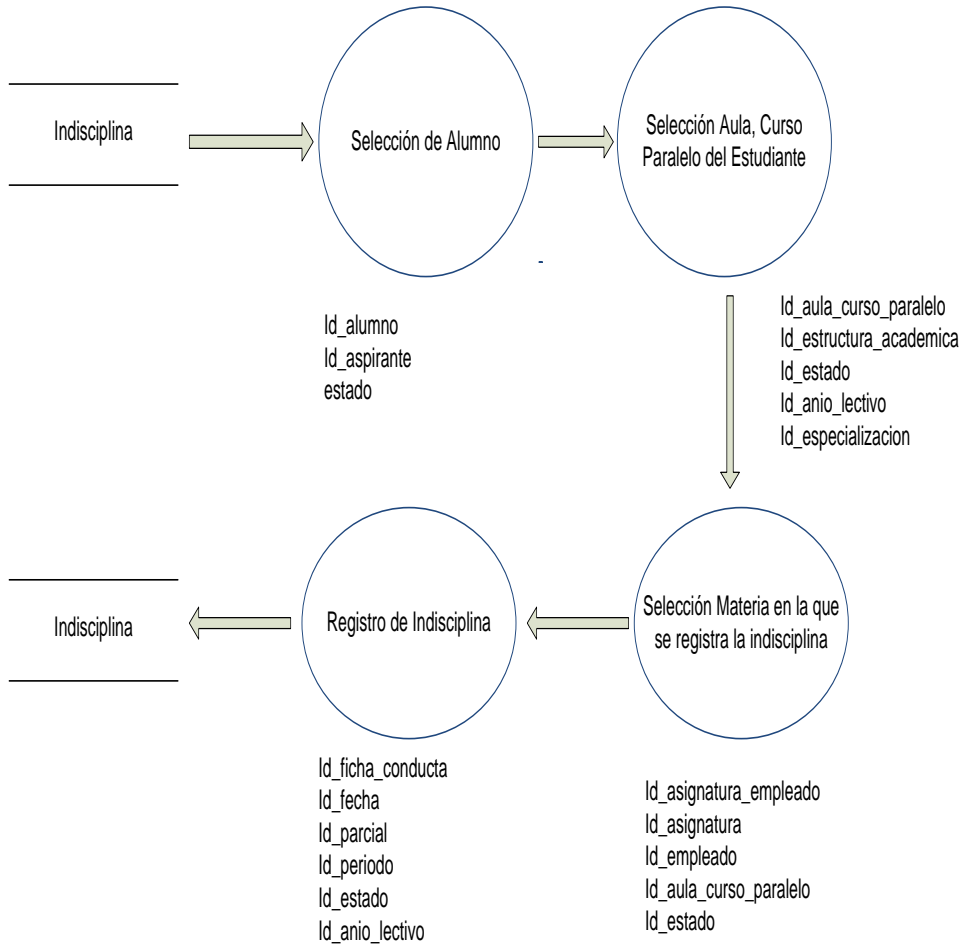


Figura #10: Diagrama de indisciplina de faltas por alumno, materia y curso

Fuente: Los autores.

3.1.5 Registro de Indisciplina de Estudiante

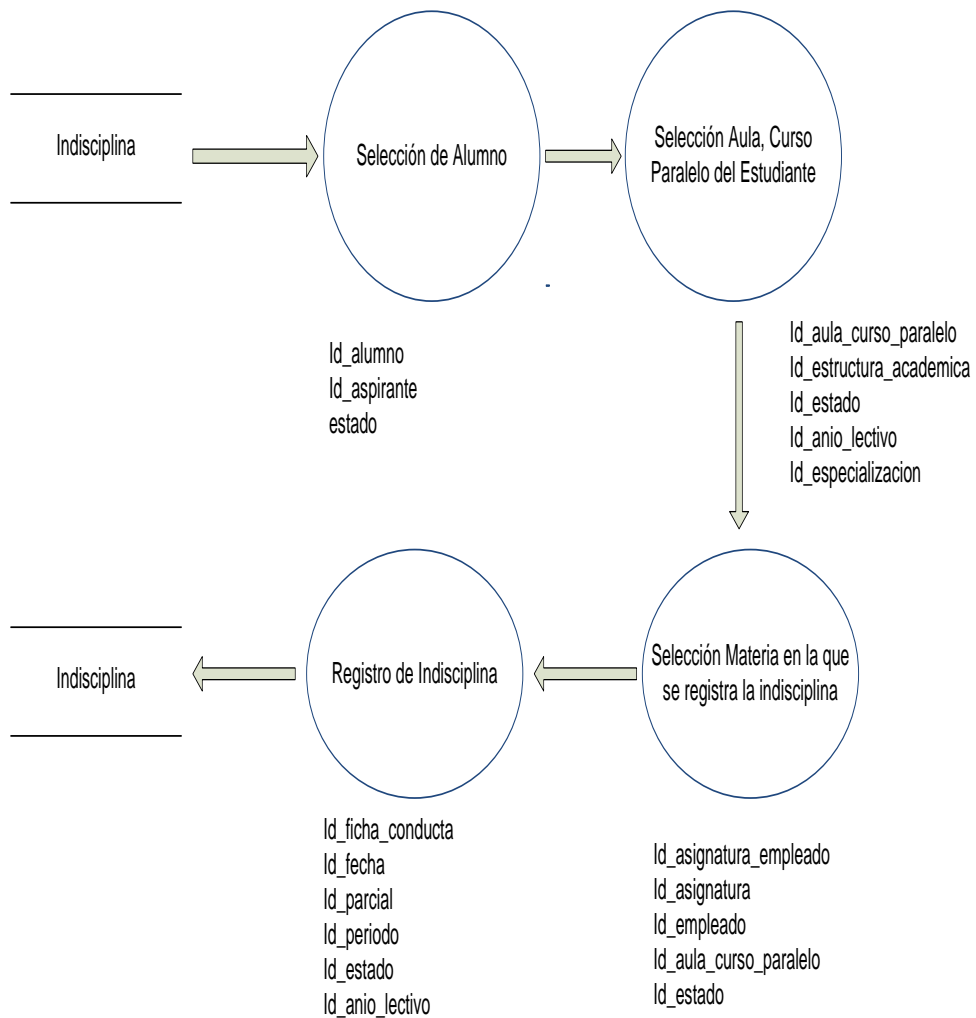


Figura #10: Diagrama de indisciplina de faltas por alumno, materia y curso

Fuente: Los autores.

3.1.6 Ingreso de Pruebas de Admisión

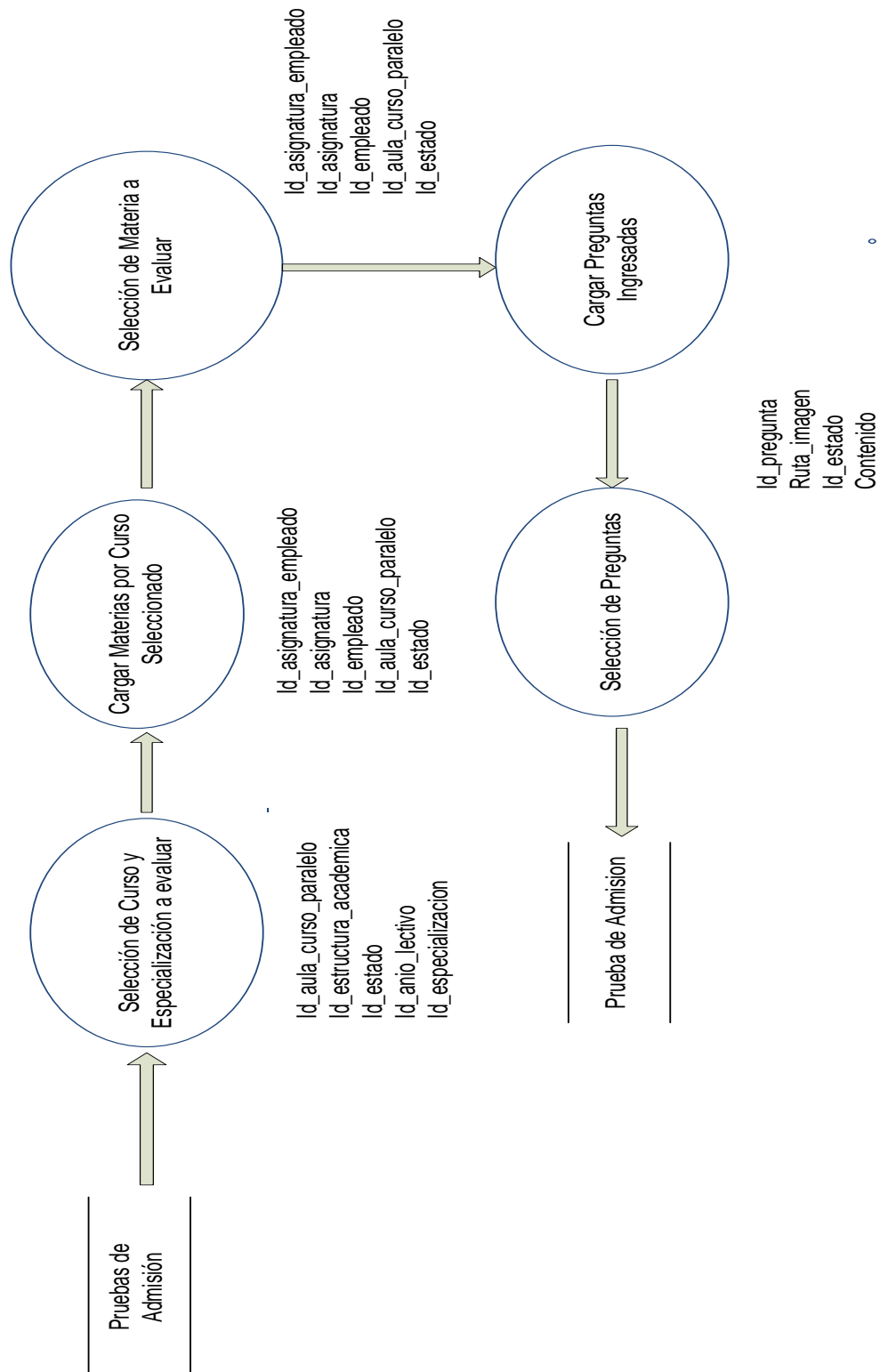


Figura #11: Diagrama de ingreso de pruebas de admisión, materia, curso y especialidad

Fuente: Los autores.

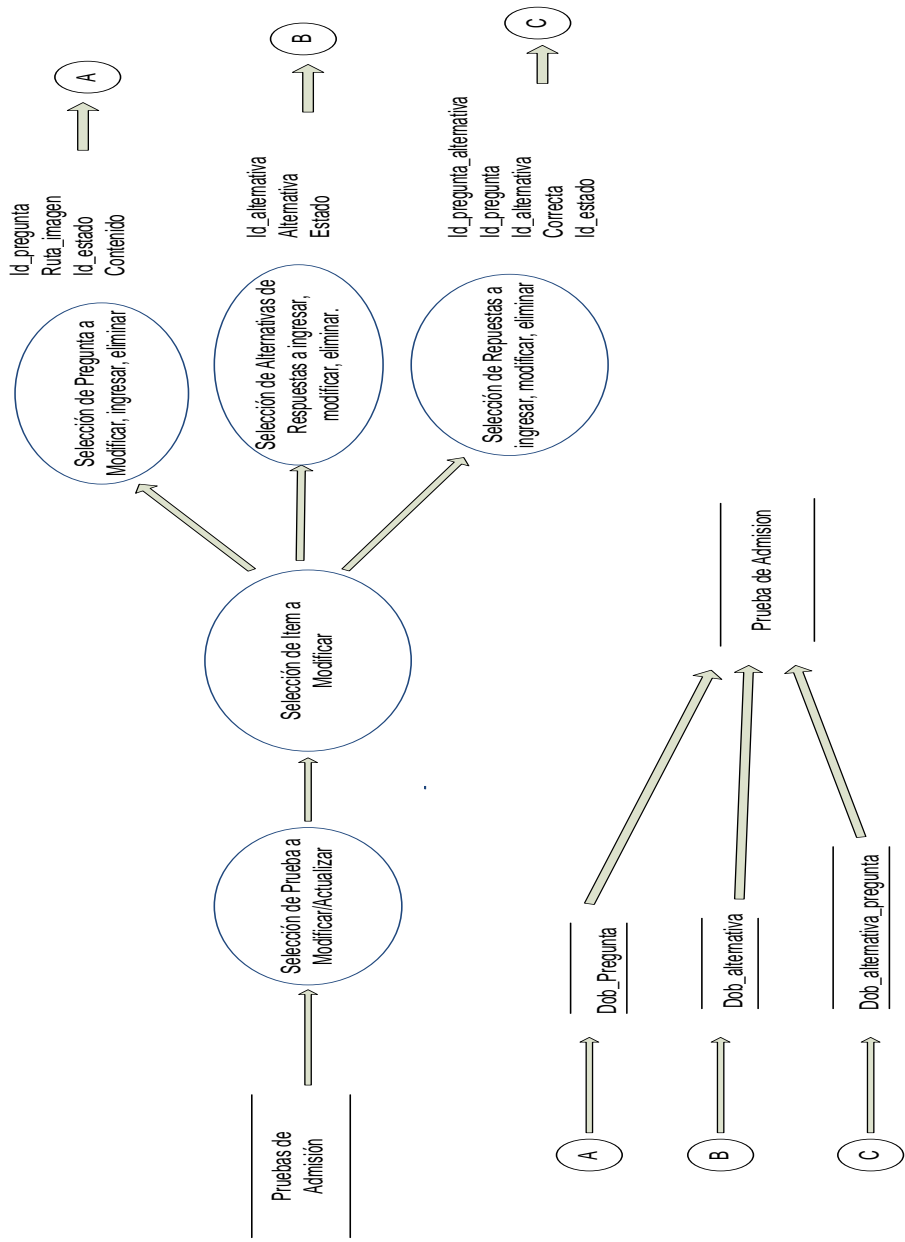


Figura #12: Actualización, Ingreso, modificación de Preguntas, alternativas y Respuestas de Pruebas de Admisión

Fuente: Los autores.

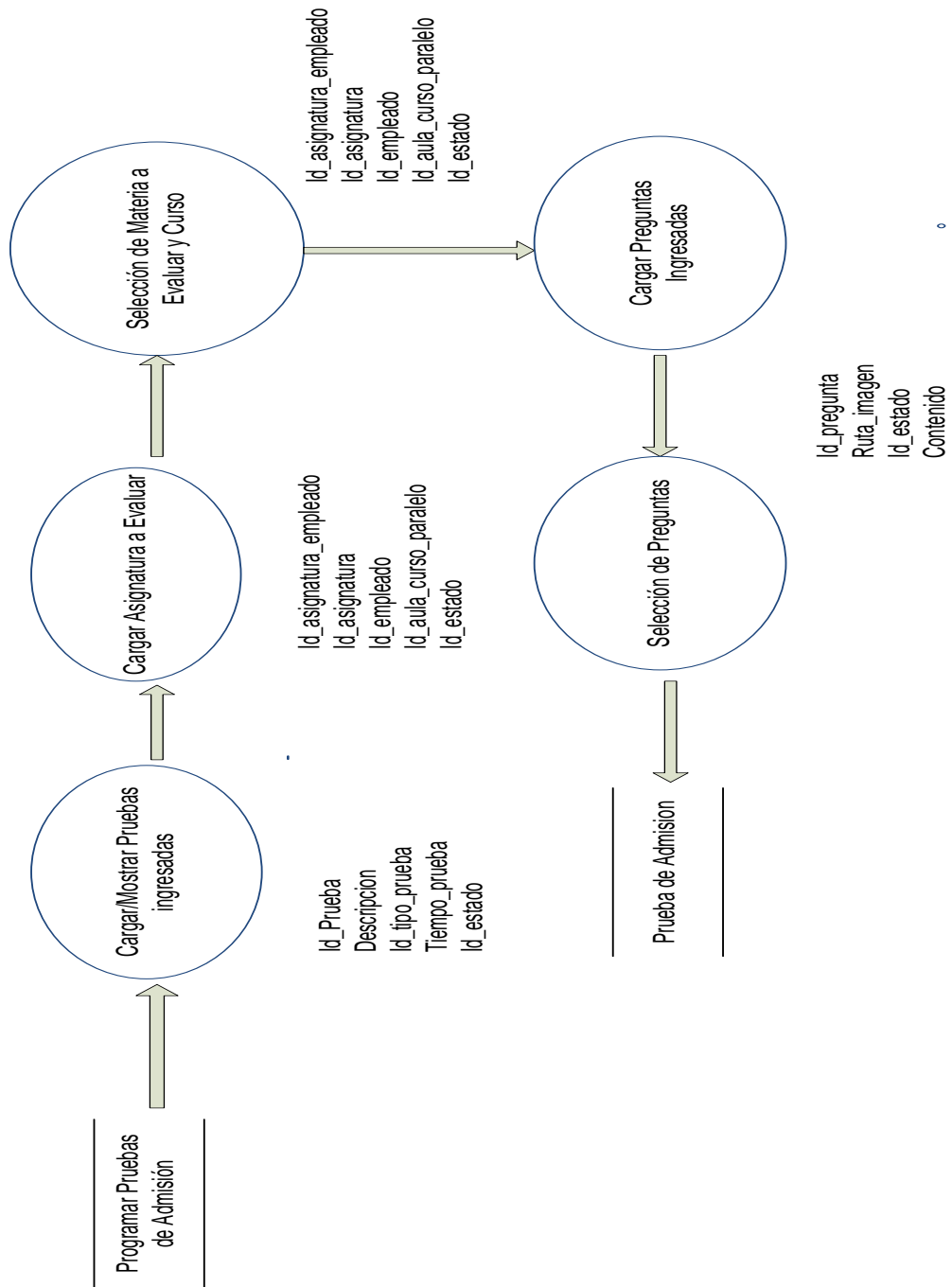


Figura #13: Programación de Pruebas de Admisión

Fuente: Los autores.

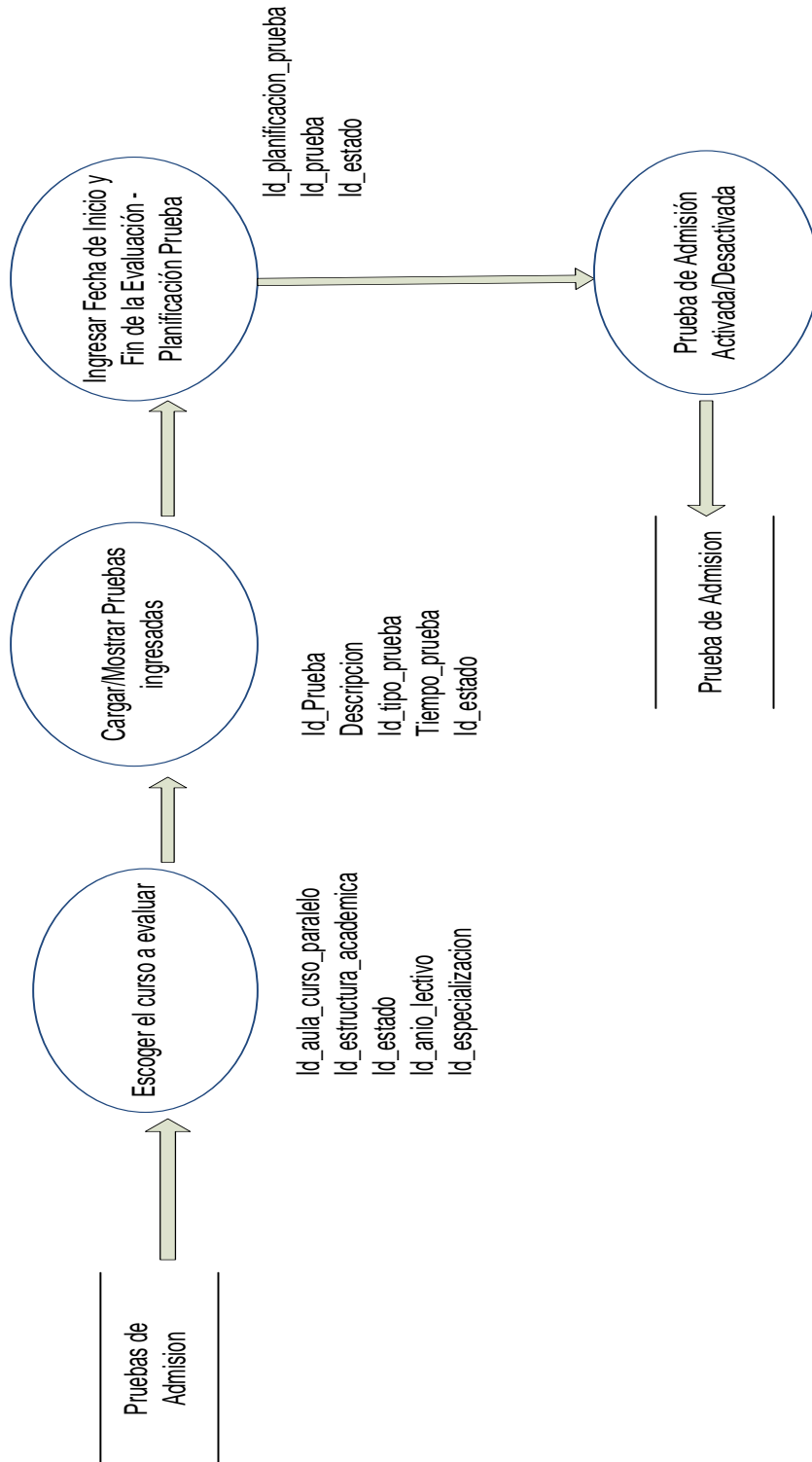


Figura #14: Programación de fechas de activación y desactivación de pruebas de admisión

Fuente: Los autores.

3.1.7 Ingreso Categoría Calificación

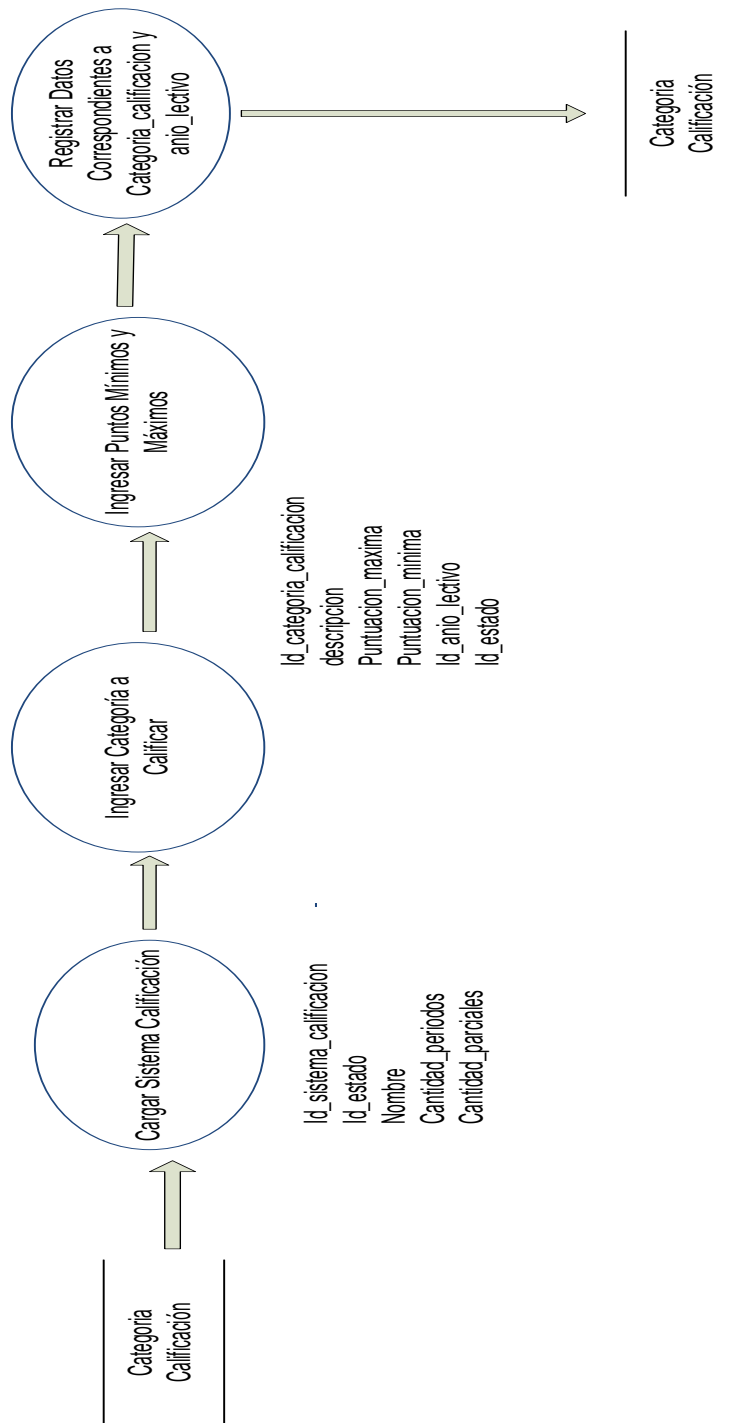


Figura #15: Diagrama de ingreso de una categoría calificación.

Fuente: Los autores.

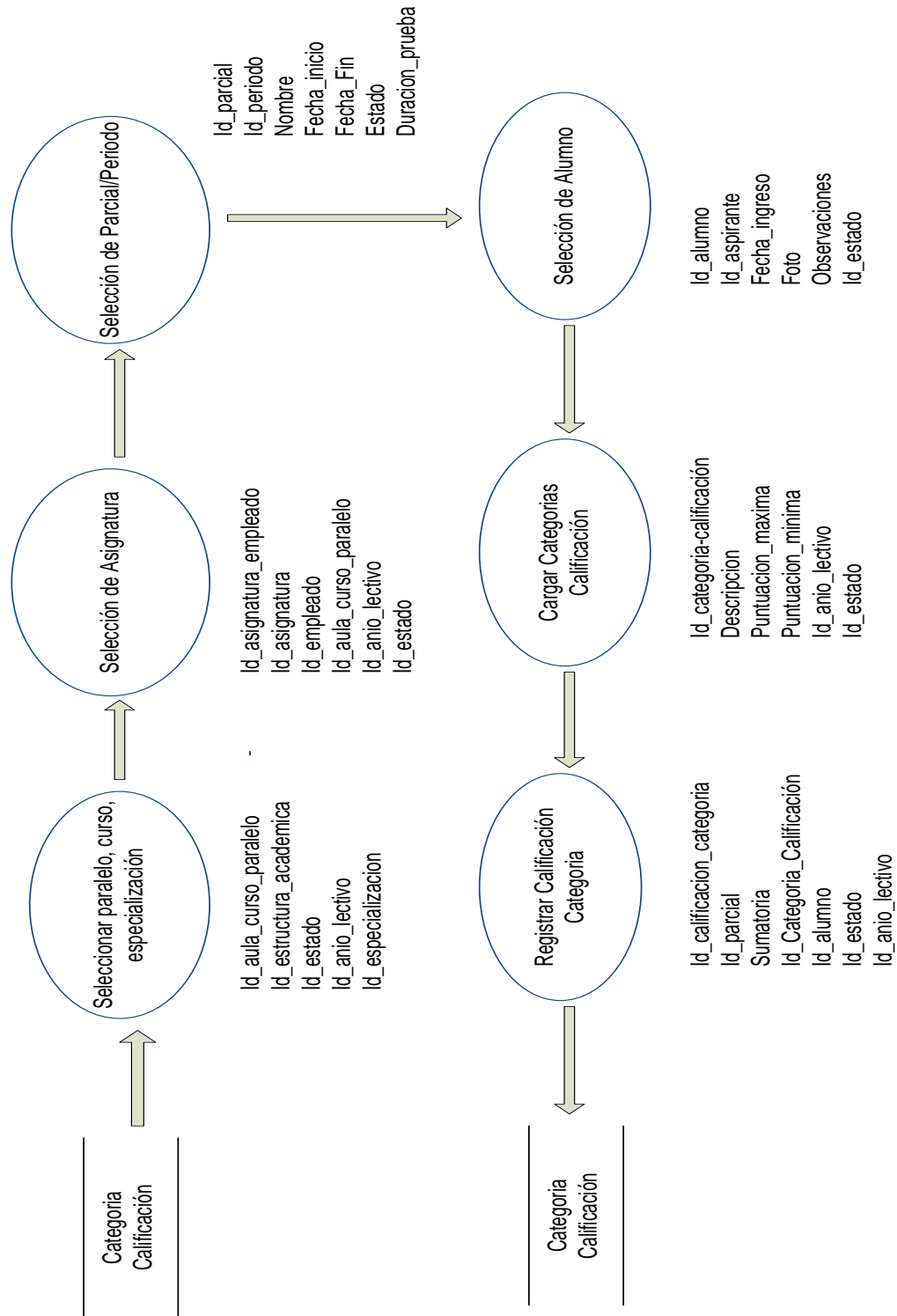


Figura #16: Registro de calificación por categoría.

Fuente: Los autores.

3.1.8 Modificación de Nota Parcial

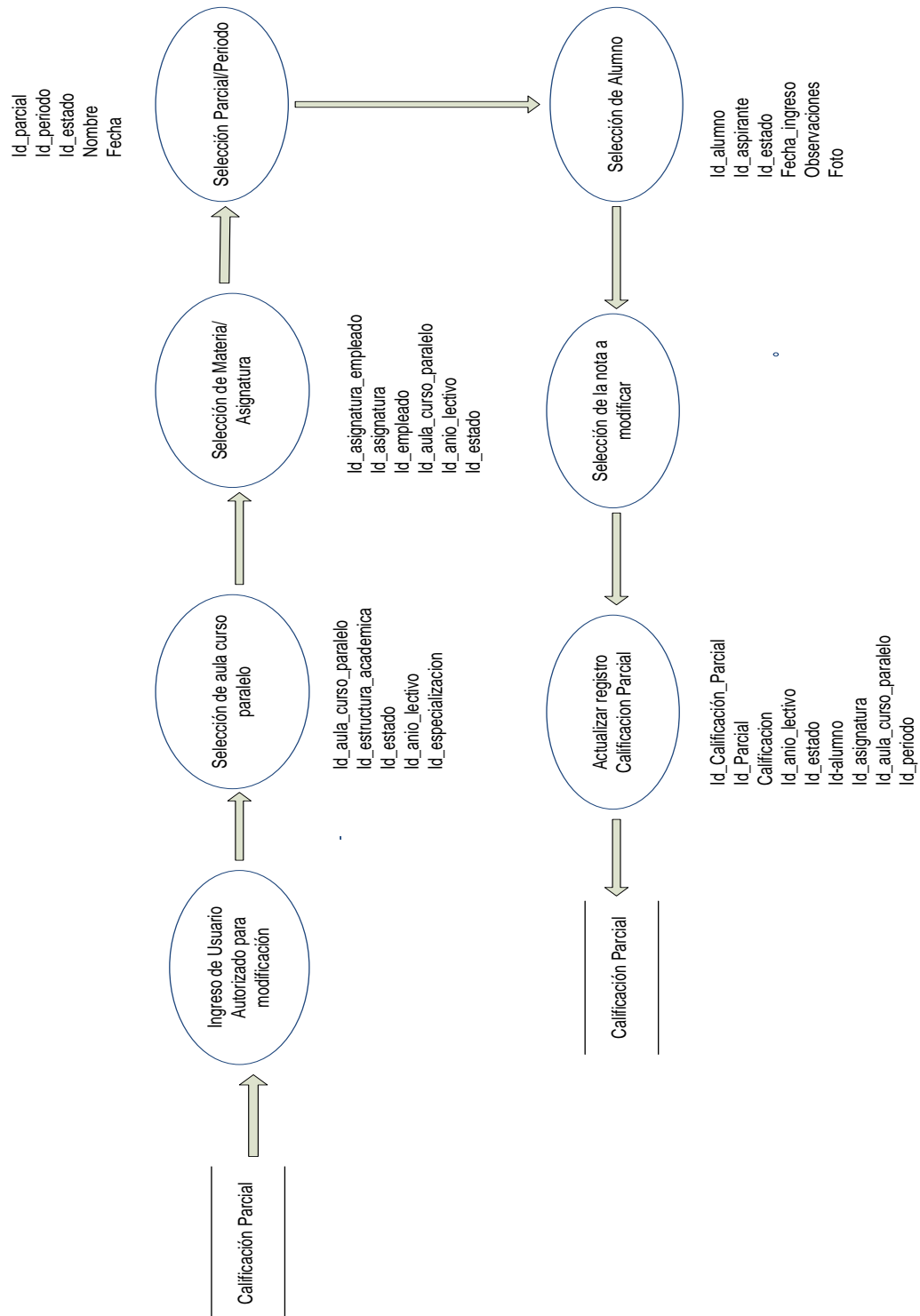


Figura #17: Modificación de una nota parcial realizada por secretaría.

Fuente: Los autores.

3.1.9 Registro de Materias a Evaluar en Examen de Grado

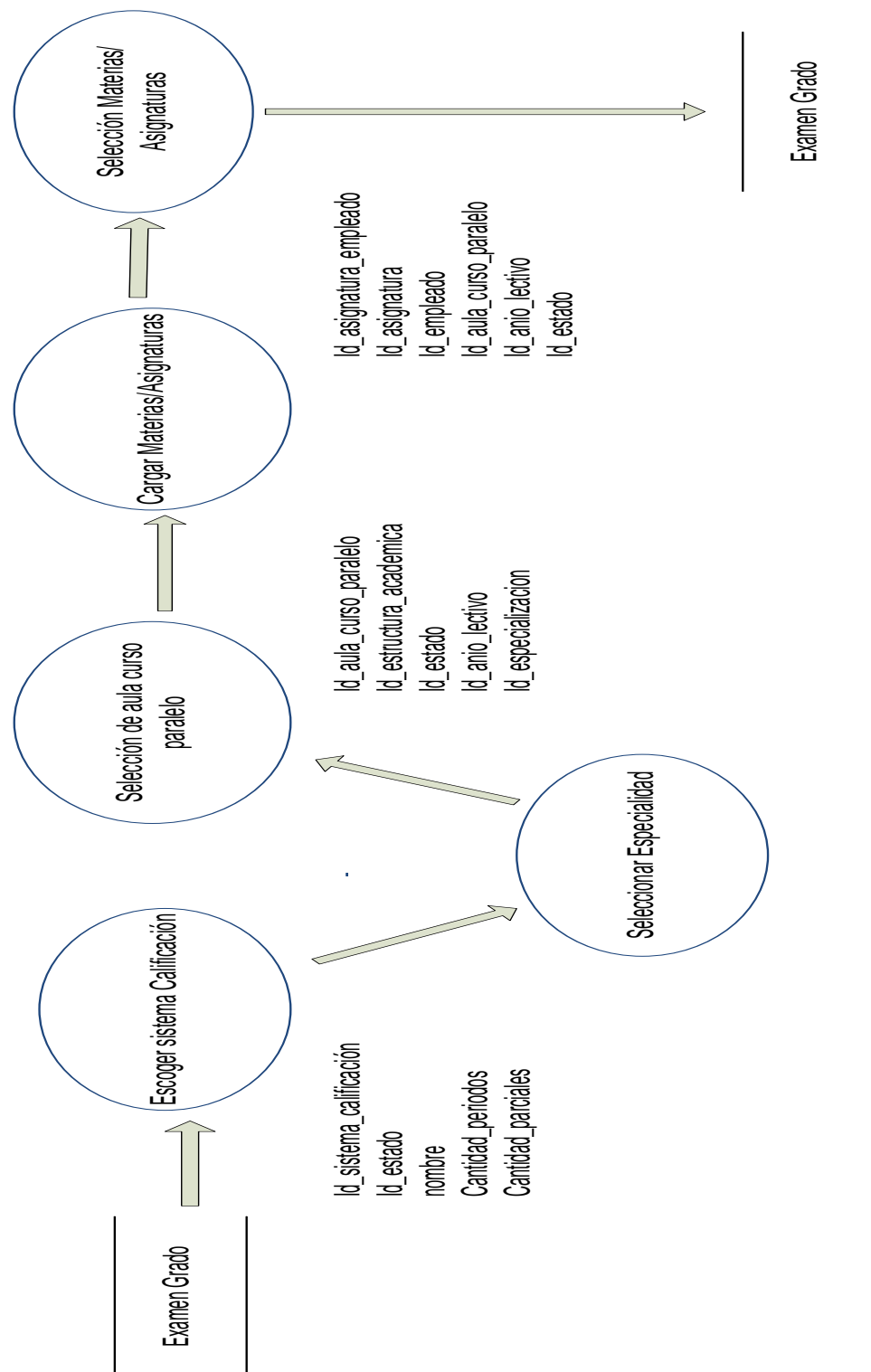


Figura #18: Registro y selección de las materias a evaluar para examen de grado por especialidad

Fuente: Los autores.

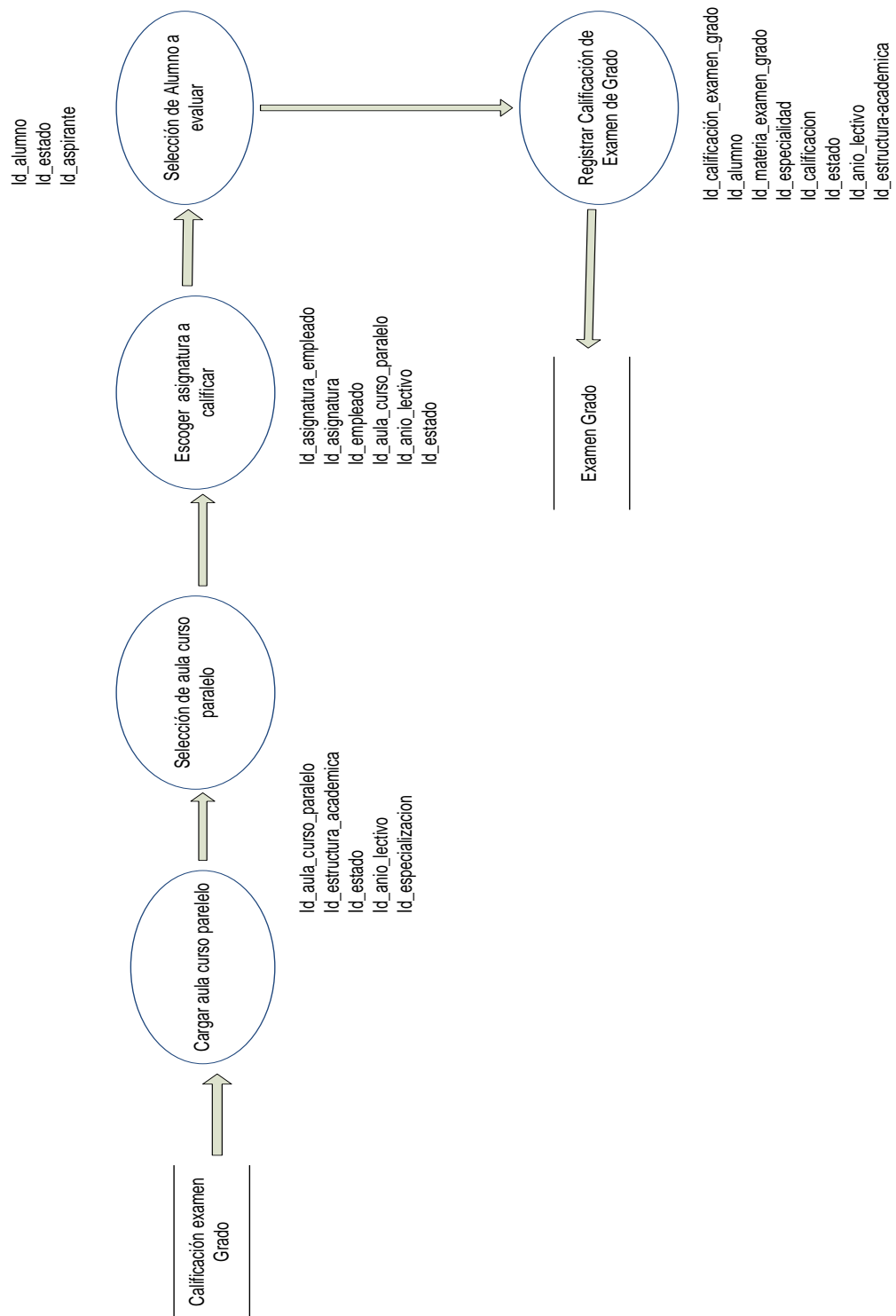


Figura #19: Registro de calificación de examen de grado.

Fuente: Los autores

3.2.1 Ingreso de Paracadémico

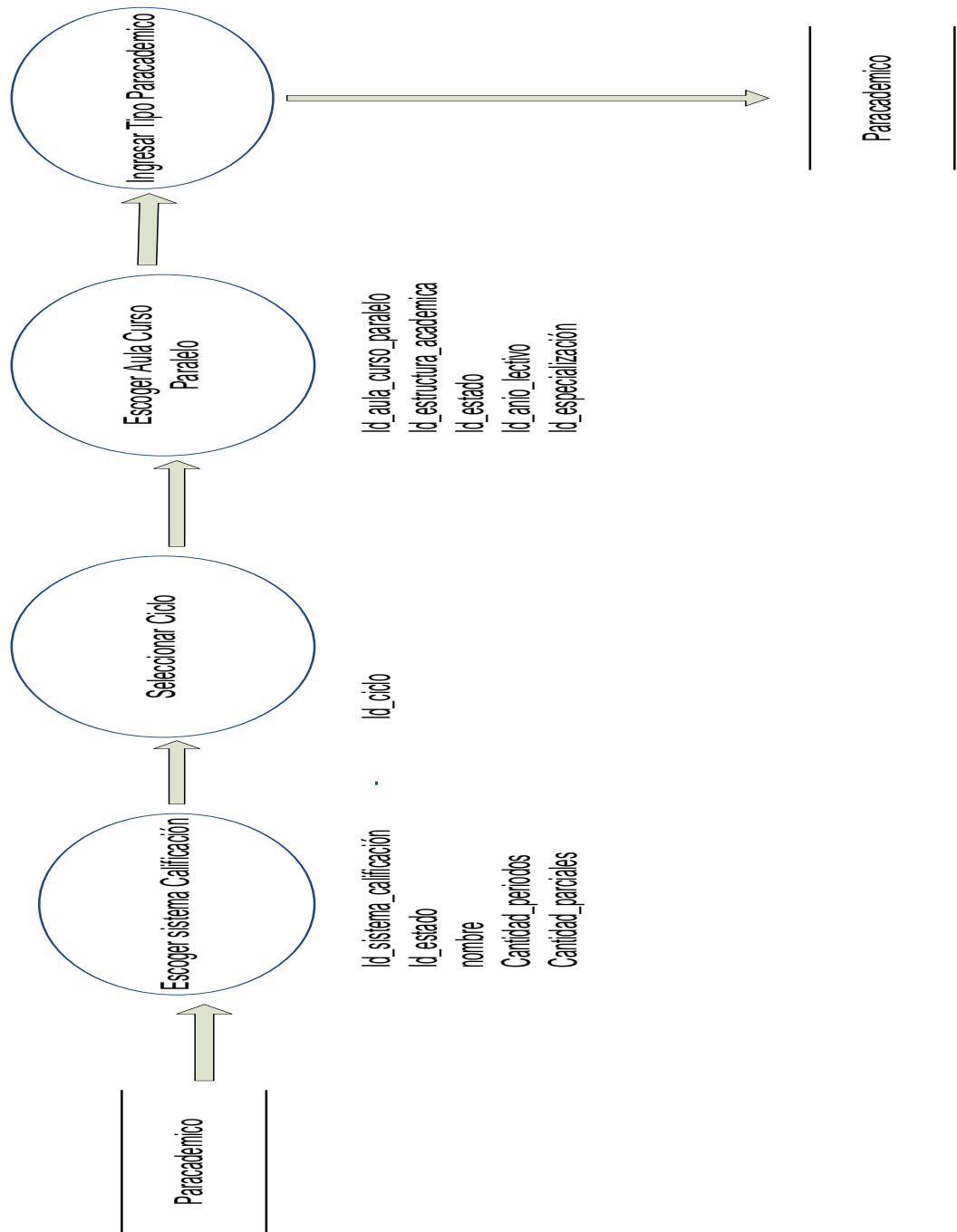


Figura #20: Registrar el tipo de paracadémico y el curso a utilizar.

Fuente: Los autores

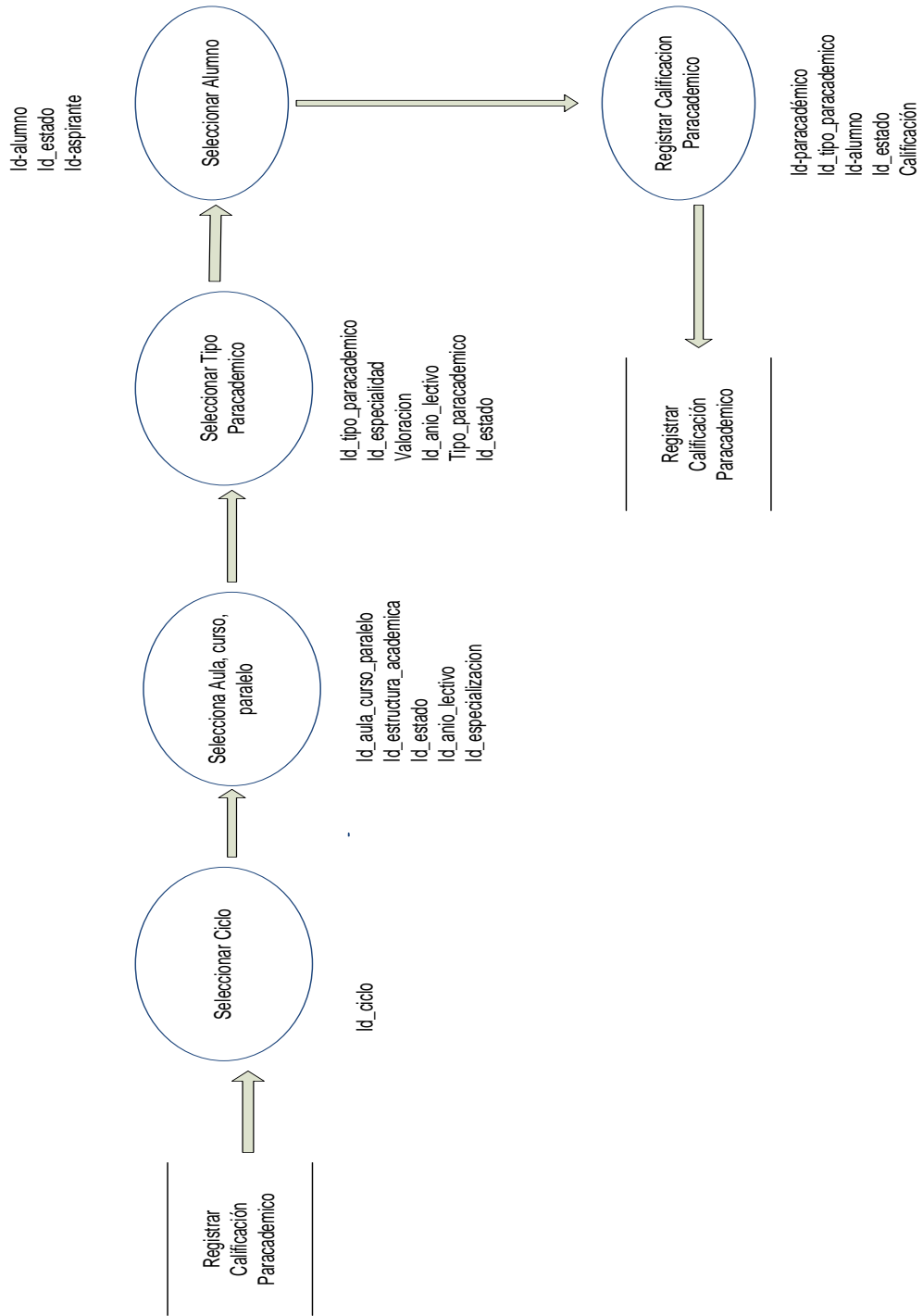


Figura #21: Registrar la calificación de paracadémico.

Fuente: Los autores.

3.3 Diagrama de Clase

3.3.1 Diagrama de Calificación

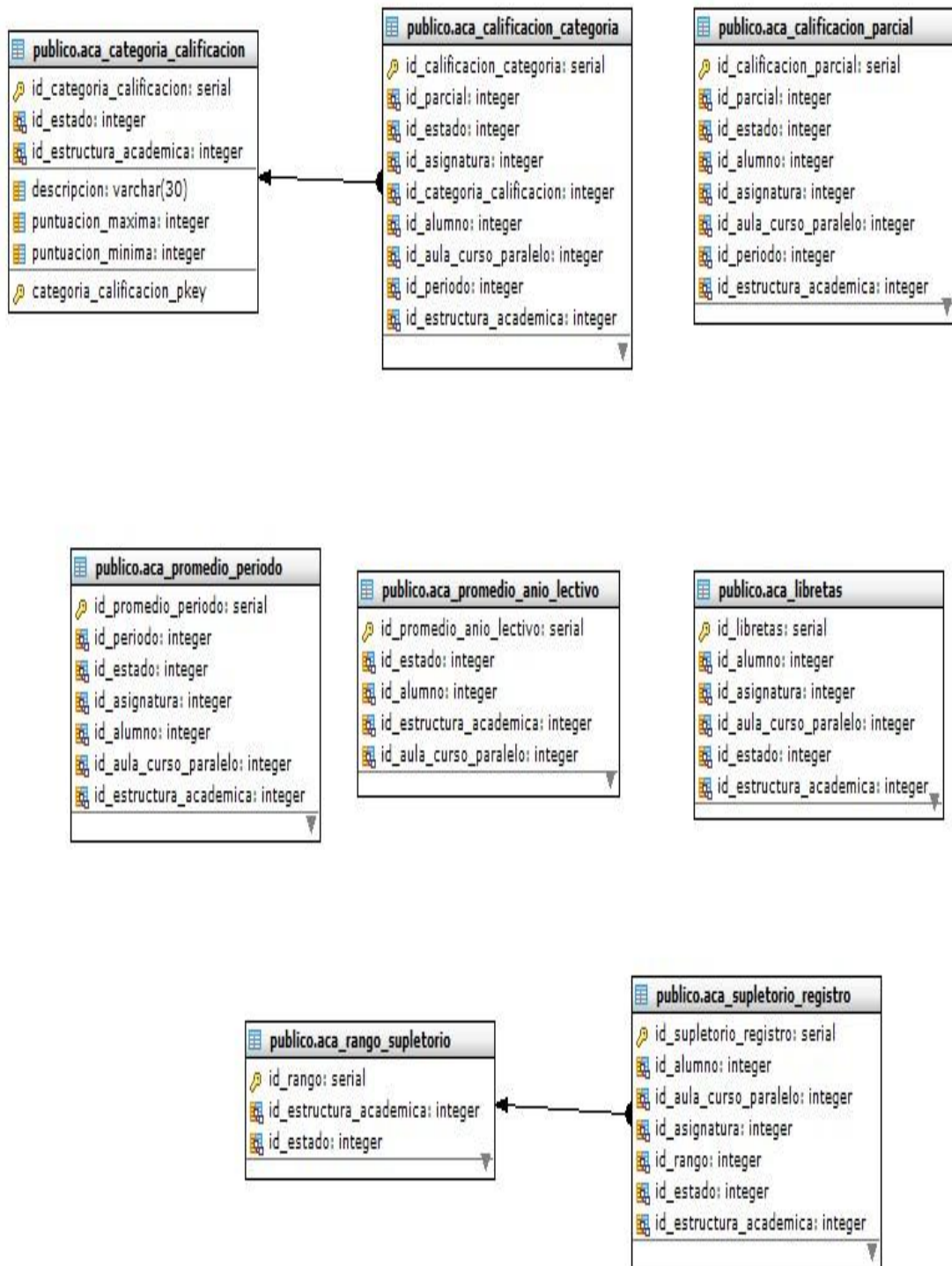


Figura #22: Diagrama de clase correspondiente a calificación categoría, calificación parcial, promedios, supletorio.

Fuente: Los autores

3.3.2 Diagrama de Supervisión y Tutoría

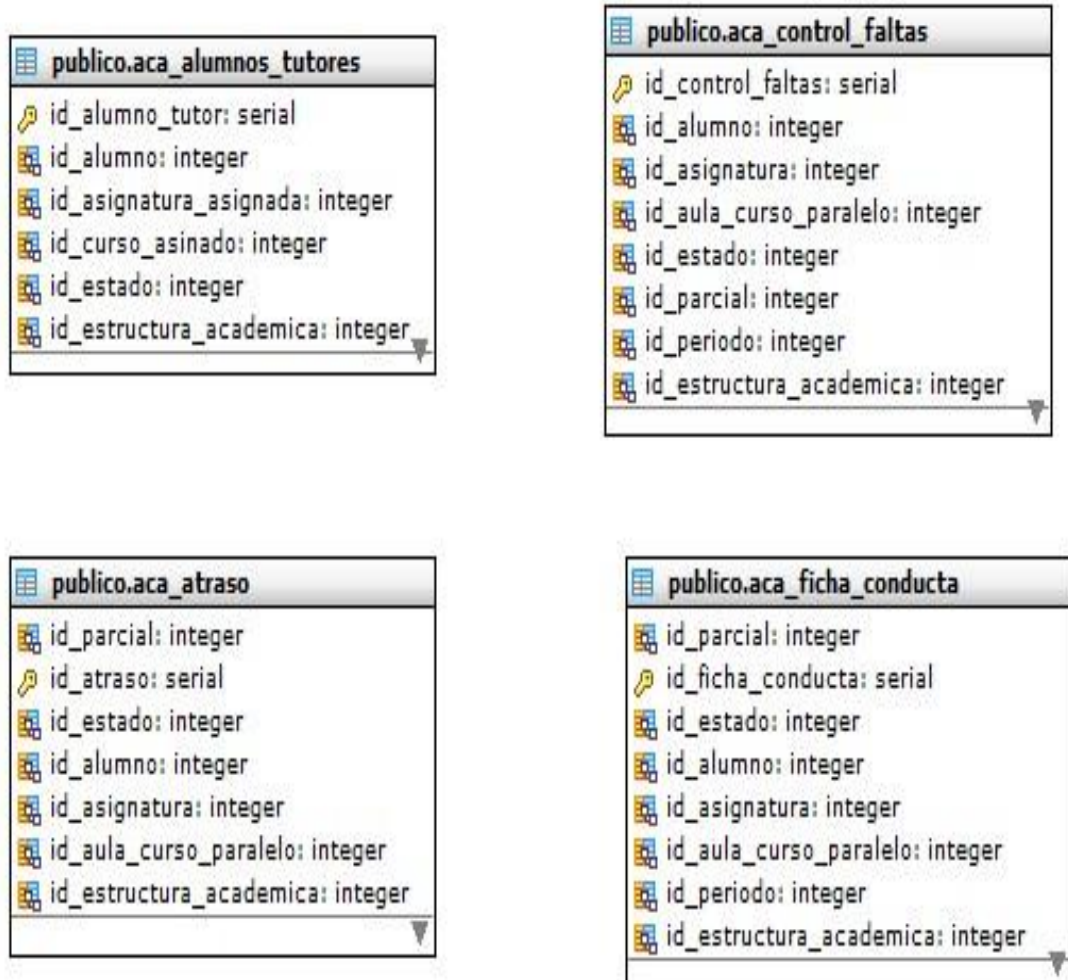


Figura #23: Diagrama de clase correspondiente a tutoría, atraso, faltas y conducta.

Fuente: Los autores.

3.3.3 Diagrama de Promedios de Grado

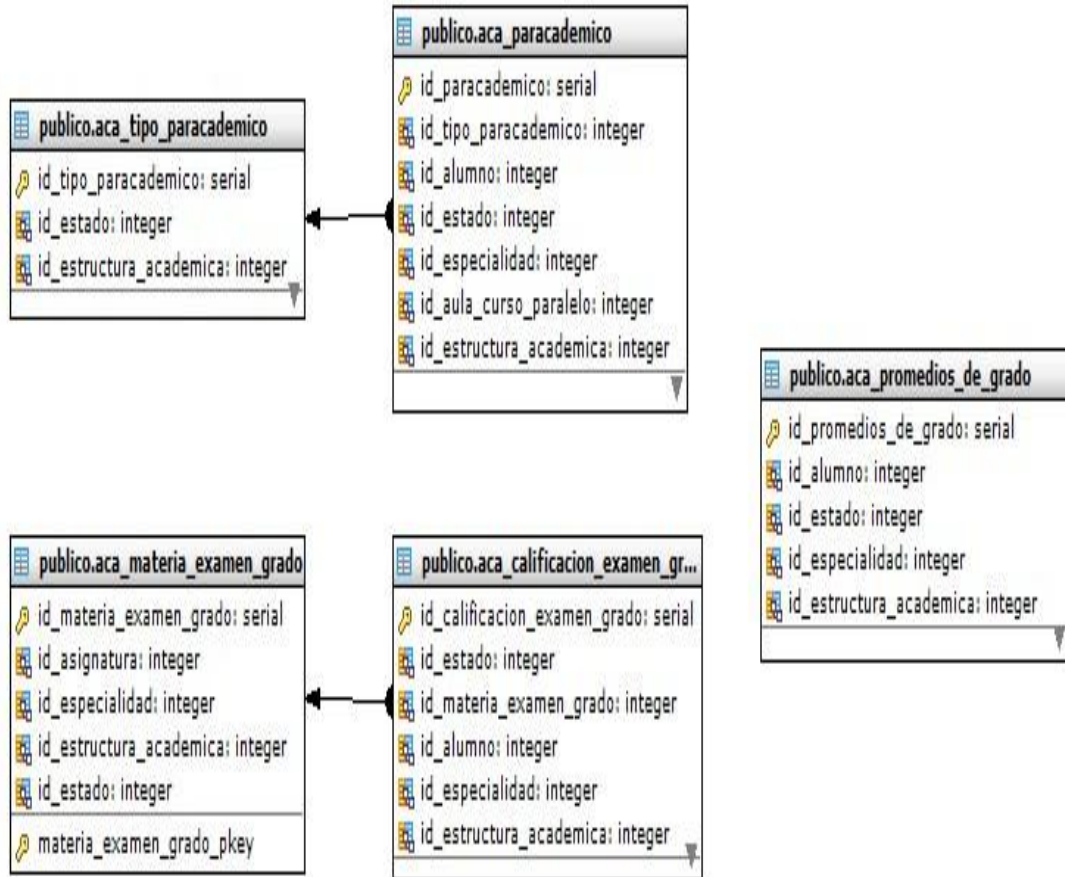


Figura #24: Diagrama de clase correspondiente a tutoría, atraso, faltas y conducta.

Fuente: Los autores

3.4 Diagrama de Casos de Uso

3.4.1 Calificación Parcial

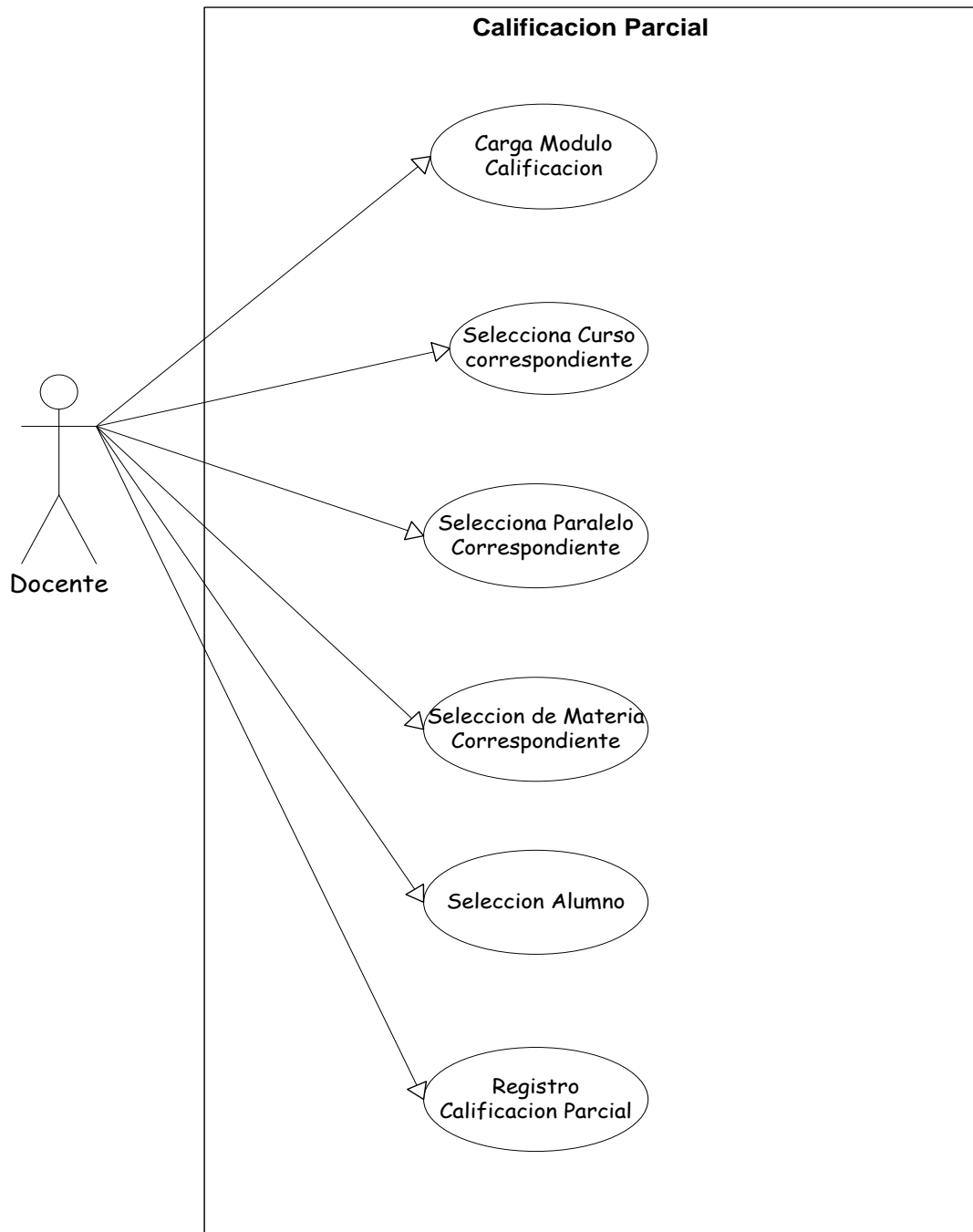


Figura # 25: Diagrama de caso de uso de calificación parcial.

Fuente: Los autores

Nombre:	Calificación Parcial
Autor:	Los Autores
Fecha:	30/05/2012
Descripción: Permite almacenar la calificación parcial de un alumno.	
Actores: Docente.	
Precondiciones: Se debe haber realizado previo trabajo de investigación, deber, lección, etc.	
Flujo Normal: <ol style="list-style-type: none"> 1) El docente selecciona curso, paralelo y materia. 2) Selecciona el alumno 3) Registra la calificación del alumno. 	
Flujo Alternativo: Se valida la calificación del alumno con el medio calificado, sea lección, examen, investigación, etc.	
Poscondiciones: La calificación se registra y almacena en la base de datos del sistema.	

Tabla #8: Cuadro descriptivo del caso de uso calificación parcial.

Fuente: Los autores.

3.4.2 Control de Asistencia

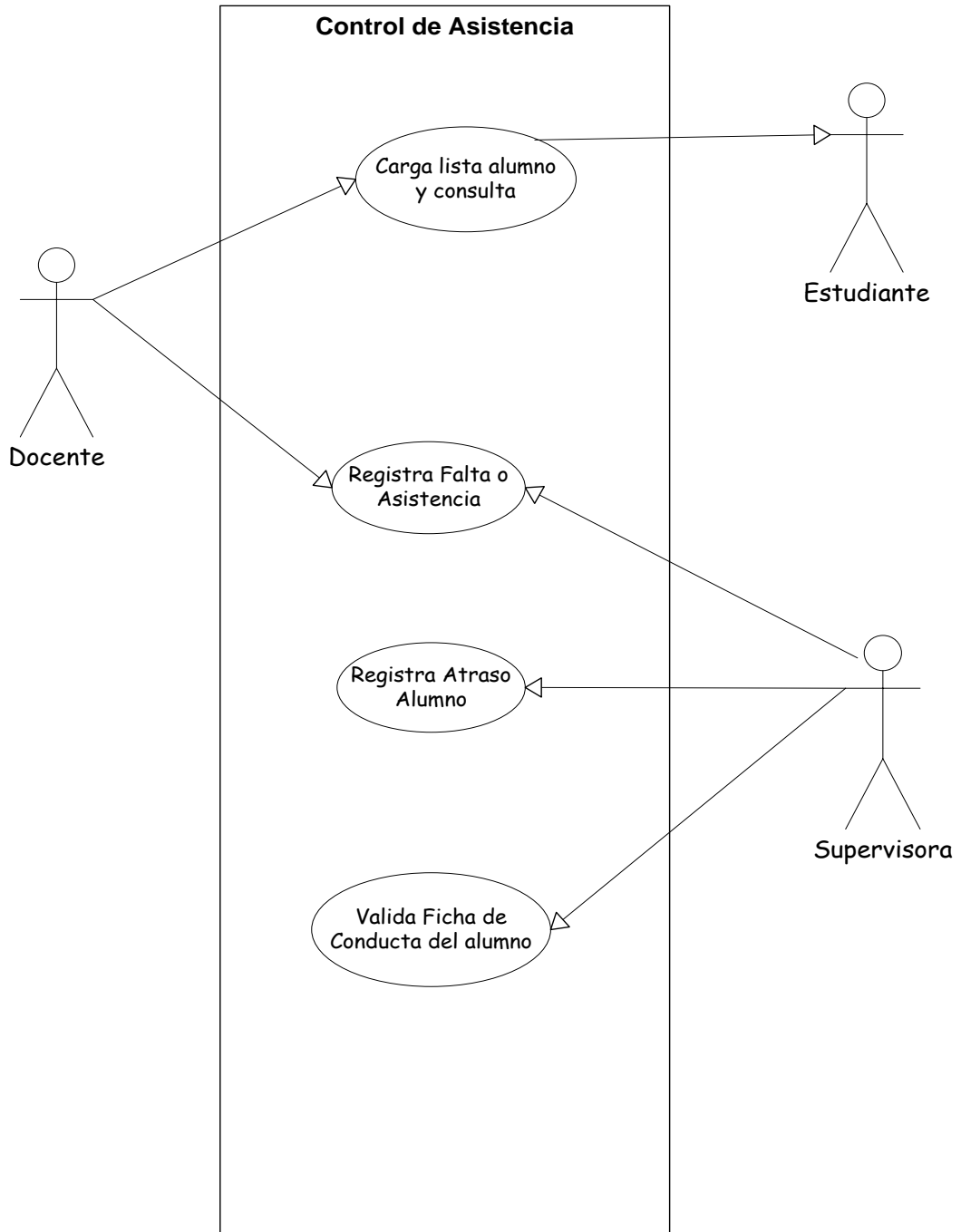


Figura #26: Diagrama de caso de uso de control de asistencia

Fuente: Los autores

Nombre:	Control de Asistencia.
Autor:	Los Autores
Fecha:	30/05/2012
Descripción: Permite registrar, almacenar y controlar la asistencia del alumnado.	
Actores: Estudiante, Supervisora, Docente.	
Precondiciones: La ausencia de alumno debe ser notificada por el docente hacia la supervisora.	
Flujo Normal: <ol style="list-style-type: none"> 1) El docente valida asistencia del alumno en su clase. 2) Registra la falta y reporta al supervisor/a. 3) La supervisora confirma la falta del alumno y según corresponda abre una ficha de conducta o direcciona el caso al DOBE. 	
Flujo Alternativo: Se valida el código del alumno para confirmar curso y paralelo asignado y proceder a realizar el registro de la asistencia.	
Poscondiciones: El detalle de la asistencia ha sido almacenado en el sistema.	

Tabla #9: Cuadro descriptivo del caso de uso de control de asistencia.

Fuente: Los autores.

3.4.3 Gestión de Conflicto de Convivencia

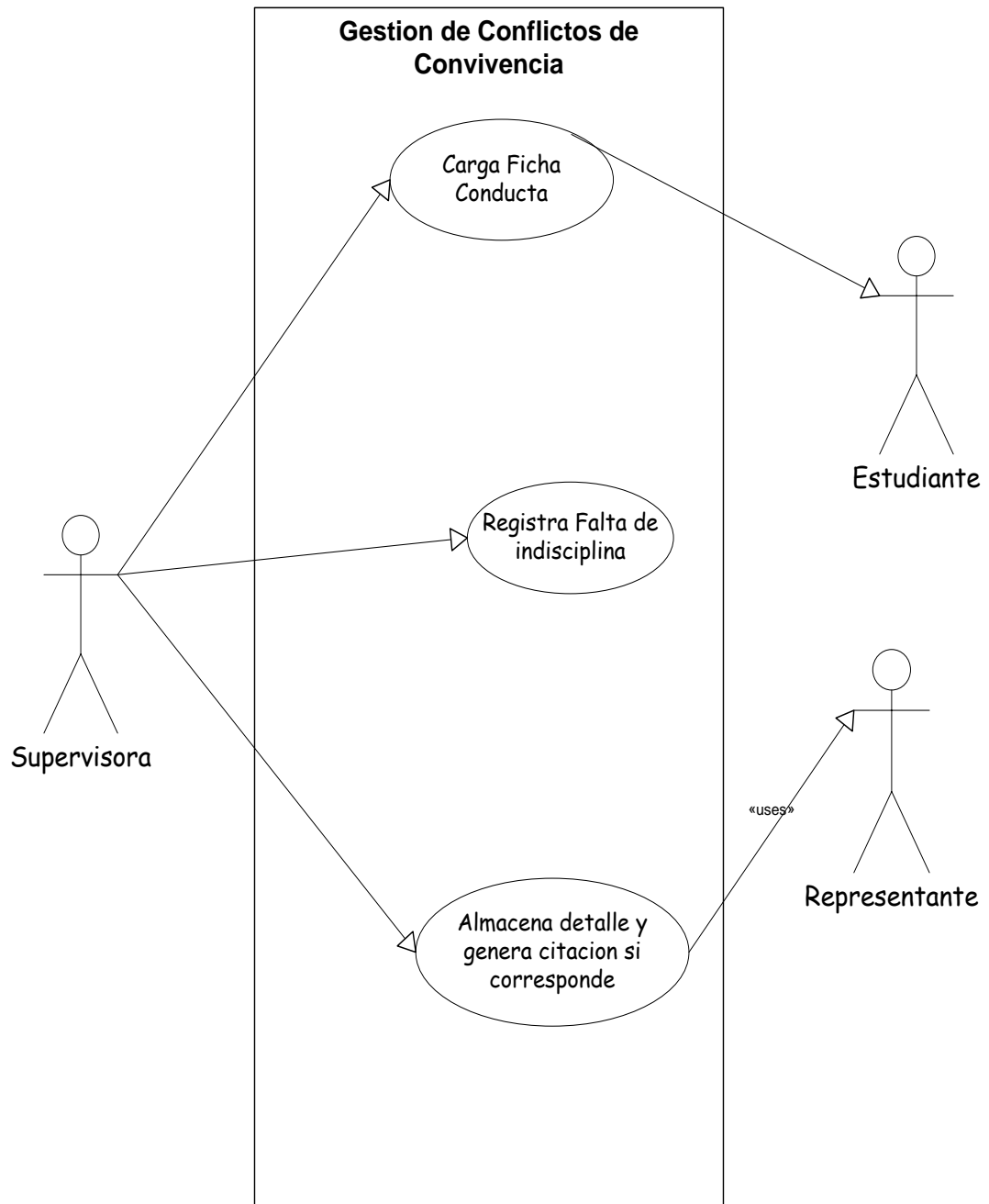


Figura # 27: Diagrama de caso de uso de gestión de convivencia.

Fuente: Los autores.

Nombre:	Gestión de Conflictos de Convivencia.
Autor:	Los Autores
Fecha:	30/05/2012
Descripción: Permite almacenar información acerca de una falta de conducta por parte del alumno.	
Actores: Estudiante, Supervisora, Representante.	
Precondiciones: El alumno debe cometer una falta que atente contra el reglamento de la institución.	
Flujo Normal: <ul style="list-style-type: none"> 4) El alumno comete la falta 5) La falta es comentada/presenciada por el/la supervisora. 6) La supervisora registra el acto de indisciplina y genera una citación para el representante. 	
Flujo Alternativo: Se valida el código del alumno para confirmar curso y paralelo asignado y proceder a generar la citación.	
Poscondiciones: El detalle de la falta ha sido almacenado en el sistema	

Tabla #10: Cuadro descriptivo del caso de uso de gestión de convivencia.

Fuente: Los autores.

3.4.4 Tutoría

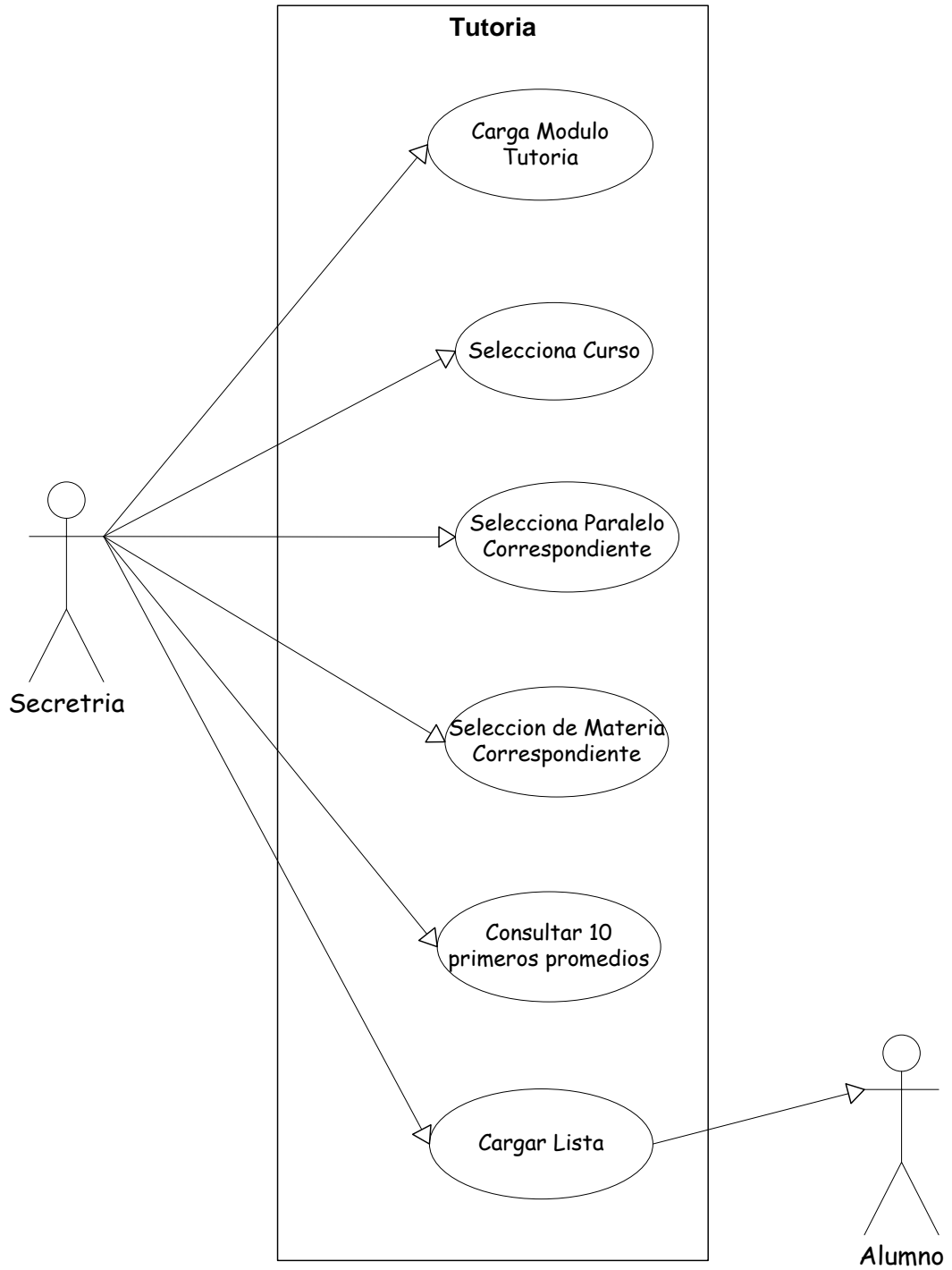


Figura #28: Diagrama de caso de uso para tutoría.

Fuente: Los autores.

Nombre:	Tutoría
Autor:	Los Autores
Fecha:	30/05/2012
Descripción: Permite consultar los mejores promedios en cada materia con el fin de seleccionar posible ayudantes o tutores.	
Actores: Secretaria, Alumno.	
Precondiciones: El alumno debe tener una nota/promedio en un rango muy bueno- sobresaliente para ser considerado como posible tutor.	
Flujo Normal: <ol style="list-style-type: none"> 1) Secretaría realiza consulta de calificaciones por curso, paralelo y materia. 2) Realiza una pre-selección de los posibles tutores 3) Obtiene el listado final de los alumnos que serán asignado tutores. 	
Flujo Alternativo: Se valida el record académico del alumno para poder confirmar la selección como tutor.	
Poscondiciones: El record académico, las calificaciones y promedio deben estar almacenados para poder realizar la consulta, selección y validación.	

Tabla #11: Cuadro descriptivo del caso de uso de tutoría.

Fuente: Los autores.

3.5 Diagrama de Secuencia

3.5.1 Diagrama de calificación de un determinado periodo y parcial

Registro de Calificación Docente

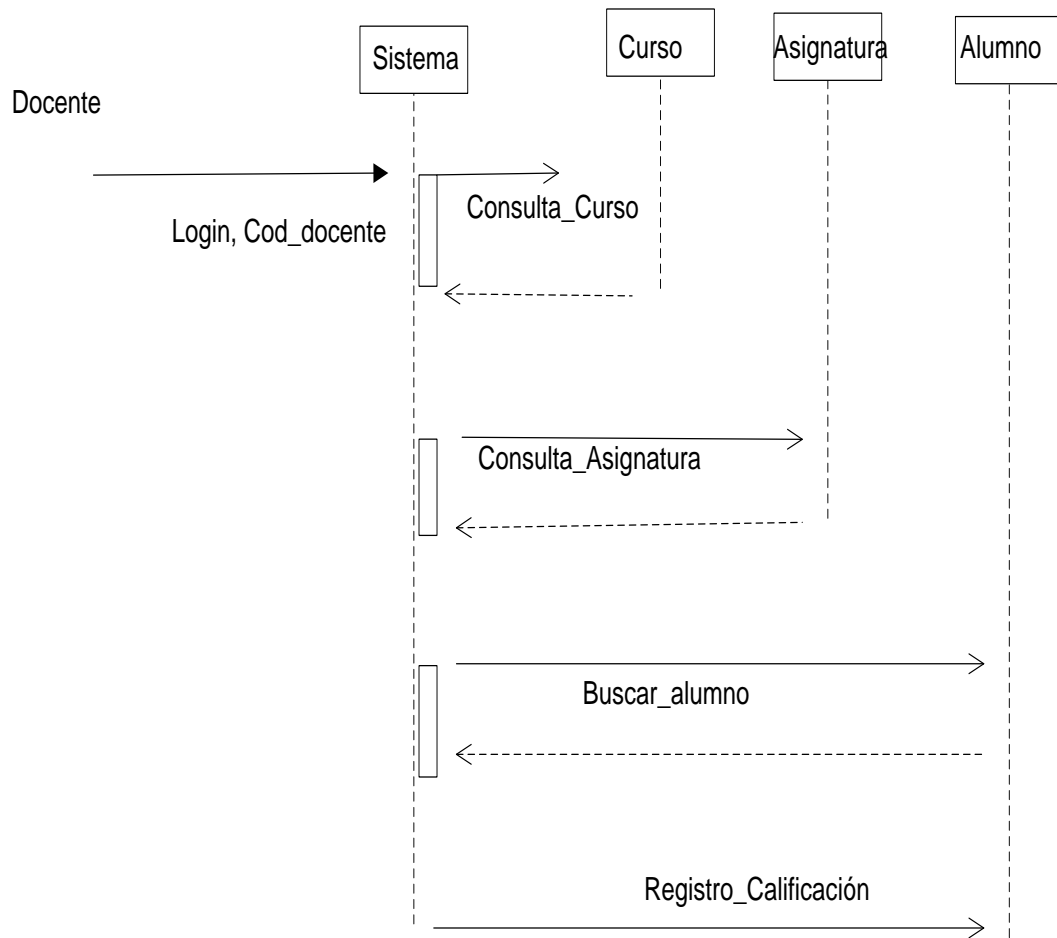


Figura #29: Diagrama de secuencia para calificación docente.

Fuente: Los autores

3.5.2 Diagrama de generación de promedio de grado

Generación Promedio Grado

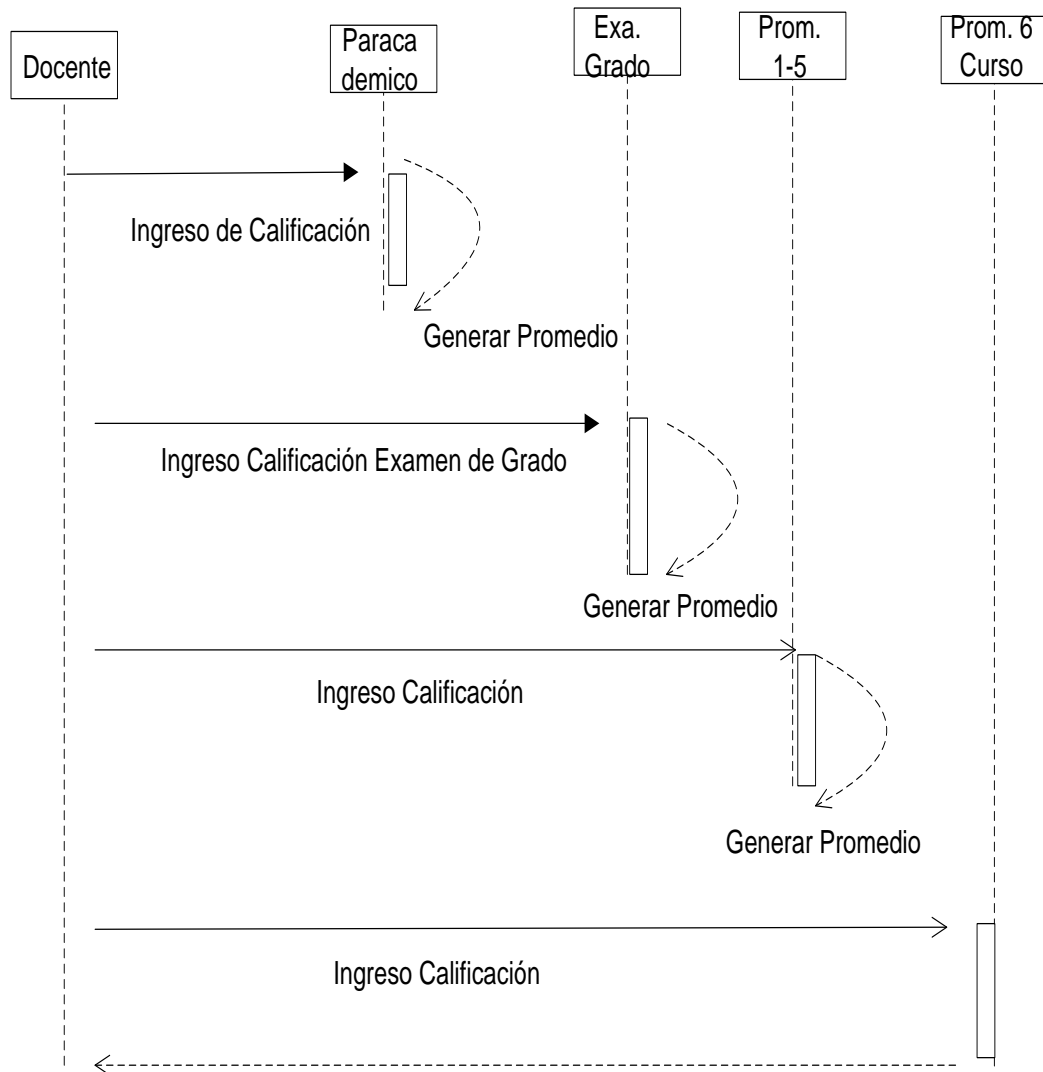


Figura #30: Diagrama de secuencia para la generación de promedio

Fuente: Los autores

3.5.3 Diagrama de tutoría

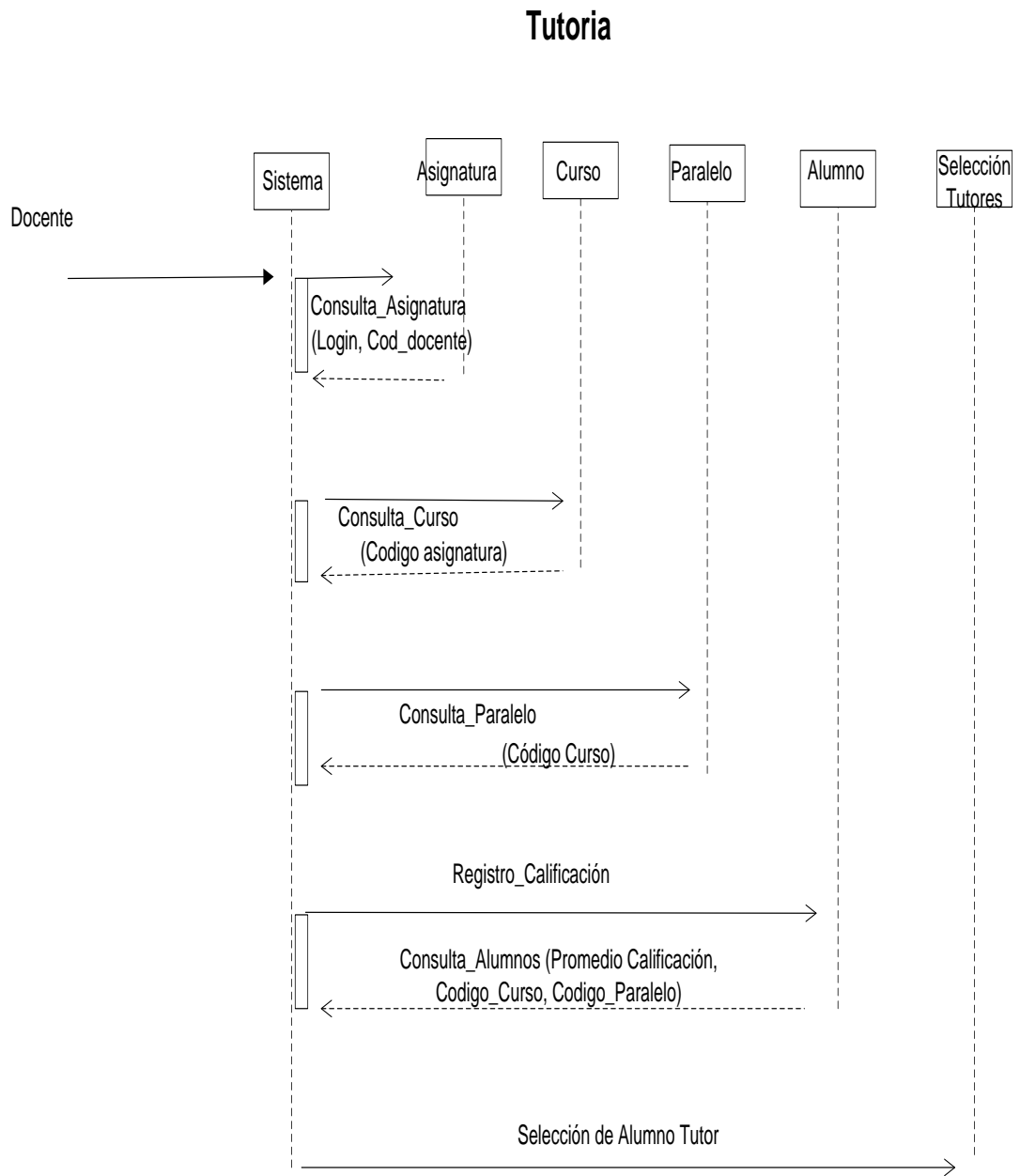


Figura #31: Diagrama de secuencia para tutoría.

Fuente: Los autores

3.6 Diagrama de Estado

3.6.1 Diagrama de calificación parcial de un determinado período

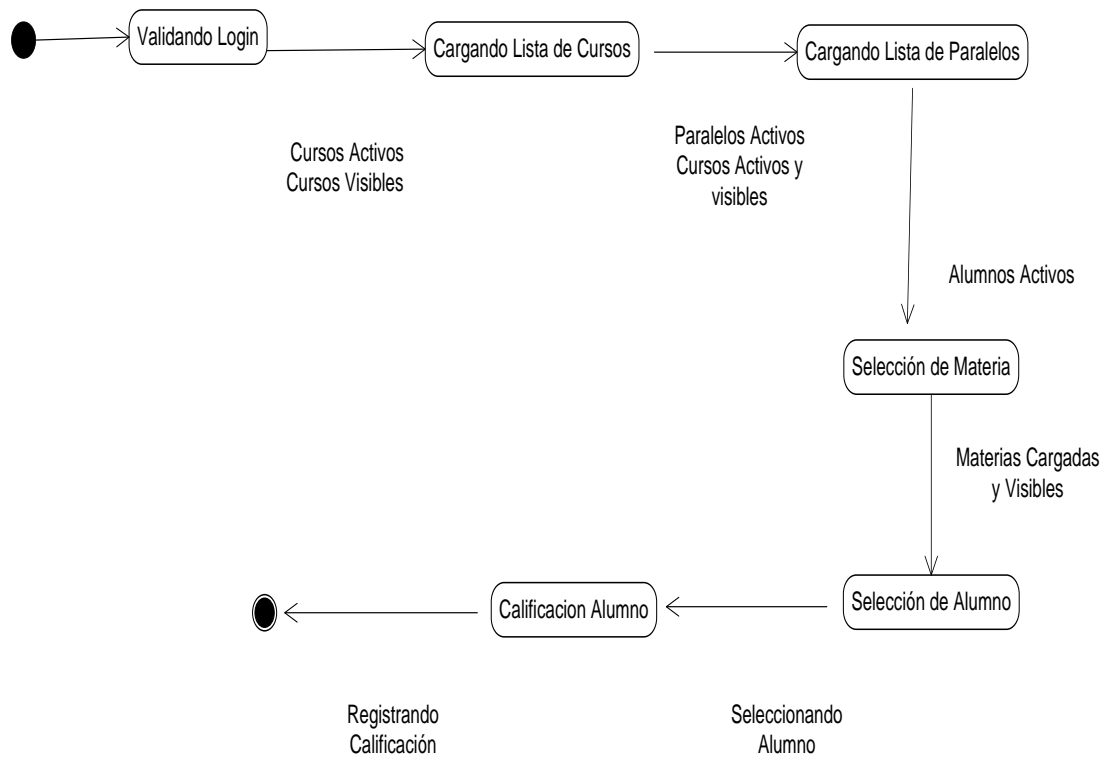


Figura #32: Diagrama de estado para la calificación parcial de un período.

Fuente: Los autores.

3.6.2 Diagrama de promedio para obtener nota de grado

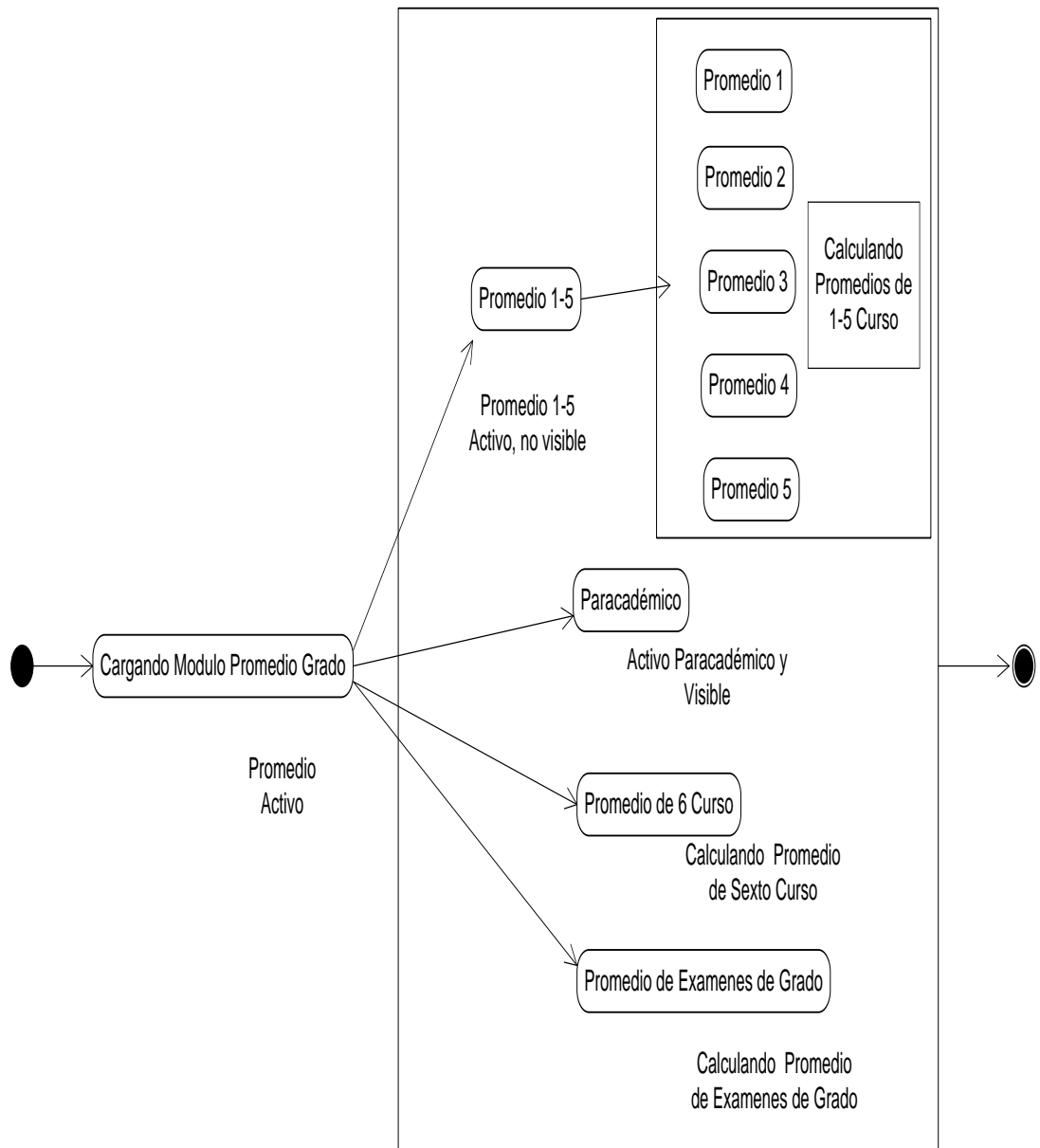


Figura #33: Diagrama de estado para la generación de promedio de grado.

Fuente: Los autores.

3.6.3 Diagrama de Tutoría

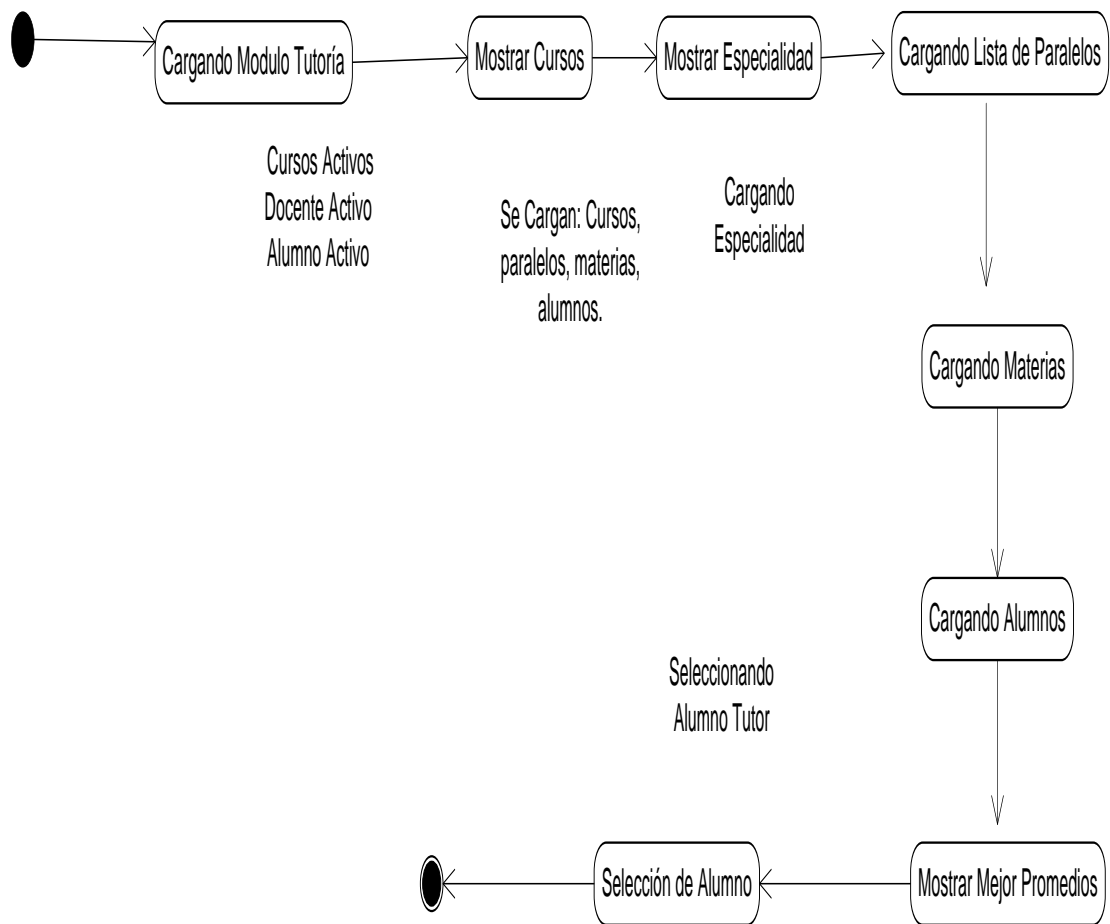


Figura #34: Diagrama de estado para tutoría.

Fuente: Los autores.

3.7 Modelo de Procesos

3.7.1 Modelo de Calificación Parcial

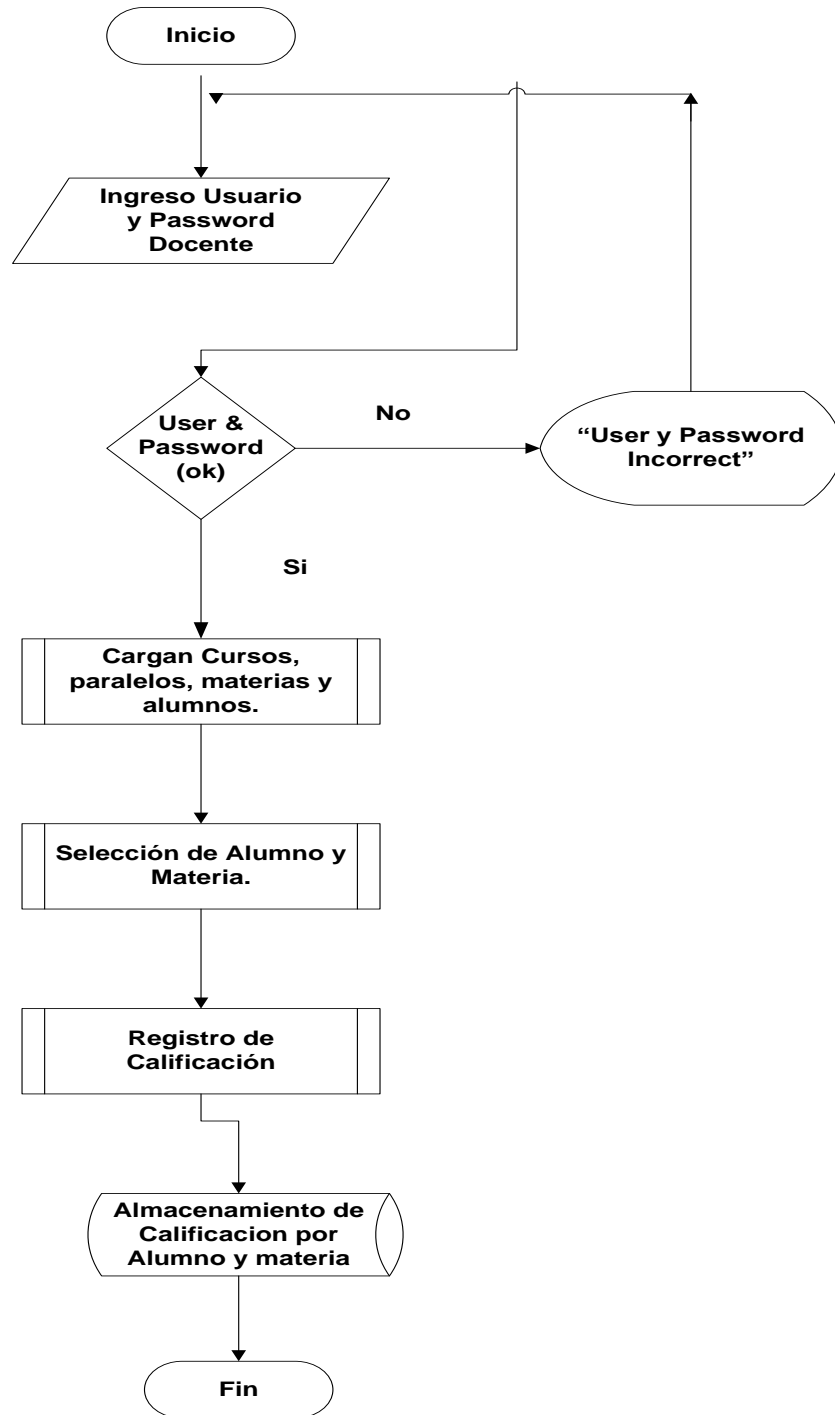


Figura #35: Modelo de Proceso para Calificación Parcial.

Fuente: Los autores.

3.7.2 Modelo de Tutoría

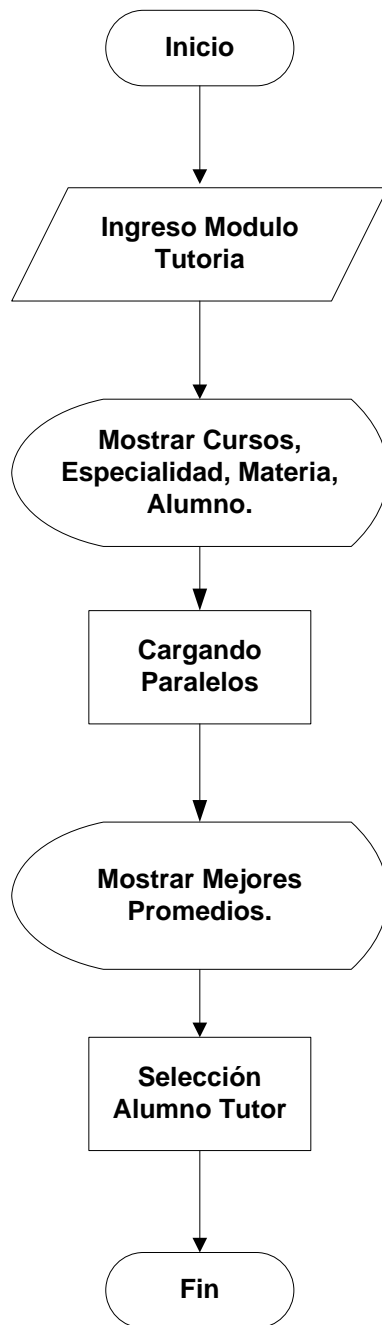


Figura #36: Modelo de Tutoría para Alumnos.

Fuente: Los autores.

3.7.3 Modelo de Promedio de Nota de Grado

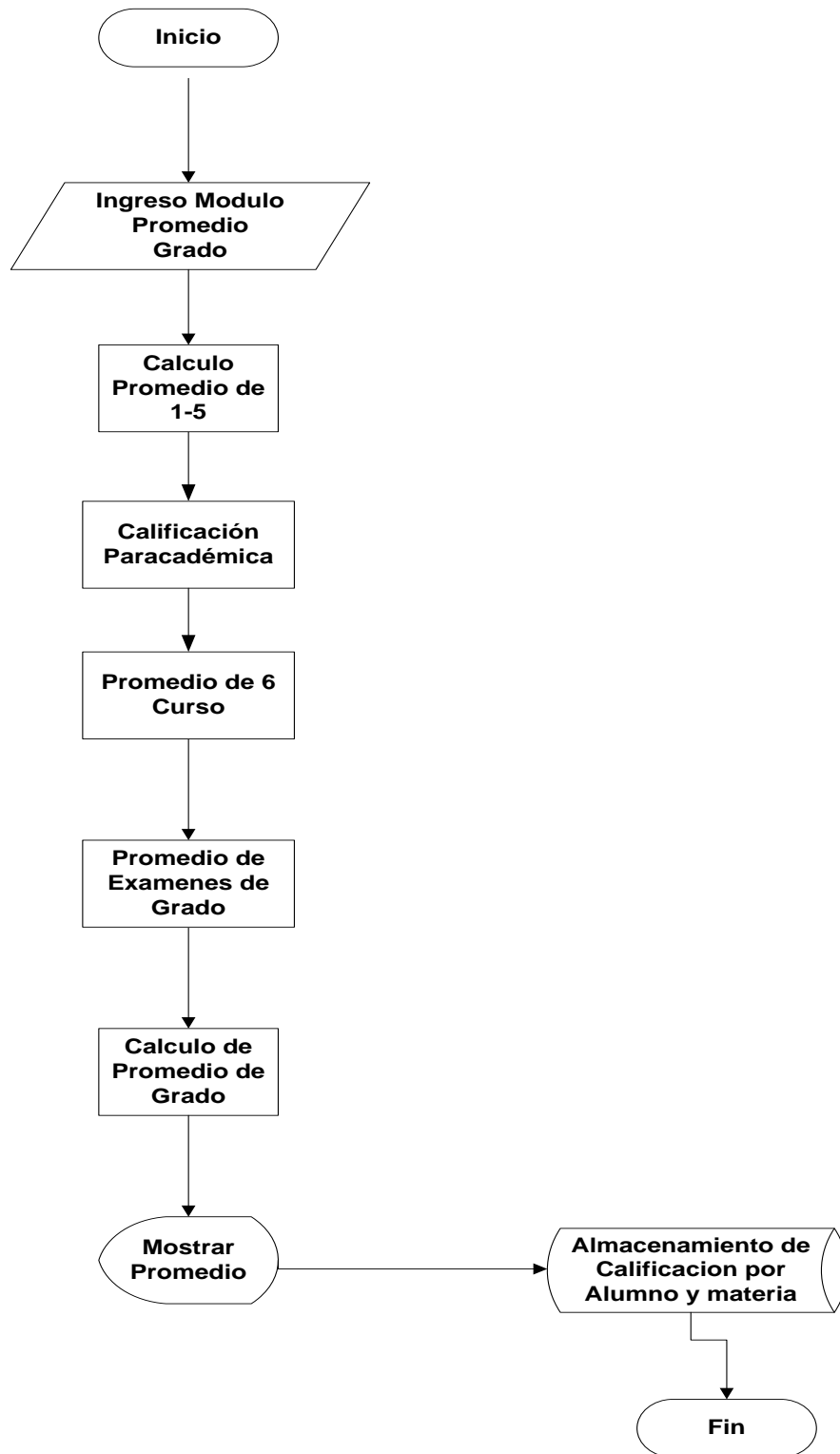


Figura #37: Modelo de Promedio de Nota de Grado.

Fuente: Los autores.

3.8 Modelo Entidad Relación

3.8.1 Diagrama de Calificación

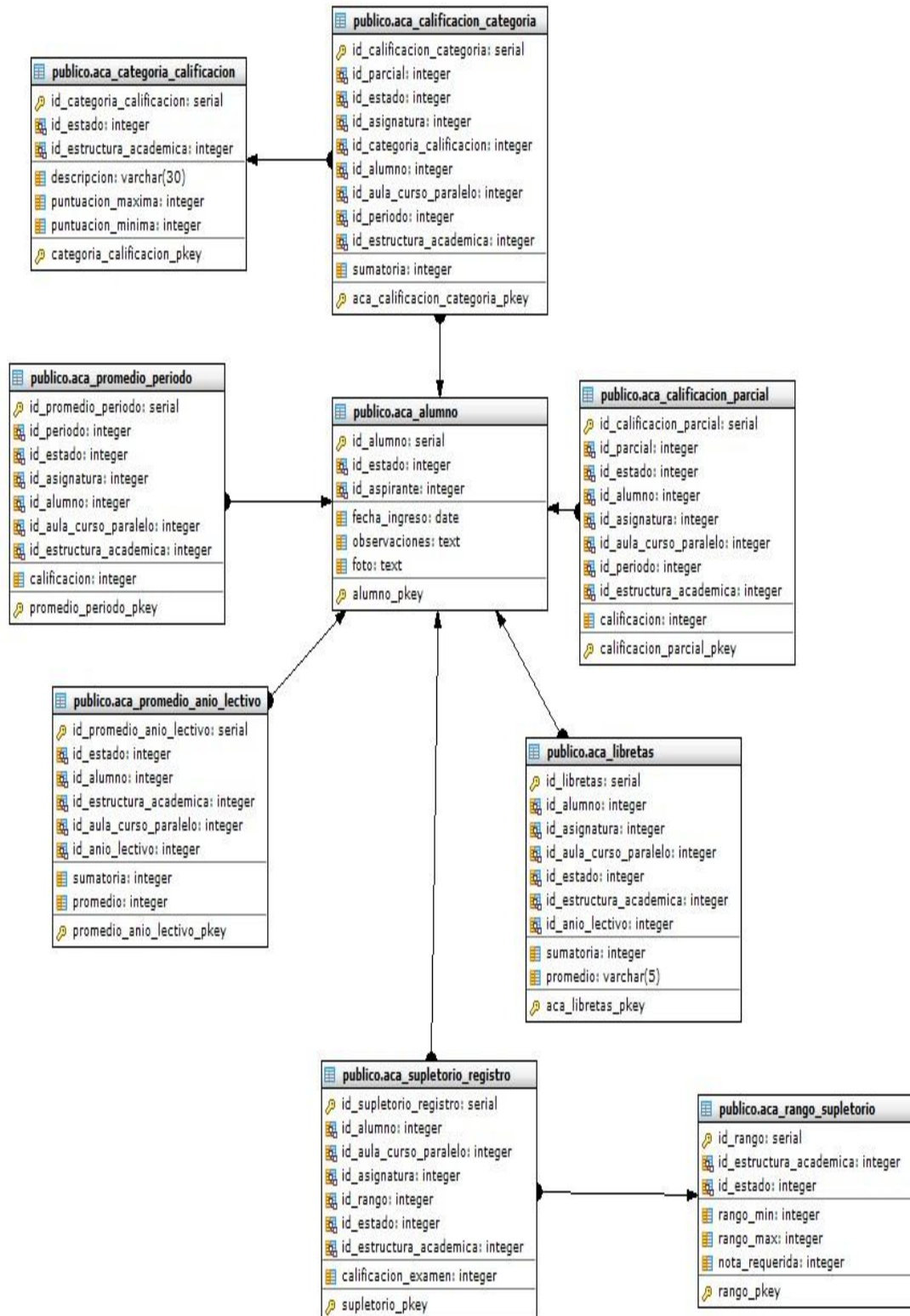


Figura #38: Modelo de entidad relación correspondiente al proceso de calificación.
Fuente: Los autores.

3.8.2 Diagrama de Promedio de Grado

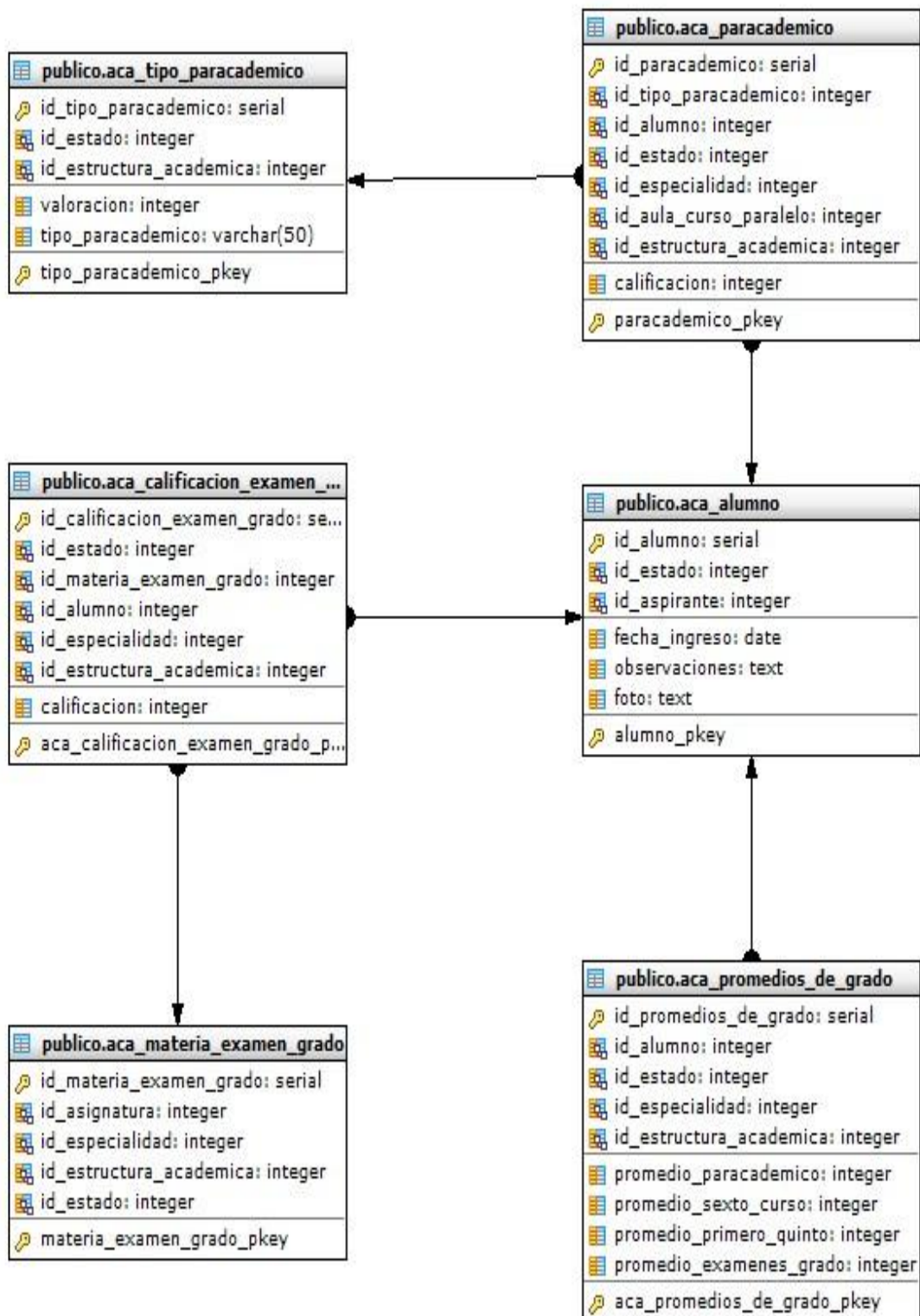


Figura #39: Diagrama entidad relación de promedio de grado.

Fuente: Los autores.

3.8.3 Diagrama de Supervisión y Tutoría

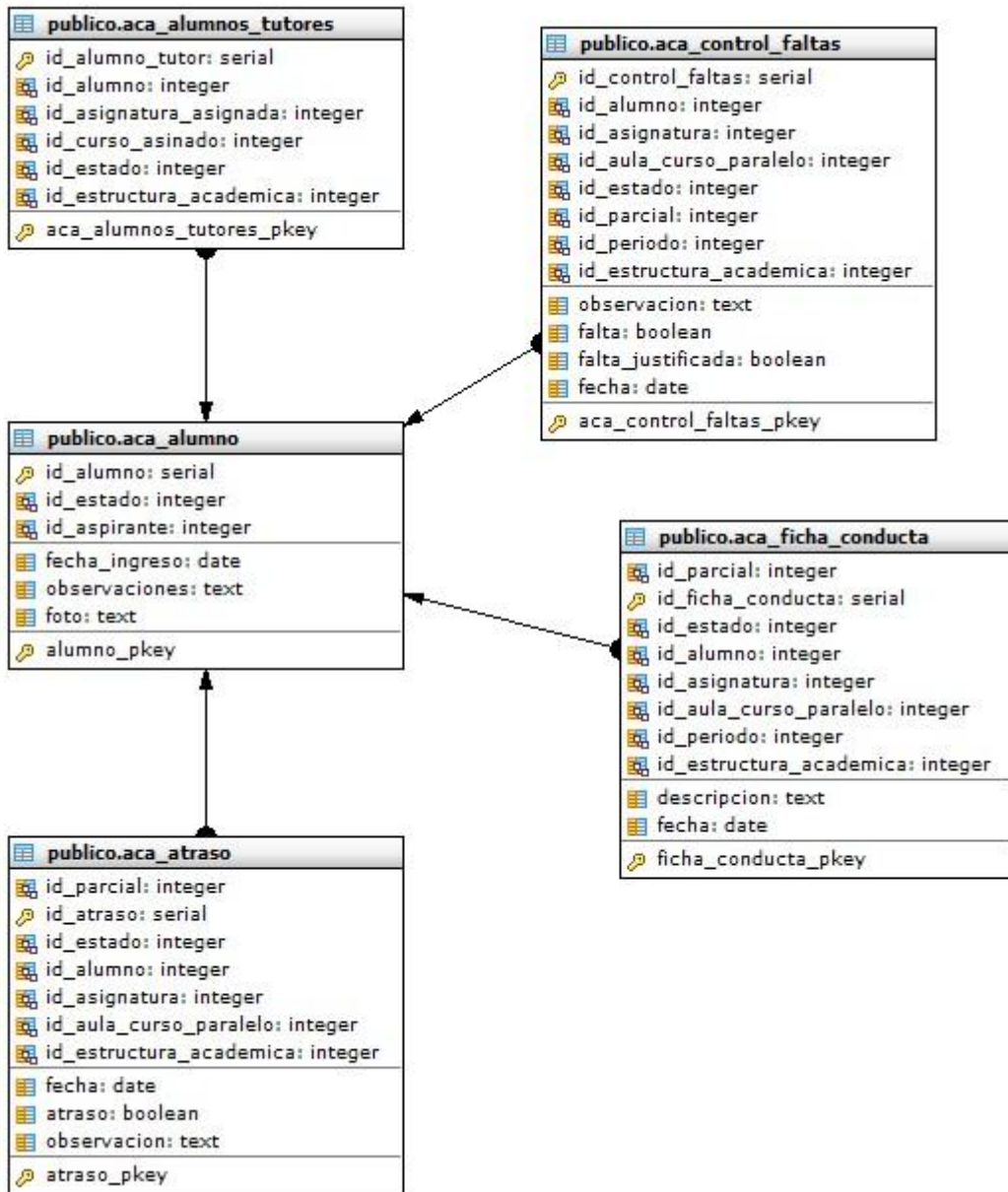


Figura #40: Diagrama entidad relación de supervisión y tutoría.

Fuente: Los autores.

3.9 Diccionario de Datos

Table "aca_atraso"					
Field	Data type	PK	FK	Not Null	Comment
id_parcial	integer		✓	✓	Clave Foranea de la tabla Parcial
id_atraso	serial	✓		✓	identifica el parcial
fecha	date			✓	Fecha en la que se registra el atraso.
atraso	boolean				Valor verdadero o falso
observacion	varchar(30)				Alguna observación particular.
id_anio_lectivo	integer		✓	✓	Clave Foranea de la tabla anio_lectivo
id_estado	integer		✓	✓	Clave Foranea de la tabla estado, muestra el estado del alumno
id_alumno	integer		✓	✓	Coódigo del alumno atrasado.
id_asignatura	integer		✓	✓	Código de materia en caso de atraso.
id_aula_curso_paralelo	integer		✓	✓	Clave Foranea de la tabla aula_curso_paralelo
id_estructura_academica	integer		✓	✓	Clave Foranea identifica la estructura académica vigente.

Tabla #12: Diccionario de datos de la tabla 'aca_atraso'

Fuente: Los autores

Table "aca_calificacion_categoria"					
Field	Data type	PK	FK	Not	Comment
id_calificacion_categoria	serial	✓		✓	Clave principal secuencial que identifica la calificación de la categoría.
id_parcial	integer		✓	✓	Clave foranea, identifica el parcial al que corresponde la categoría.
sumatoria	double precision			✓	Contiene la suma de las calificaciones de una misma categoría, identificadas por parcial.
id_anio_lectivo	integer		✓	✓	Identifica el año lectivo correspondiente.
id_estado	integer		✓	✓	Identifica el estado de calificacion_categoria.
id_asignatura	integer		✓	✓	Identifica la asignatura a la que corresponde la calificación de la categoría.
id_categoria_calificacion	integer		✓	✓	Identifica la categoría que se está calificando.
id_alumno	integer		✓	✓	Identifica el alumno al cual se está calificando.
id_aula_curso_paralelo	integer		✓	✓	Identificaa el aula, curso, paralelo al que pertenece el alumno.

Tabla #13: Diccionario de datos de la tabla 'aca_calificación_categoria'

Fuente: Los autores.

Table "aca_calificacion_parcial"

Field	Data type	PK	FK	Not Null	Comment
id_calificacion_parcial	serial	✓		✓	Clave principal secuencial, identifica la calificación parcial.
id_parcial	integer		✓	✓	Identifica el parcial correspondiente a calificar.
calificacion	double precision			✓	Representa la calificación del parcial.
id_anio_lectivo	integer		✓	✓	Identifica el código del año lectivo.
id_estado	integer		✓		Representa el estado de calificación parcial.
id_alumno	integer		✓	✓	Identifica el alumno a quien corresponde calificación.
id_asignatura	integer		✓	✓	Representa la asignatura que se esta calificando.
id_aula_curso_paralelo	integer		✓	✓	Codigo representa el aula, curso y paralelo en el que se encuentra el alumno.
id_periodo	integer		✓	✓	Identifica el periodo en el cual se encuentra el alumno.
id_estructura_académica	integer		✓	✓	Registra el sistema de calificación actual y en uso.

Tabla #14: Diccionario de datos de la tabla 'aca_calificación_parcial'
Fuente: Los autores.

Table "aca_categoria_calificacion"

Field	Data type	PK	FK	Not Null	Comment
id_categoria_calificacion	serial	✓		✓	Código que representa la categoría calificación.
descripcion	varchar(30)			✓	Detalle de la categoría respectiva.
id_anio_lectivo	integer		✓	✓	Identifica el año lectivo.
puntuacion_maxima	double precision				Puntuación máxima que puede tener la categoría.
puntuacion_minima	double precision				Puntuación mínima que puede tener la categoría.
id_estado	integer		✓		Representa el estado de categoría calificación.
id_estructura_académica	integer		✓	✓	Registra el sistema de calificación actual y en uso.

Tabla #15: Diccionario de datos de la tabla 'aca_categoria_calificación'
Fuente: Los autores

Table "aca_control_faltas"					
Field	Data type	PK	FK	Not	Comment
id_control_faltas	integer	✓		✓	Código secuencial que representa el control de faltas.
id_alumno	integer		✓	✓	Identifica el alumno a quien se le asigna falta.
observacion	text			✓	Un detalle de la falta registrada.
id_asignatura	integer		✓	✓	Asignatura en la que se registra la falta/fuga.
id_aula_curso_paralelo	integer		✓	✓	Aula, curso y paralelo al que pertenece el alumno.
falta	boolean			✓	Registra la falta con tipo verdadero o falso.
falta_justificada	boolean			✓	Registra si la falta ha sido justificada con verdadero o falso.
id_anio_lectivo	integer		✓	✓	Identifica el año lectivo vigente.
id_estado	integer		✓	✓	Indica el estado de control de faltas.
fecha	date			✓	Describe la fecha en la que se registra la falta.
id_parcial	integer		✓	✓	Registra el parcial en el cual se comete la falta.
id_periodo	integer		✓		Registra el período en el cual se comete la falta.

Tabla #16: Diccionario de datos de la tabla 'aca_control_faltas'

Fuente: Los autores

Table "aca_ficha_conducta"

Field	Data type	PK	FK	Not	Comment
id_parcial	integer		✓	✓	Registra el parcial en el cual se apertura la ficha_conducta.
descripcion	varchar(30)			✓	Descripción de la ficha_conducta.
id_estructura_academica	integer		✓	✓	Identifica la estructura académica vigente.
id_ficha_conducta	serial	✓		✓	Código principal y secuencial que representa cada registro en ficha_conducta.
fecha	date			✓	Fecha en la que se registra la ficha_conducta.
id_estado	integer		✓	✓	Estado de la ficha_conducta.
id_alumno	integer		✓	✓	Identifica el alumno a quien se le abre la ficha médica.
id_asignatura	integer		✓	✓	Corresponde a la materia en la que se abre la ficha.
id_aula_curso_paralelo	integer		✓	✓	Identifica el aula, curso y paralelo al cual pertenece el alumno.
id_periodo	integer		✓	✓	Registra el periodo en el cual se registra la ficha.

Tabla #17: Diccionario de datos de la tabla 'aca_ficha_conducta'

Fuente: Los autores.

Table "aca_materia_examen_grado"

Field	Data type	PK	FK	Not	Comment
id_materia_examen_grado	serial	✓		✓	Identifica cada registro de forma exclusiva y secuencial.
id_asignatura	integer		✓	✓	Identifica la materia a evaluar en examen de grado.
id_anio_lectivo	integer		✓	✓	Registra el código del año lectivo vigente.
id_especialidad	integer		✓	✓	Identifica la especialidad a la cual corresponde la materia.
id_estructura_academica	integer		✓	✓	Identifica la jornada, la entidad.
id_estado	integer		✓	✓	Registra el estado de aca_materia_examen_grado.

Tabla #18: Diccionario de datos de la tabla 'aca_materia_examen_grado'

Fuente: Los autores.

Table "aca_paracademico"					
Field	Data	PK	FK	Not	Comment
id_paracademico	serial	✓		✓	Identifica cada registro de forma secuencial.
id_tipo_paracademico	integer		✓	✓	Identifica el código del tipo paracademico que se esta evaluando.
id_alumno	integer		✓	✓	Registra el código del alumno que se esta calificando el paracademico.
calificacion	Integer			✓	Almacena la calificación de paracademico.
id_estado	integer		✓	✓	Identifica el estado de aca_paracademico.
id_aula_curso_parallel	integer		✓	✓	Registra el aula, curso y paralelo al que pertenece el alumno.
id_estructura_academica	integer		✓	✓	Identifica la estructura académica vigente.
id_especialidad	integer		✓	✓	Identifica la especialidad del alumno.

Tabla #19: Diccionario de datos de la tabla 'aca_paracademico'
Fuente: Los autores.

Table "aca_promedio_anio_lectivo"					
Field	Data	PK	FK	Not	Comment
id_promedio_anio_lectivo	serial	✓		✓	Identifica cada registro de forma única.
sumatoria	Integer			✓	Almacena la sumatoria de los promedios correspondiente a cada periodo, para poder validar si esta aprobado, suspenso, reprobado.
promedio	Integer			✓	Almacena promedio general del alumno por año lectivo, dato requerido para el promedio general de grado. Justificación: soporta el ingreso de la nota si el alumno va a ingresar y no tiene historico en la institucion.
id_estado	integer		✓	✓	este codigo debe identificar el alumno que esta en supletorio, aprobado, reprobado.
id_alumno	integer		✓	✓	Codigo que identifica el alumno a quien corresponde el promedio anual.
id_estructura_academica	integer		✓	✓	Identifica la estructura vigente en el año lectivo.
id_aula_curso_parallel	integer		✓	✓	Identifica el codigo del aula, curso y paralelo del alumno.
id_anio_lectivo	integer		✓	✓	Identifica el año lectivo vigente.

Tabla #20: Diccionario de datos de la tabla 'aca_promedio_anio_lectivo'
Fuente: Los autores

Table "aca_promedio_grado"					
Field	Data type	PK	FK	Not	Comment
id_promedio_de_grado	serial	✓		✓	Esta tabla almacena la calificación de los promedios de acuerdo a la categoría_promedio_grado.
calificacion	double precision			✓	Calificación correspondiente al promedio de grado.
promedio_paracademico	integer			✓	Calificación correspondiente al promedio paracadémico.
id_estado	integer		✓	✓	Identifica el estado de aca_promedio_grado.
id_alumno	integer		✓	✓	Identifica el alumno a quien corresponde el promedio de grado.
id_especialidad	integer		✓	✓	Identifica la especialidad del alumno.
promedio_sexto_curso	integer			✓	Representa la calificación correspondiente al promedio de sexto curso.
id_estructura_academica	integer		✓	✓	Identifica la estructura académica vigente.
promedio_primer_quinto	integer			✓	Calificación correspondiente al promedio de primer a quinto curso.
promedio_exámenes_grado	integer			✓	Calificación correspondiente al promedio de los exámenes de grado.

Tabla #21: Diccionario de datos de la tabla 'aca_promedio_grado'

Fuente: Los autores.

Table "aca_promedio_período"

Field	Data type	PK	FK	Not	Comment
id_promedio_período	serial	✓		✓	Identifica cada registro en la tabla aca_promedio_período
id_período	integer		✓	✓	Corresponde al código del período al que corresponde el promedio.
calificacion	double precision			✓	Registra la calificación del promedio del período.
id_año_lectivo	integer		✓	✓	Identifica el código del año lectivo.
id_estado	integer		✓	✓	Identifica el estado de aca_promedio_período.
id_asignatura	integer		✓	✓	Identifica la asignatura a la que corresponde el promedio.
id_alumno	integer		✓	✓	Identifica el código del alumno al que corresponde el promedio del período.
id_aula_curso_paralelo	integer		✓	✓	Corresponde al aula, curso, paralelo en el cual esta el alumno.
id_estructura_academica	integer		✓	✓	Identifica la estructura vigente en el período.

Tabla #22: Diccionario de datos de la tabla 'aca_promedio_período'
Fuente: Los autores.

Table "aca_rango_supletorio"

Field	Data type	PK	FK	Not	Comment
id_rango	serial	✓		✓	Registra el valor secuencial de cada registro en la tabla.
rango_min	integer			✓	Registra la calificación mínima para un determinado rango.
rango_max	integer			✓	Registra la calificación máxima para un determinado rango.
nota_requerida	integer			✓	Almacena la nota requerida para aprobar, esta nota depende del rango.
id_año_lectivo	integer		✓	✓	Identifica el año lectivo vigente.
id_estructura_academica	integer		✓	✓	Registra la entidad y sistema de calificación vigente.
id_estado	integer		✓	✓	Identifica el estado aca_rango_supletorio.

Tabla #23: Diccionario de datos de la tabla 'aca_rango_supletorio'
Fuente: Los autores.

Table "aca_supletorio_registro"					
Field	Data type	PK	FK	Not	Comment
id_supletorio_registro	serial	✓		✓	Clave serial que identifica cada registro de forma exclusiva.
id_alumno	integer		✓	✓	Identifica el alumno en supletorio.
id_aula_curso_paralelo	integer		✓	✓	Identifica el aula, curso y paralelo del alumno en supletorio.
id_asignatura	integer		✓	✓	Identifica la materia implicada en supletorio.
id_rango	integer		✓	✓	Identifica el código del rango, para validar cuanto necesita obtener el alumno para aprobar.
id_estado	integer		✓	✓	Registra el estado de aca_supletorio_registro.
id_anio_lectivo	integer		✓	✓	Identifica el año lectivo vigente.

Tabla #24: Diccionario de datos de la tabla 'aca_supletorio_registro'

Fuente: Los autores.

Table "aca_tipo_paracademico"					
Field	Data type	PK	FK	Not	Comment
id_tipo_paracademico	serial	✓		✓	Valor secuencial que identifica cada registro.
id_estructura_academica	integer		✓	✓	Identifica el sistema de calificación y entidad.
valoracion	double precision			✓	Registra la calificación
id_anio_lectivo	integer		✓	✓	Registra el año lectivo vigente.
tipo_paracademico	varchar(50)			✓	Registra el tipo de paracademico ingresado
id_estado	integer		✓	✓	Identifica el estado de aca-tipo_paracademico.

Tabla #25: Diccionario de datos de la tabla 'aca_tipo_paracademico'

Fuente: Los autores.

Table "aca_alumnos_tutores"

Field	Data	PK	FK	Not	Comment
id_alumno_tutor	serial	✓		✓	Clave principal secuencial que identifica cada registro.
id_alumno	integer		✓	✓	codigo que identifica al alumno.
id_asignatura_asignada	integer		✓		codigo que representa la materia a la cual se designa para tutoria.
id_curso_asinado	integer		✓		Representa el curso al cual se le va a dar tutoría.
id_estado	integer		✓	✓	Identifica el estado de aca_alumnos_tutores.
id_estructura_academica	integer		✓	✓	Identifica la estructura académica vigente.

Tabla #26: Diccionario de datos de la tabla 'aca_alumnos_tutores'

Fuente: Los autores.

Table "aca_libretas"

Field	Data type	PK	FK	Not Null	Comment
id_libretas	serial	✓		✓	Codigo secuencial que identifica cada registro de forma única.
id_alumno	integer		✓	✓	Identifica el código del alumno a quien corresponde la libreta.
id_asignatura	integer	✓	✓		Codigo de asignatura que va registrado en libreta.
id_aula_curso_paralelo	integer		✓		Registra el aula, curso y paralelo a quien corresponde la libreta.
sumatoria	integer	✓	✓		Registra la suma los promedios por periodo.
id_estado	integer		✓		Estado de aca_libretas
promedio	integer	✓	✓		Identifica el promedio general del año lectivo.
id_estructura_academica	integer	✓			Identifica el sistema de calificación al que pertenece la estructura académica.
id_anio_lectivo	integer				Identifica el año lectivo vigente.

Tabla #27: Diccionario de datos de la tabla 'aca_libretas'

Fuente: Los autores.

Table "aca_calificación_examen_grado"					
Field	Data type	PK	FK	Not	Comment
id_calificacion_examen_grado	serial	✓		✓	Registra el código secuencial de cada registro en aca_calificación_examen_grado.
calificacion	integer			✓	Registra la calificación como tal.
id_estado	integer		✓	✓	Almacena el estado de la tabla aca_calificación_examen_grado.
id_materia_examen_grado	integer		✓	✓	Identifica la materia que se califica en el examen de grado.
id_alumno	integer		✓	✓	Registra el código del alumno.
id_especialidad	integer		✓	✓	Identifica el código de la especialidad en la que se encuentra el alumno.
id_estructura_academica	integer		✓	✓	Identifica la estructura vigente en el año lectivo que se esta utilizando.

Tabla #28: Diccionario de datos de la tabla 'aca_calificación_examen_grado'

Fuente: Los autores.

3.10 Funciones y Trigger

Este punto tiene que ser desarrollado por los autores.

3.10.1 Función “publico.insertarpromedioexamengrado()”

Esta función según el año lectivo, el estado del alumno, la especialidad, me permite recorrer las calificaciones de cada examen de grado realizado por un determinado estudiante y obtener el promedio correspondiente.

Adicionalmente la función tiene embebido un disparador, el mismo que se activa al momento de realiza un insert (Agregar) o update (actualización) en la tabla aca_calificación_examen_grado.

```
-- Function: publico.insertarpromedioexamengrado()
-- DROP FUNCTION publico.insertarpromedioexamengrado();

CREATE OR REPLACE FUNCTION publico.insertarpromedioexamengrado()
  RETURNS trigger AS
$BODY$
DECLARE
  v_id_alumno integer = new.id_alumno;
  v_id_anio_lectivo integer = new.id_anio_lectivo;
  suma double precision;
  num_calificaciones integer;
  promedio double precision;

  cont integer=0;

  cursorCalificacion CURSOR FOR Select distinct (id_alumno),id_anio_lectivo,
id_estado, id_especialidad from publico.aca_calificacion_examen_grado
                                where id_anio_lectivo = v_id_anio_lectivo
                                and id_alumno = v_id_alumno;

BEGIN

      FOR recCalif IN cursorCalificacion LOOP

          Select count(*) into num_calificaciones from
publico.aca_calificacion_examen_grado
```

```

        where id_anio_lectivo = recCalif.id_anio_lectivo
        and id_alumno = recCalif.id_alumno;

        Select sum(calificacion) into suma from
publico.aca_calificacion_examen_grado
        where id_anio_lectivo = recCalif.id_anio_lectivo
        and id_alumno = recCalif.id_alumno;

        promedio = suma / num_calificaciones;

        SELECT count(*) into cont
        FROM publico.aca_promedios_de_grado
        where id_anio_lectivo = recCalif.id_anio_lectivo
        and id_alumno = recCalif.id_alumno;

        IF (cont = 0)
        THEN
            insert into
publico.aca_promedios_de_grado(id_alumno, id_especialidad, id_anio_lectivo,
id_estado, promedio_paracademico)
            values(recCalif.id_alumno, recCalif.id_especialidad,
recCalif.id_anio_lectivo, recCalif.id_estado, promedio);
        ELSE
            update publico.aca_promedios_de_grado
            SET promedio_examenes_grado = promedio
            where id_anio_lectivo = recCalif.id_anio_lectivo
            and id_alumno = recCalif.id_alumno;
        END IF;

        END LOOP;

        RETURN null;
    END;
$BODY$
    LANGUAGE plpgsql VOLATILE
    COST 100;
ALTER FUNCTION publico.insertarpromedioexamengrado() OWNER TO
sistema_siga;

```

Tabla #29: Ejemplo de función para insertar el promedio examen de grado.

Fuente: Los autores.

3.10.2 Job

Para el funcionamiento de este proyecto se ha desarrollado dos JOB en los cuales se programara la activación y desactivación automáticas del estado de período para el módulo de calificación.

Recomendación.- las fechas deben estar dentro del período para el cual fue programado el Job, de lo contrario no funcionaria.

Para la creación y configuración de Job primero debemos crear un servicio llamado **PgAgent**, para lo cual se debe contar con los siguientes archivos:



Figura #41: Archivos necesarios para creación y configuración de un job.

Fuente: Los autores.

Los archivos arriba mencionados pueden ser descargados de la web de postgresQL. Una vez descargados los archivos deberán ser copiados al directorio de instalación de postgresQL como por ejemplo:

C:\Archivos de Programa\PostgreSQL\8.4\bin

Luego se procede ejecutar la siguiente consulta en la base de datos 'postgres'. La consulta es el contenido del archivo PgAgent, el cual debe ser abierto desde el programa 'PgAdmin III.

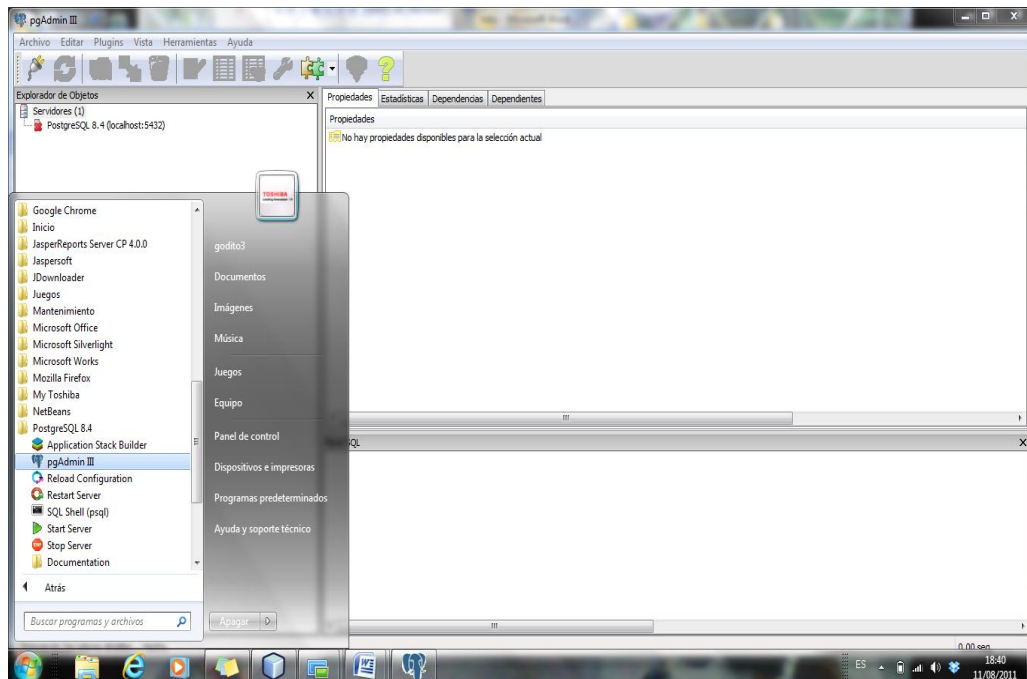
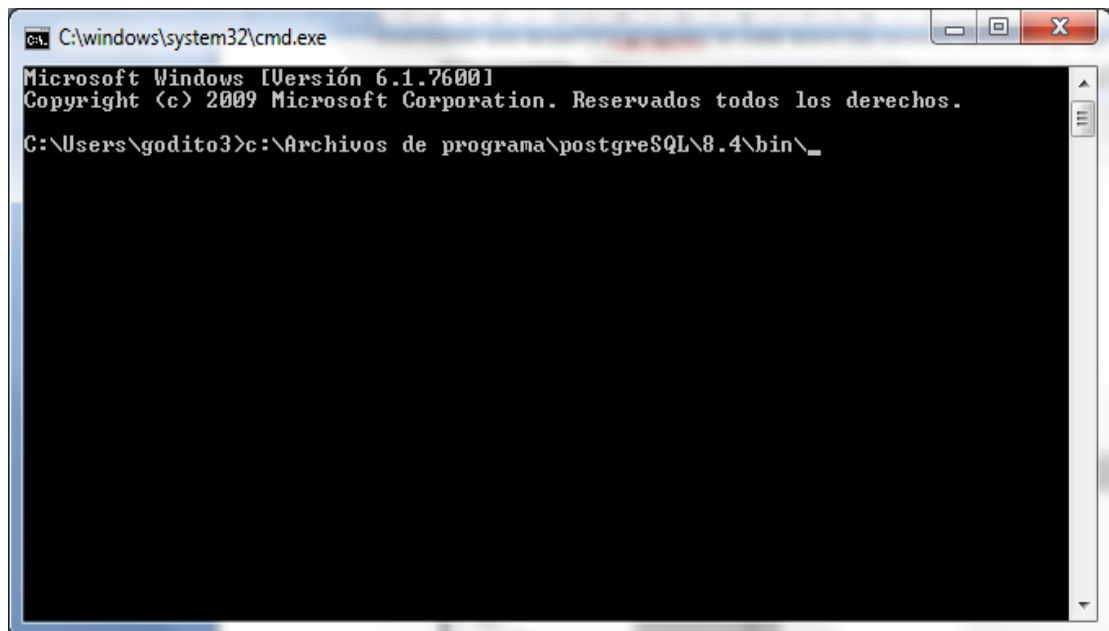


Figura #42: Ejecución de consulta para la base de datos.

Fuente: Los autores.

A continuación se ingreso al siguiente directorio a través de consola

c:\Archivos de programa\postgresql\8.4\bin



En esa misma ventana de comandos, una vez ingresado al directorio anteriormente mostrado, se instala el servicio pgAgent con la siguiente línea de comando:

```
pgAgent.exe INSTALL pgAgent -u postgres -p admin -l2
hostaddr=localhost dbname=postgres user=postgres
password=admin
```

En dbname va el nombre de la base de datos donde se va a instalar pgAgent, la cual sería la base 'postgres', y asimismo, en user, va el usuario postgres, y la contraseña 'admin'.

Una vez terminada la instalación del pgAgent, se procede a abrir el programa 'PgAdmin III' de PostgreSQL 8.4.

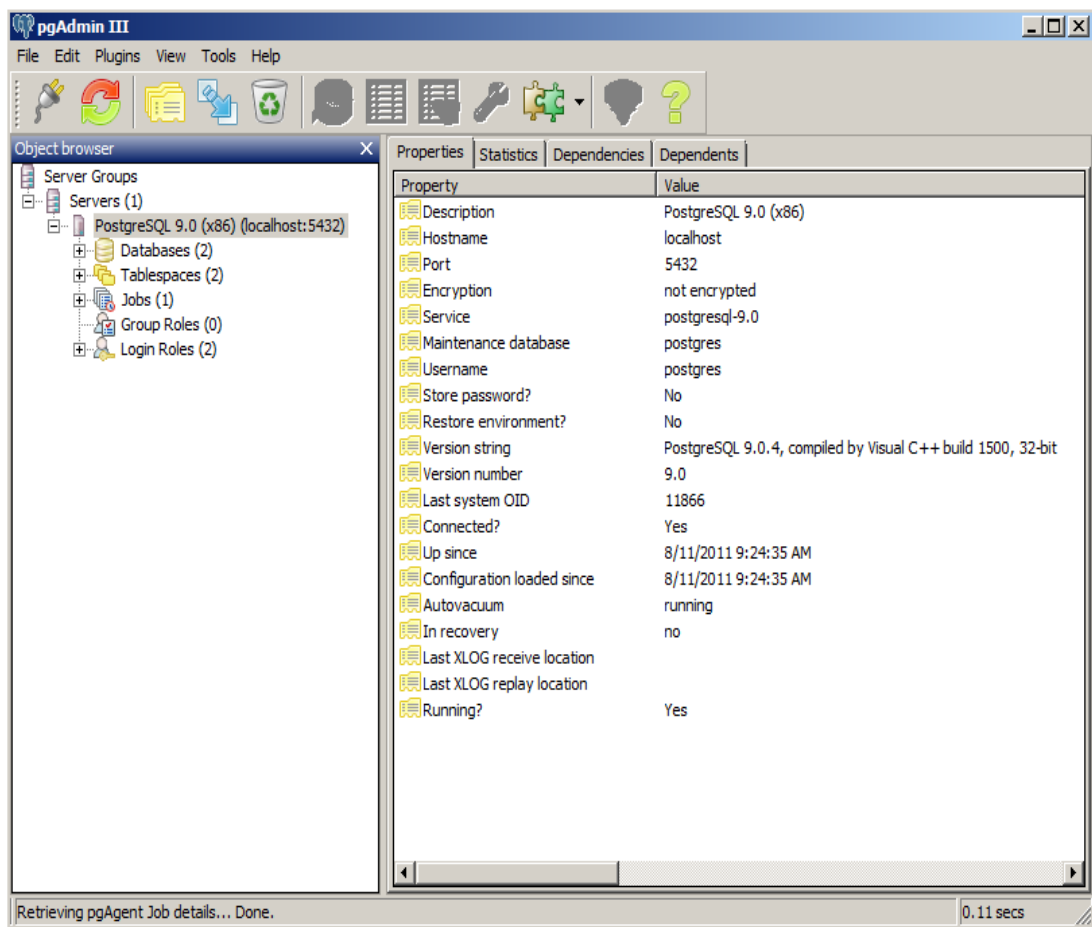


Figura #43: Muestra la pantalla inicial cuando se abre PgAdmin III.

Fuente: Los autores.

En el servidor PostgreSQL 8.4, procedemos a crear los jobs.

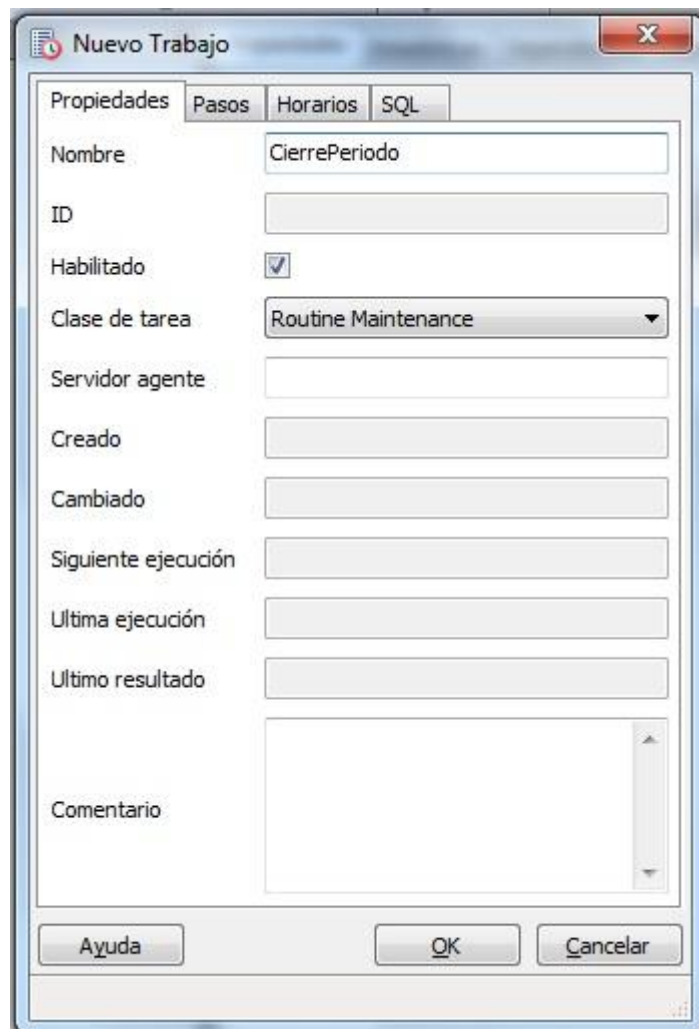
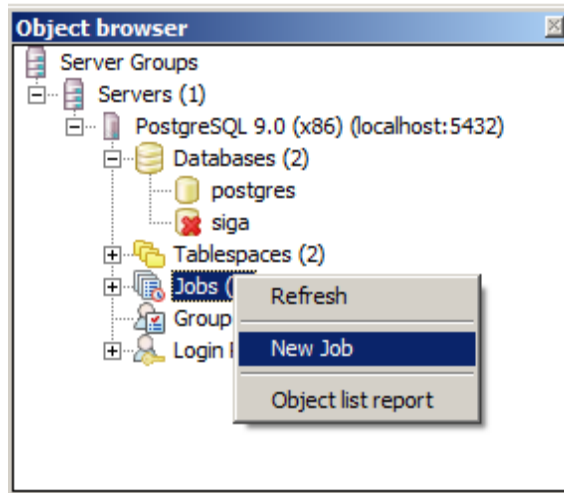
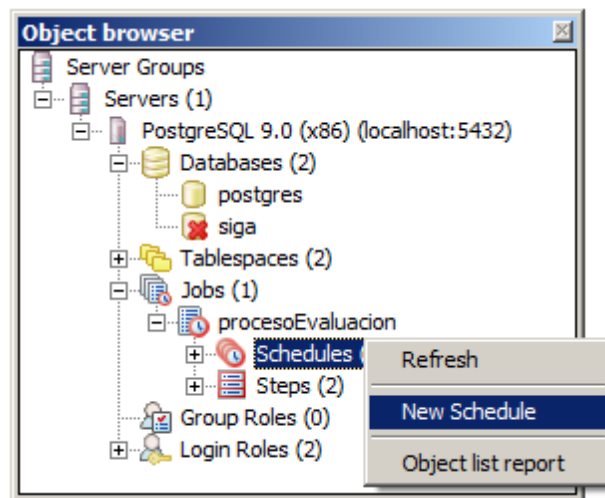


Figura #44: Ventana que muestra los parámetros de configuración de un job.

Fuente: Los autores.

El nombre que le damos al Job, será 'CierrePeríodo', y le damos click en el botón 'OK'.



Luego se procede a crear un Schedule, para programar el fecha de activación y desactivación del período en el cual funcionarán los 'Steps' (pasos), todo esto manejado por el Job.

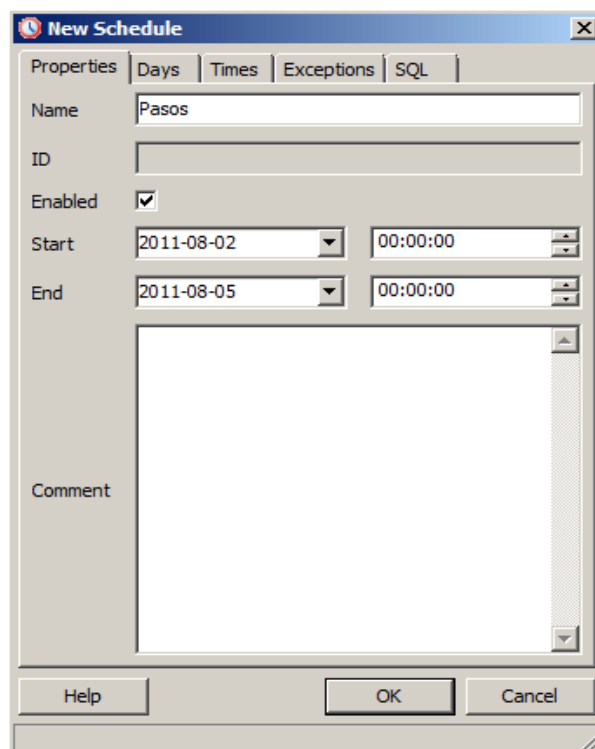
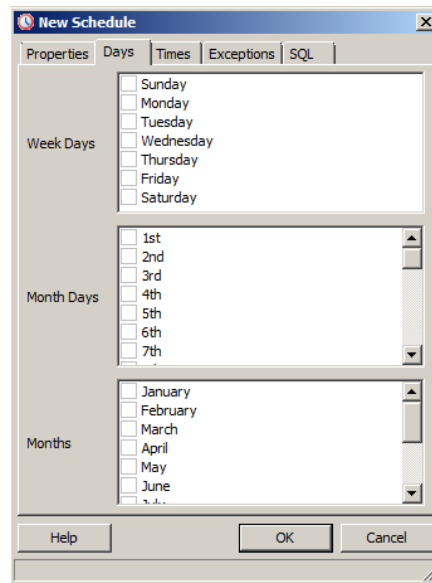


Figura #45: Muestra los parámetros de fecha de un job.

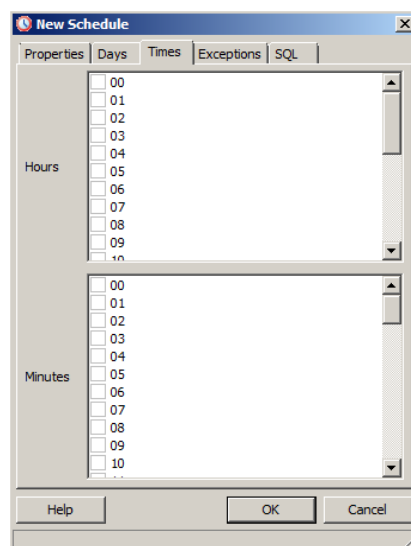
Fuente: Los autores.

Aquí se le da la fecha inicial, y la fecha final para cuando funcionarán los procesos automáticos, manejados por dicho Job.



Se selecciona que días de la semana y del mes, y qué meses deseamos que funcionen los procesos automáticos. Aunque para éste proyecto se seleccionan todos los días de la semana y del mes, y todos los meses.

Así mismo seleccionamos horas y minutos para los cuales deseamos que funcionen



estos procesos. Al terminar todo lo anterior, damos click en el botón OK, y se configura el 'Schedule'. Por lo general en las horas y minutos se seleccionan '00' para que se ejecuten los procesos desde tiempo 0.

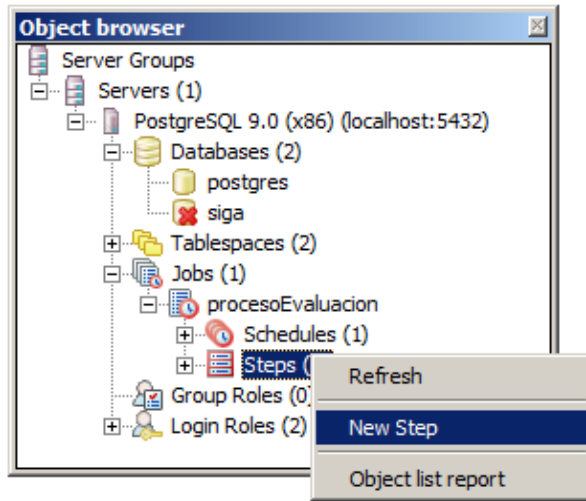
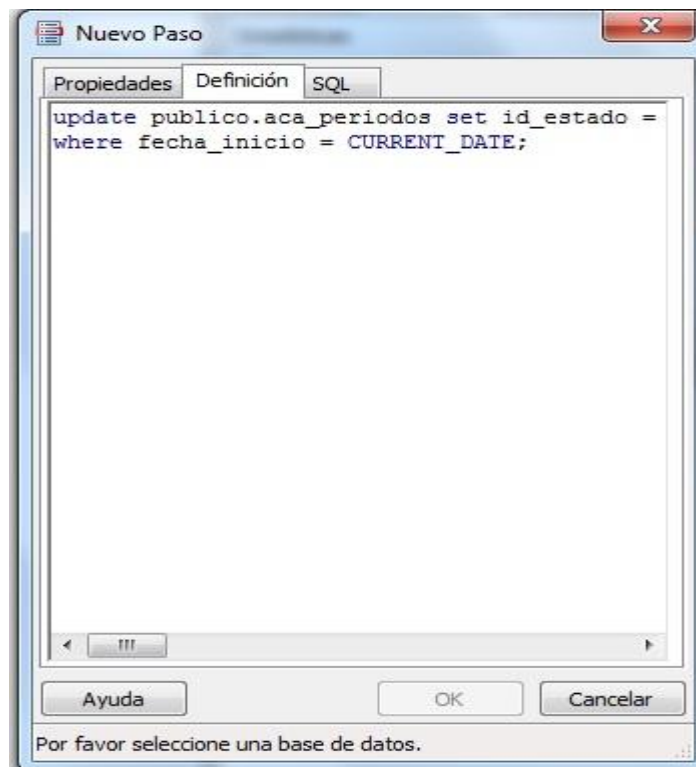


Figura #46: Muestra la ventana para creación de un nuevo paso.

Fuente: Los autores.

Una vez configurado el Schedule, procedemos a programar los Step (pasos), los cuales servirían para la activación y desactivación automática de cierre de período. En las siguientes ventanas se mostrará cómo crear los pasos de activación y desactivación de período.



En la definición se inserta la consulta SQL deseada. En este caso, es una consulta de actualización, la cual nos indica que el período estará activo en un rango de fechas, definido por el usuario en el sistema.

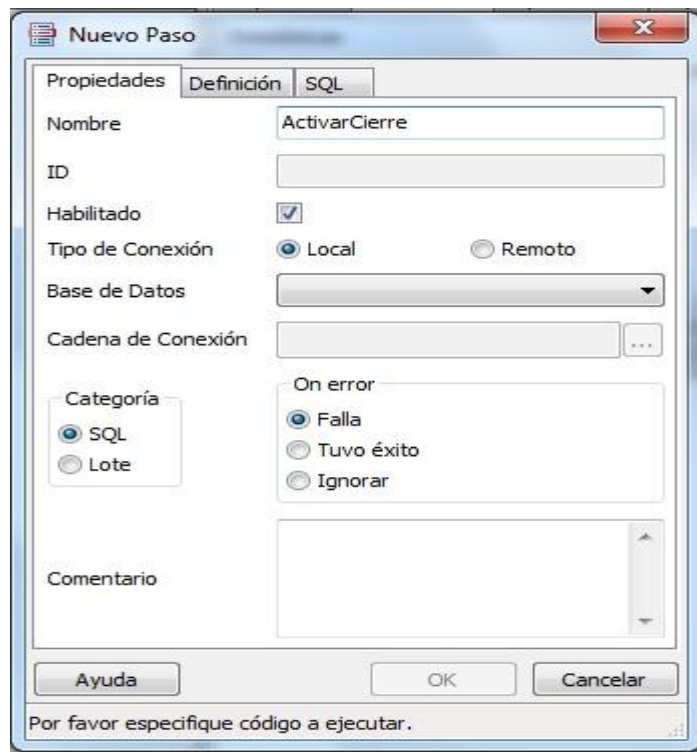
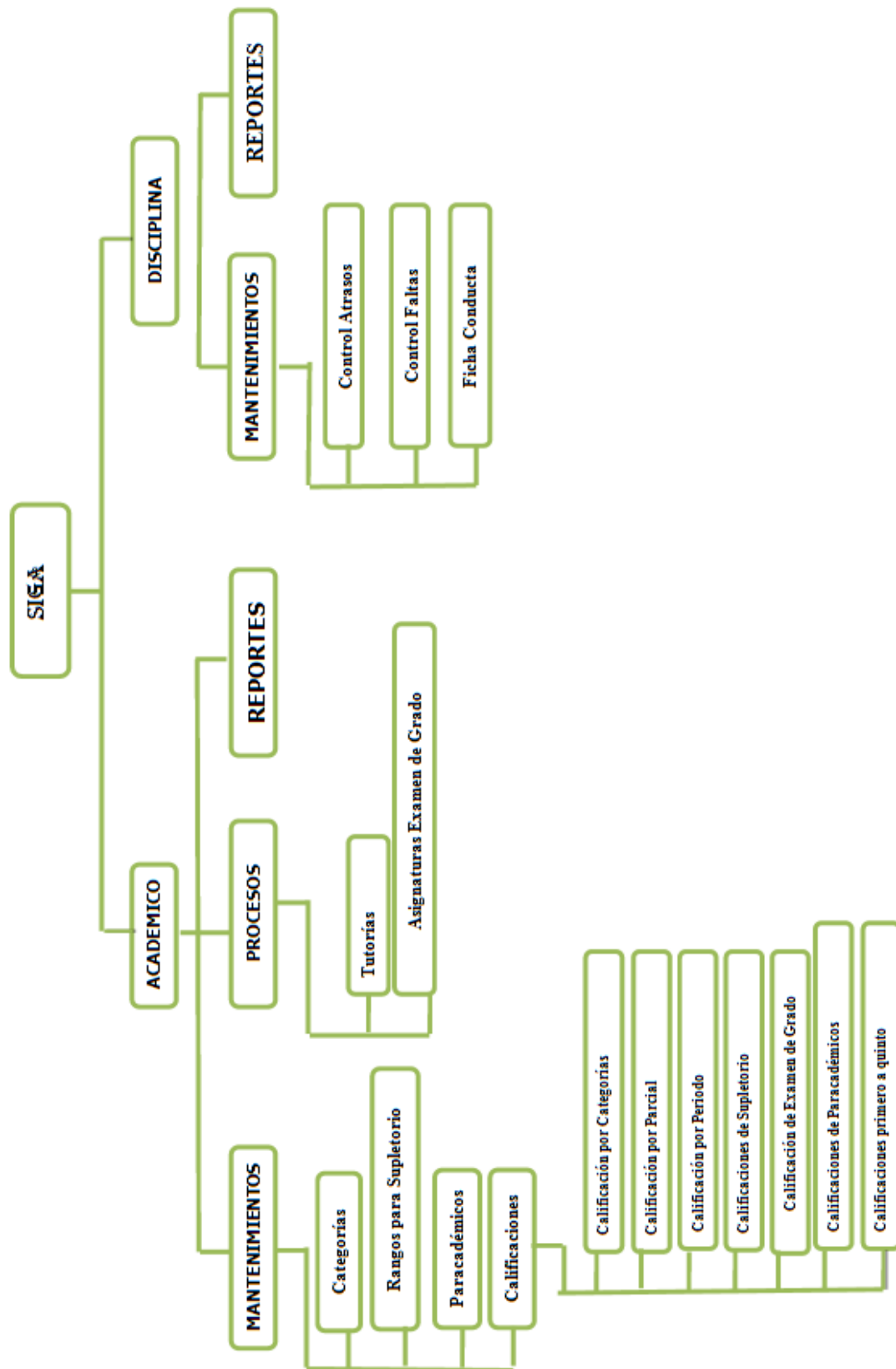


Figura #47: Se especifica el código a ejecutar el nuevo paso.

Fuente: Los autores.

3.11 Diagrama Jerárquico de la aplicación.



MANTENIMIENTOS ACADEMICOS

Nivel 1._



Tabla #30: Muestra el nivel de mantenimiento académico.

Fuente: Los autores.

Nivel 2._

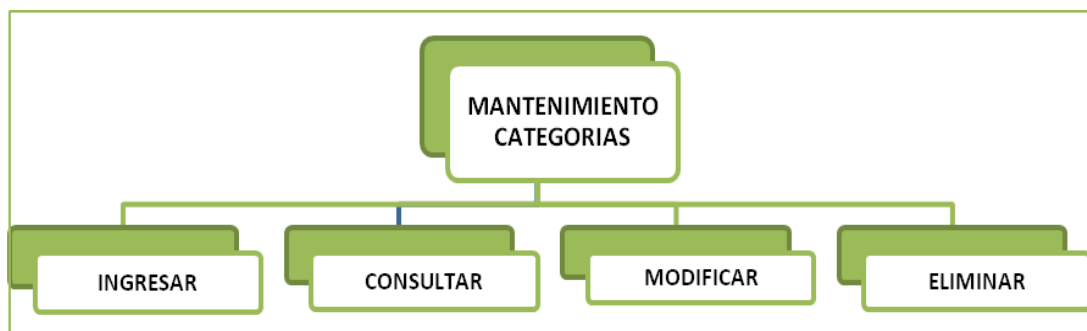




Tabla #31: Muestra el nivel de mantenimiento de categorías y calificaciones.

Fuente: Los autores.

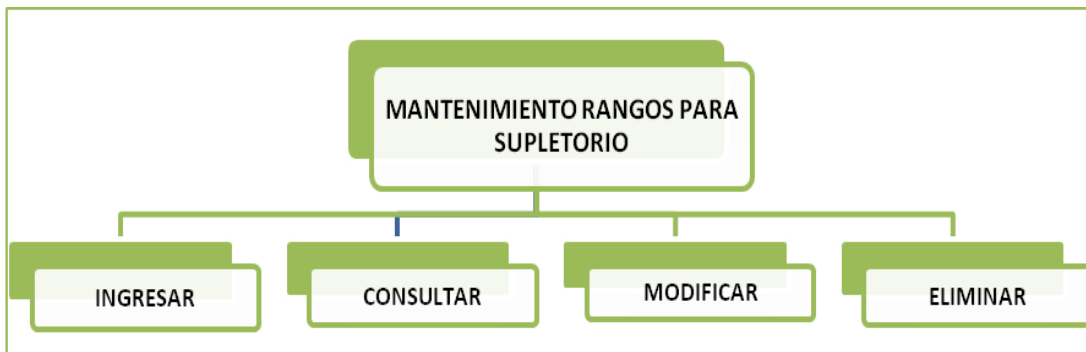
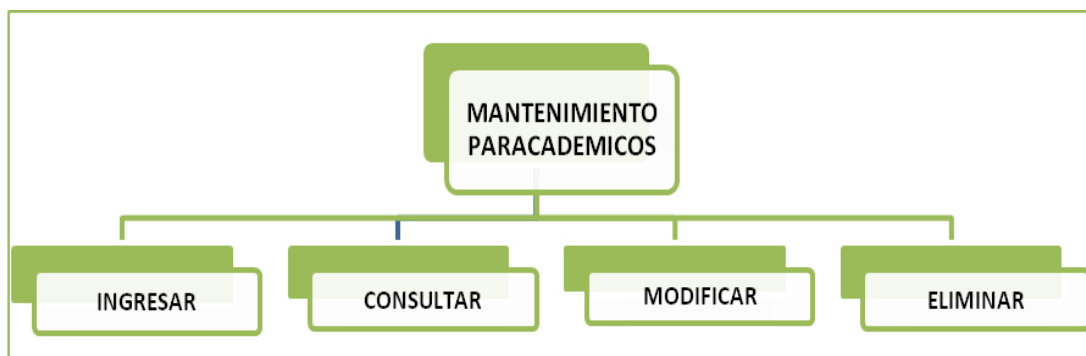


Tabla #32: Muestra el nivel de paracadémico y rango supletorio.

Fuente: Los autores.

MANTENIMIENTOS DISCIPLINA

Nivel 1._



Nivel 2._

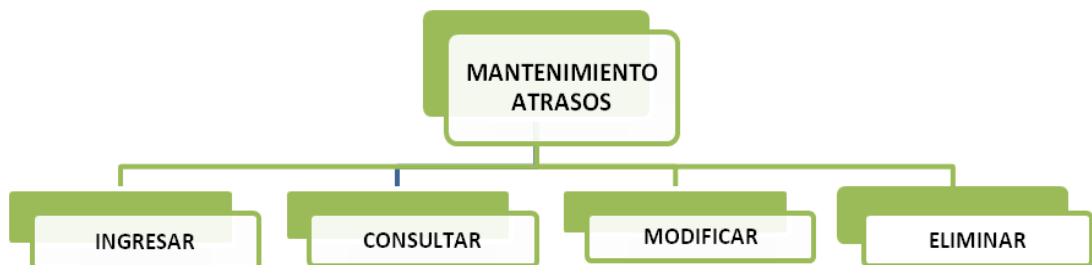


Tabla #33: Muestra opciones para el mantenimiento de disciplina y atrasos.

Fuente: Los autores.

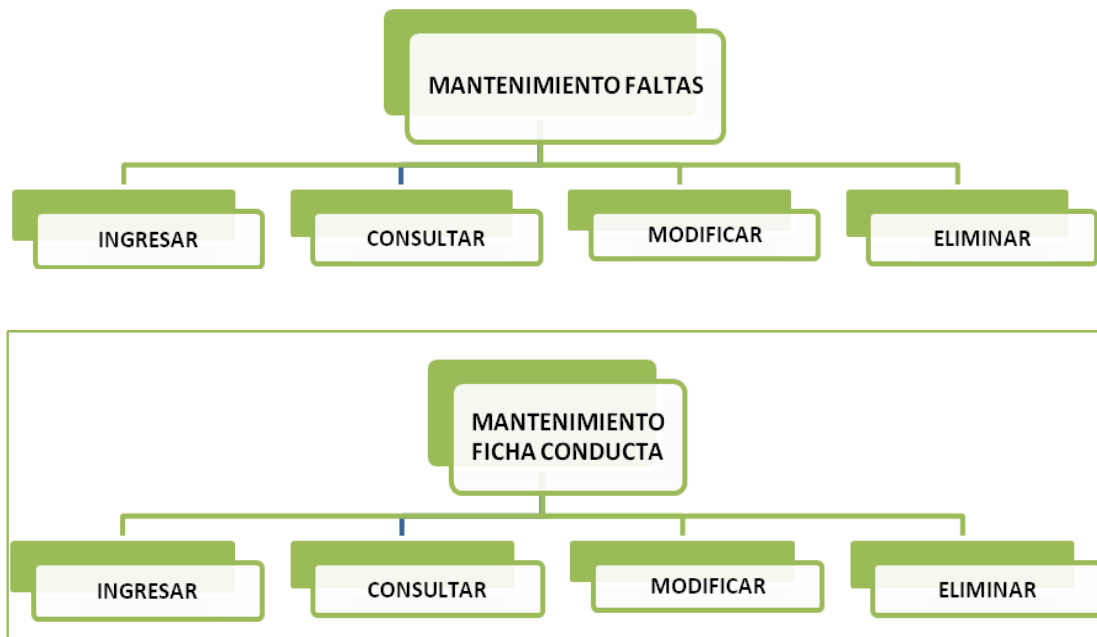
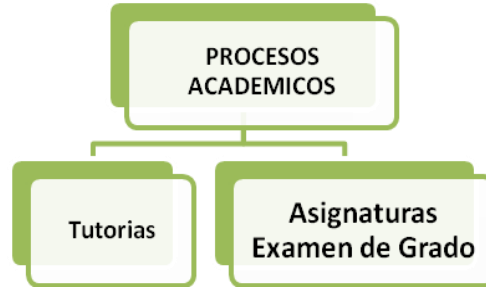


Tabla #34: Muestra opciones de mantenimiento faltas y ficha conducta.

Fuente: Los autores.

PROCESOS ACADEMICOS

Nivel 1._



Nivel 2._

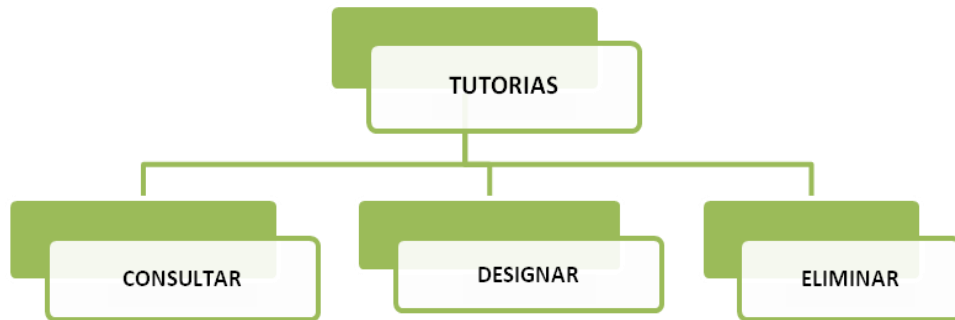


Tabla #35: Muestra el nivel para el proceso académico y tutoría.

Fuente: Los autores.



Tabla #36: Opciones para la opción de asignaturas examen de grado.

Fuente: Los autores.

3.12 Manual de Usuario

3.12.1 Pantalla de Inicio

Nuestro sistema posee una opción de seguridad en la cual el docente o secretaria deberá registrar su usuario y contraseña, una vez verificados estos datos se mostrara el menú correspondiente al modulo de calificaciones.

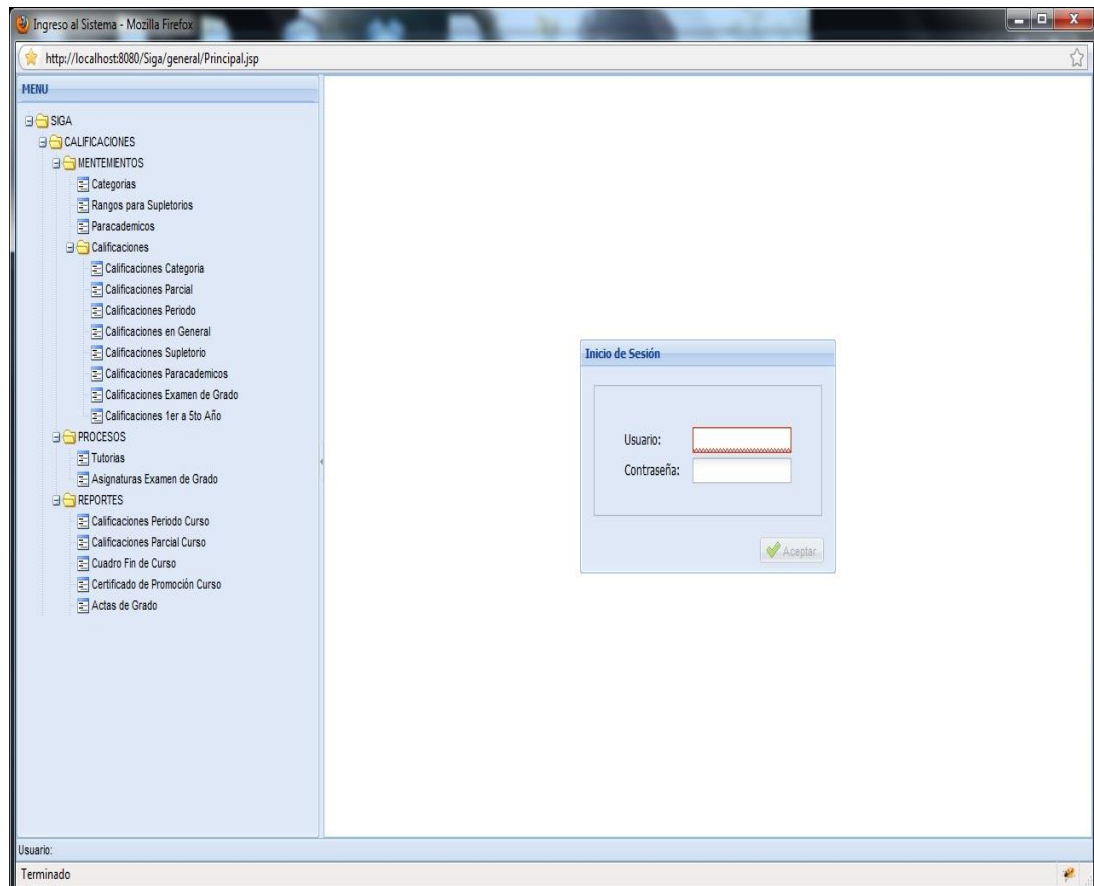


Figura #48: Muestra la pantalla inicial del sistema y autenticación.

Fuente: Los autores.

3.12.2 Mantenimientos

Los formularios de mantenimientos poseen un mismo esquema como podemos ver a continuación, debido a que todos realizan las mismas funciones.

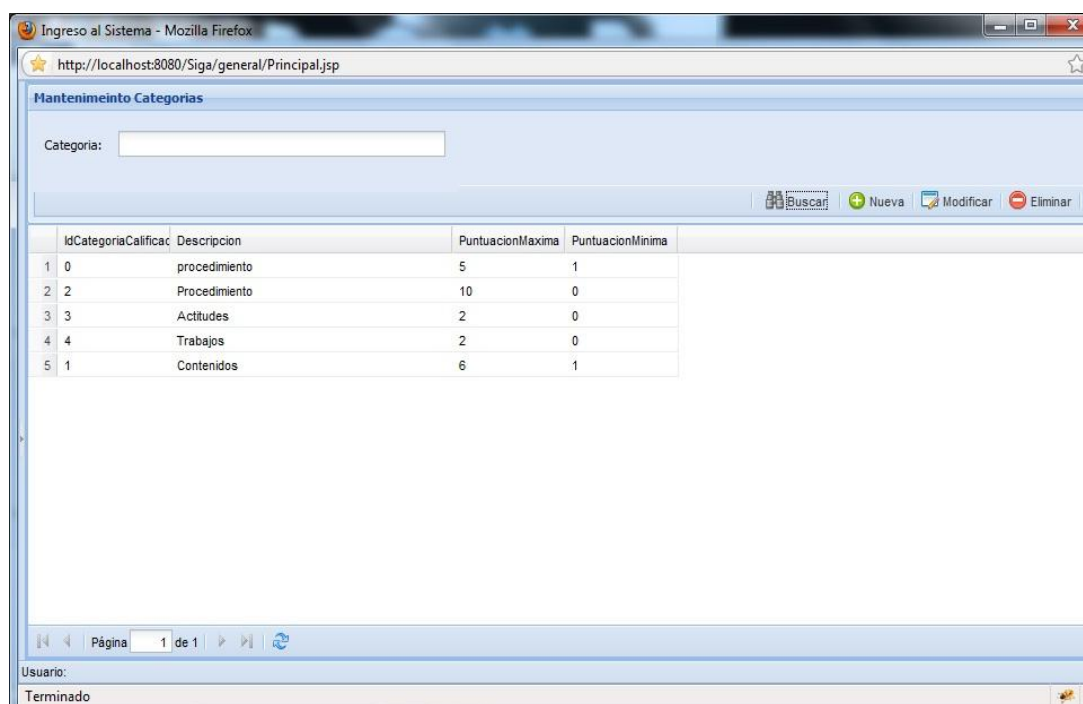


Figura #49: Muestra opciones de mantenimiento para todos los procesos.

Fuente: Los autores.

3.12.3 Ingresos

Los formularios de ingreso de categoría, paracadémico, rangos de supletorio también poseen el mismo esquema, pero las características a ingresar variaran según la opción seleccionada.



Figura #50: Muestra opciones para ingreso de una categoría de calificación.

Fuente: Los autores.

Para ingresar un nuevo registro el usuario debe seleccionar la opción nueva de la pantalla de mantenimiento, donde se activara una ventana flotante con los campos a ingresar.

3.12.4 Mantenimiento Calificaciones

Para el mantenimiento de calificaciones existen dos esquemas de calificación diferentes:

1. Dirigida al docente.
2. Dirigida a secretaria.

MantenimientoCalificacionesDocente

Para el registro de calificaciones por parte del docente, una vez registrado su usuario y contraseña se le mostrará una pantalla con el listado de todas las materias asignadas al mismo, así como su respectivo curso.

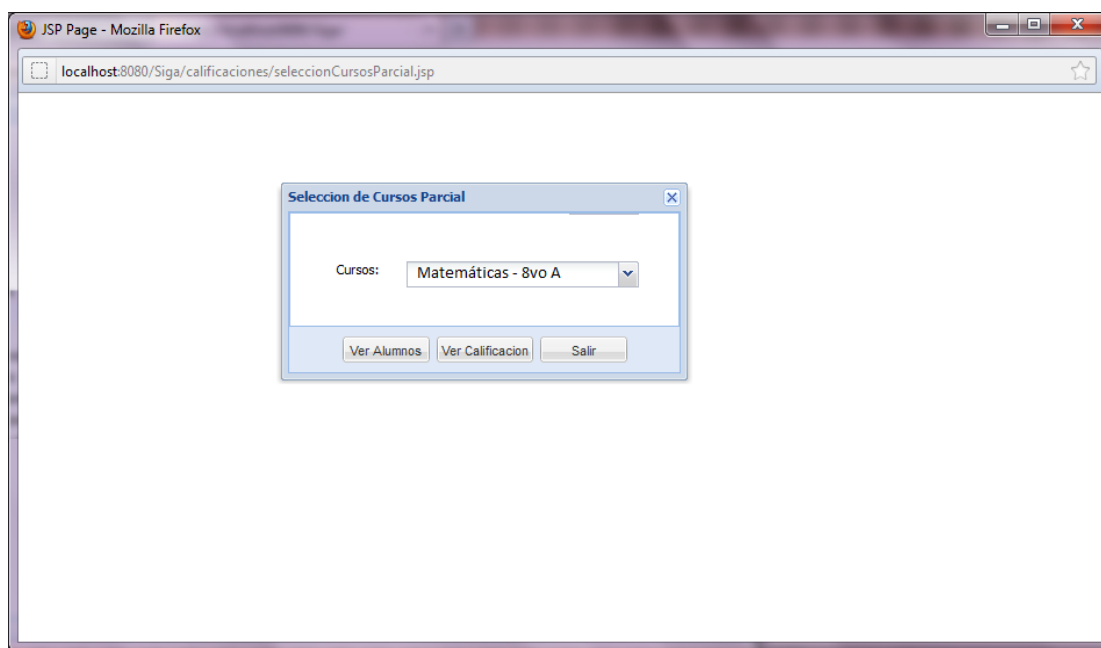
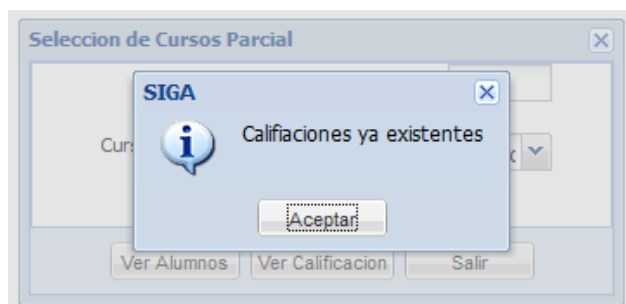


Figura #51: Muestra opciones de mantenimiento para calificación docente.

Fuente: Los autores.

El registro de las calificaciones por parte del docente se realizará por categoría, parcial o periodo, si las calificaciones se han registrado previamente según la opción seleccionada se le mostrará la siguiente alerta.



Si las calificaciones no han sido registradas anteriormente le aparecerá el listado de los alumnos a calificar, como se muestra a continuación.

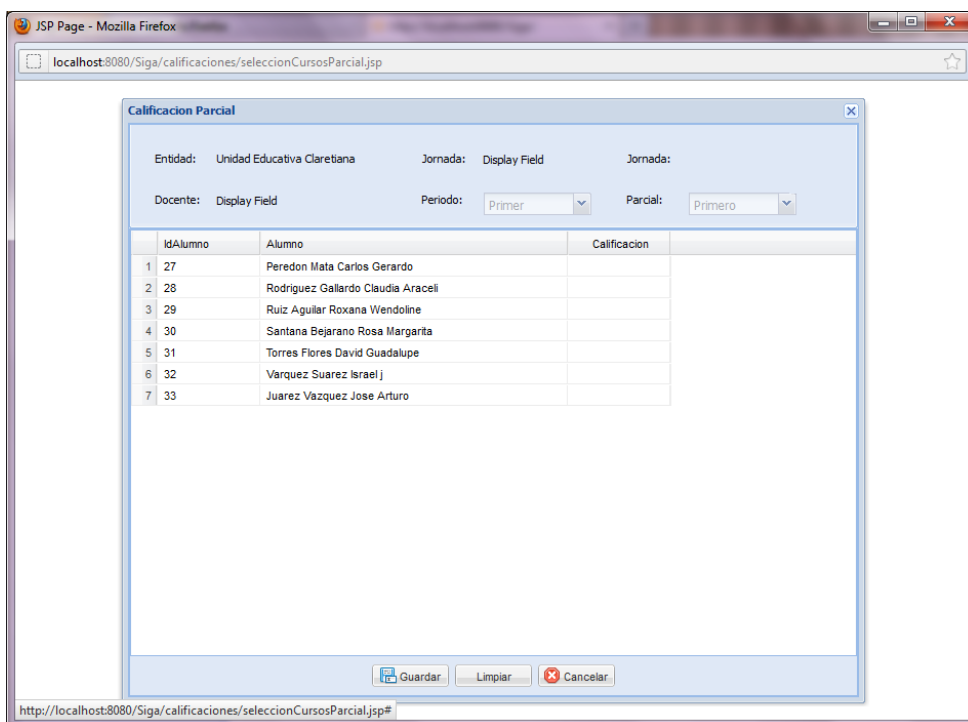


Figura #52: Muestra la lista de alumnos disponibles a calificar.

Fuente: Los autores.

Al finalizar el registro respectivo se mostrara la alerta respectiva.



Adicional el docente podrá realizar consultas de las calificaciones previamente ingresadas de una respectiva materia y curso.

2010-2011

Calificaciones Informatica

Curso: 10m0 A

Especialidad: Ciclo Basico

Alumnos	Primero Periodo					Segundo Periodo					Tercero Periodo					Total periodo
	Examen Parcial	Primero Parcial	Segundo Parcial	Tercero Parcial	Promedio	Examen Parcial	Primero Parcial	Segundo Parcial	Tercero Parcial	Promedio	Examen Parcial	Primero Parcial	Segundo Parcial	Tercero Parcial	Promedio	
Arguello Sanchez Alejandro	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Banda Estrada David	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Bueno Contreras Jose	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Camacho Cordero Juan	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Sonzalez Tokman Gil	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Loredo Mendez Maria	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Medina Herrera Francisco	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18

Figura #53: Imagen que muestra calificaciones correspondientes a una materia.

Fuente: Los autores.

Mantenimiento Calificaciones Secretaria

En la siguiente pantalla la secretaria tendrá la opción de consultar y modificar una calificación ya existente o registrar una nueva.

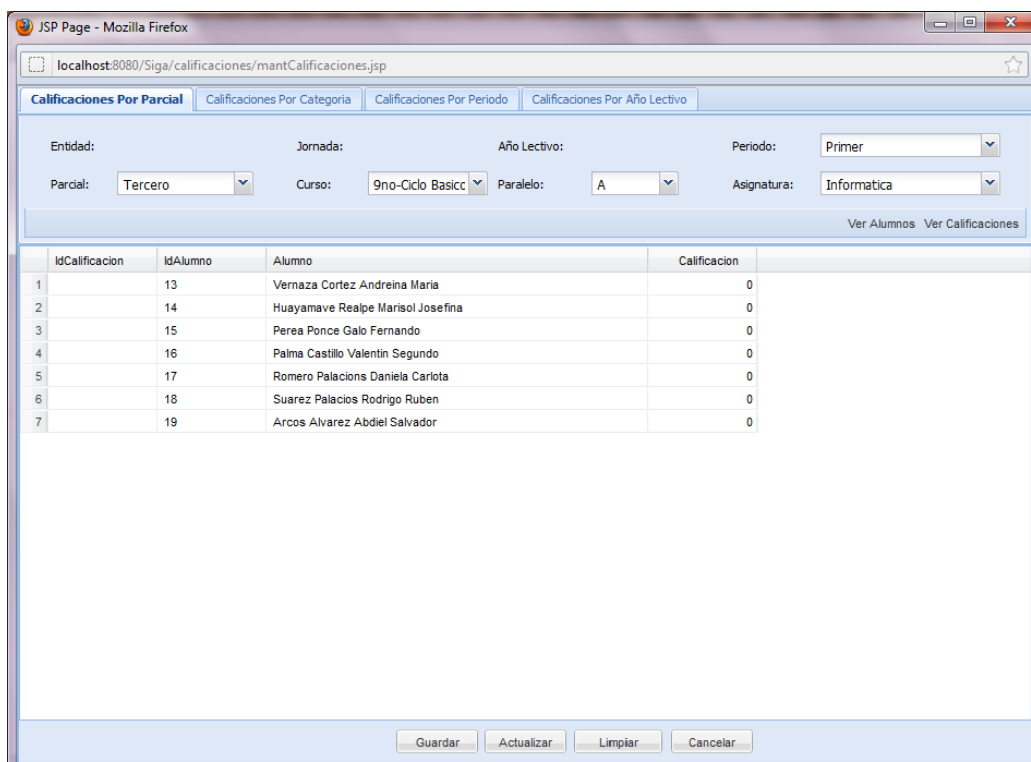


Figura #54: Imagen que muestra el mantenimiento de calificacion por secretaria.

Fuente: Los autores.

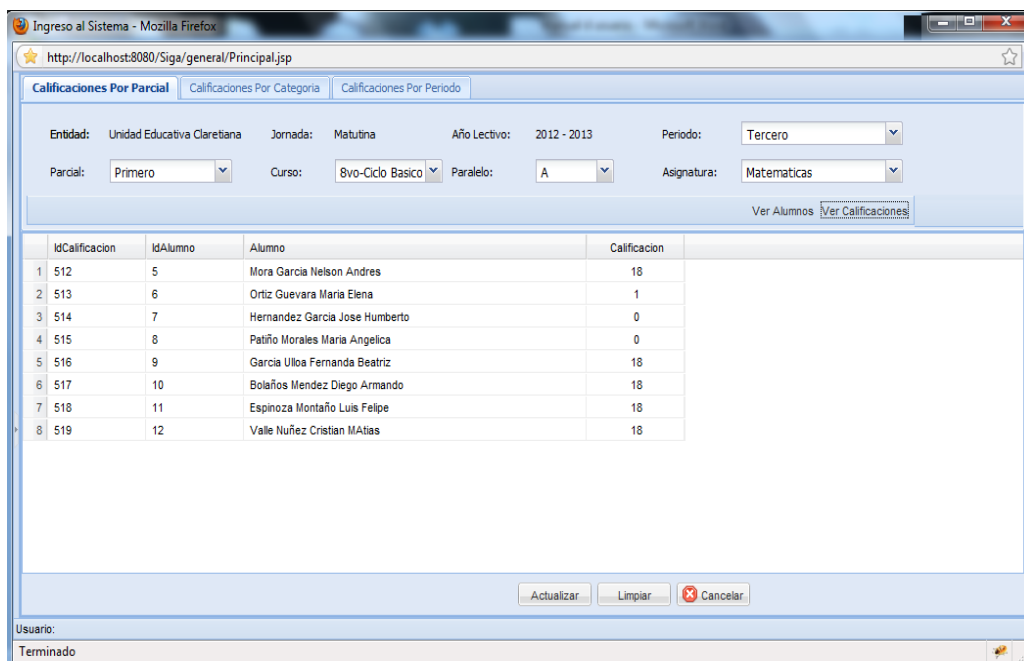


Figura #55: Imagen de las opciones disponible para secretaría para modificación o consulta.

Fuente: Los autores.

Para el respectivo ingreso o consulta de una determinada calificación, la secretaria deberá seleccionar el curso, el paralelo, la asignatura, el periodo y el parcial correspondiente. Una vez seleccionados los aspectos se le presentaran el listado de alumno correspondiente al curso seleccionado con su respectiva calificación (en el caso de consulta).

Una vez realizado el ingreso o la modificación respectiva el sistema retornara la alerta correspondiente.



Para obtener la información impresa la secretaria o la persona encargada tendrán varias opciones a su elección, entre ellas:

➤ Calificaciones por curso y por periodo o a su vez por parcial.

UNIDAD EDUCATIVA CLARETIANA
2010-2011

Calificaciones Tercer Periodo

Curso: 10m0 A Especialidad: Ciclo Basico

Asignaturas	Informatica	Sociales	Matematicas	Lenguaje	Inglés	Cultura Física	Ciencias Naturales	Contabilidad	Dibujo	Conducta
	Alumnos									
Arguello Sanchez Alejandro	18	17	18	17	17	20	20	20	20	20
Banda Estrada David	18	17	19	18	19	19	19	19	19	19
Bueno Contreras Jose	18	16	17	17	19	19	19	19	19	19
Camacho Cordero Juan	18	20	18	18	20	20	20	20	20	20
Gonzalez Tolman Gil	18	17	19	18	18	20	20	20	20	20
Loredo Mendez Maria	18	18	18	19	18	19	19	19	19	19
Medina Herrera Francisco	18	19	18	20	18	18	18	18	18	18

Figura #56: Muestra las calificaciones por curso y por periodo o parcial.

Fuente: Los autores.

➤ Libreta de Calificaciones

UNIDAD EDUCATIVA CLARETIANA
Jornada matutina
2010-2011

Libreta Calificaciones

Alumno: Arguello Sanchez Alejandro
Curso: 10m0 A Ciclo Basico Especialidad: Ciclo Basico

Asignaturas	Primero Periodo					Segundo Periodo					Tercero Periodo					Promedio periodo
	Examen Parcial	Primero Parcial	Segundo Parcial	Tercero Parcial	Promedio parcial	Examen Parcial	Primero Parcial	Segundo Parcial	Tercero Parcial	Promedio parcial	Examen Parcial	Primero Parcial	Segundo Parcial	Tercero Parcial	Promedio parcial	
Informatica	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Matematicas	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Inglés	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Lenguaje	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Cultura Física	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18
Ciencias Sociales	18	18	18	17	17	18	18	18	19	18	19	17	18	19	18	18

Figura #57: Libreta de calificaciones correspondiente a un alumno.

Fuente: Los autores.

3.13 Conclusiones

Los procesos automatizados correspondientes a la gestión de convivencia, régimen académico, evaluación y gestión de calificaciones y que forman parte de la solución integrada de gestión educativa aplicable a cualquier nivel de educación están analizadas, diseñadas, desarrolladas y probadas en base a requerimientos previos planteados por la institución educativa en donde se implementará la solución, respetando el estándar actual para que se acople a cualquier sistema de educación y faciliten el procesamiento de información.

El módulo de gestión de convivencia y régimen académico permite controlar la asistencia del alumnado, permitiendo registrar la asistencia, falta o atraso de un alumno así como la justificación de una falta, este proceso es controlado por la supervisora y muestra datos consistentes y facilita identificar a tiempo problemas que presenta el alumnado.

El módulo de Evaluación y Gestión Académica fue desarrollado para facilitar la gestión del docente y secretaría permitiendo tener un registro consistente de las notas del alumnado y permitiendo poder realizar correcciones desde el menú de mantenimientos, facilita el cálculo de promedios a nivel de ciclo que use la institución como trimestre, bimestre o quinquimestre así como conocer las nota mínima requerida para aprobación del alumnado que este supletorio.

Permite obtener reportes por alumno, materia, año lectivo, período con el fin de monitorear a un curso o a un estudiante en particular, ofrece seguimiento continuo para valorar al alumnado y en el caso que se detecte algún rendimiento fuera del rango permitido poder brindar apoyo oportuno.

3.14 Recomendaciones

Con el fin de poder realizar la implementación del ERP con transparencia y de forma eficiente se requiere contar con la infraestructura necesaria para que no limite el desempeño de la solución propuesta, esto se debe a que la información debe estar siempre disponible y contar con el soporte necesario para poder brindar solución a tiempo.

Es importante que la aplicación deba permanecer relativamente separada de la base de datos por lo que se recomienda dos servidores, una para la aplicación y otro para que resida la base de datos. Adicional se requiere validar el navegador que utilizarían las estaciones de trabajo para utilizar la aplicación, así como los parches disponibles para mejorar el funcionamiento de la solución.

En la parte de Recurso Humano se requiere capacitar al personal para poder utilizar todos los recursos de la aplicación con el fin de evitar la subutilización.

Se recomienda el control de los servidores que alojaran la aplicación y la base de datos con un antivirus robusto y la asignación de un responsable para monitorear, brindar mantenimiento y soporte a los mismos con el fin de evitar caídas en la aplicación.

3.15 Bibliografía

- Ajax
<http://es.wikipedia.org/wiki/AJAX>
- De la Torre Llorente Cesar (Microsoft), Zorrilla Castro Unai (Plain Concepts), Calvarro Nelson Javier (Microsoft), Miguel Ángel Ramos Barroso (Microsoft). “Guía de Arquitectura N-Capas orientada al Dominio con .NET 4.0” Editorial; Krasis Press. Edición I.
- Eguíluz Pérez Javier, Junio 2008 “Introducción a Ajax”.
<http://www.librosweb.es/ajax/index.html>
- Extjs:
<http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>
- Grupo Planeación Estratégica, 2009. “Método Inductivo y Deductivo”.
<http://planeacionestrategica.blogspot.es/1236115440>
- Mestanza Néstor, Burgos William. “sistema de evaluación del desempeño docente mediante aplicación web para la casa salesiana ‘Cristóbal Colon’”. Tesis para la obtención del título de Ingeniero de Sistemas. Universidad Politécnica Salesiana, Guayaquil – Ecuador. Diciembre 2010.
- Netbeans: Definición
[http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Fundamentos de Programacion Parte 2.pdf](http://biblioteca.uns.edu.pe/saladocentes/archivoz/publicacionez/Fundamentos_de_Programacion_Parte_2.pdf)
- Wikipedia “Ajax”
<http://es.wikipedia.org/wiki/AJAX>
- Wikipedia “Biografía San Antonio María Claret”
http://es.wikipedia.org/wiki/Antonio_Mar%C3%ADa_Claret

- Wikipedia “Extjs”
<http://blogs.antartec.com/desarrolloweb/2008/10/extjs-lo-bueno-lo-malo-y-lo-feo/>
- Wikipedia “Herramienta de soporte para el desarrollo de aplicaciones web”
http://es.wikipedia.org/wiki/Apache_Struts.
- Wikipedia “La Entrevista”
http://es.wikipedia.org/wiki/Entrevista_period%C3%ADstica
- Wikipedia “Modelo Relacional”
http://es.wikipedia.org/wiki/Modelo_relacional
- Wikipedia “Postgres”
<http://es.wikipedia.org/wiki/PostgreSQL>
- Wikipedia “Ventajas del IDE Netbeans”
<http://es.wikipedia.org/wiki/NetBeans>

ANEXOS

3.16.1 Encuesta para la estudio de factibilidad de un sistema integrado de gestión educativa.

ENCUESTA PROYECTO “SIGA”

Universidad Politécnica Salesiana

Tema: Implementación de un software integrado de gestión académica

1. ¿En qué departamentos cree usted que sería factible optimizar el uso de recursos dentro de la institución?

Contabilidad Rectorado RRHH
Secretaria Orientación otros: _____

2. ¿Cree usted necesario integrar todas las áreas que posee su institución para lograr una comunicación eficiente entre éstas?

Si ____ No ____

3. ¿Cree usted que es necesario un análisis del desempeño por departamento con el objetivo de detectar falencias en su institución? (Si su respuesta es NO, pase a la pregunta 5)

Si ____ No ____

4. Si su respuesta anterior fue afirmativa, señale los departamentos:

Contabilidad Rectorado RRHH Pedagógica
Secretaria Orientación Otros: _____

5. ¿Posee la institución algún tipo de software que facilite el desarrollo de sus actividades? (Si su respuesta es NO, pase a la pregunta 11)

Si ____ No ____

6. ¿En qué áreas utilizan dicho software?

Contabilidad Rectorado RRHH Pedagógica
Secretaria Orientación Otros: _____

7. ¿El software actual que posee la institución le ha servido como una herramienta de trabajo que facilite el desarrollo de sus actividades y que le permite disponer de información confiable?

Nunca ____ Algunas veces ____ Casi siempre ____ Siempre ____

8. ¿El acceso, navegación y consulta de los datos se realizan de manera rápida y sencilla, lo que permite disponer de información oportuna?

Nunca ____ Algunas veces ____ Casi siempre ____ Siempre ____

9. ¿La información que usted maneja de la institución se encuentra almacenada de tal manera que facilita su búsqueda e identificación inmediata?

Nunca ____ Algunas veces ____ Casi siempre ____ Siempre ____

10. ¿Cuáles serían las limitaciones del software actual?

11. ¿Le gustaría poder realizar su trabajo de forma automatizada?

Si ____ No ____

12. Desea contar con una herramienta que le permita:

- Automatizar los procesos manuales de las institución educativa,
- Optimizar los recursos existentes,
- Integrar todas las áreas de la institución
- Obtener informes requeridos de manera rápida.
- Garantizar información de forma confiable, precisa y oportuna.
- Facilitar el acceso a la información en tiempo real.

Si _____ No _____

**ENTREVISTAS REALIZADAS PARA OBTENER INFORMACIÓN
IMPORTANTE PARA LA REALIZACIÓN DEL PROYECTO**

Entrevista al Ing. Alvaro Rivera, Responsable de cómputo de la “Unidad Claretiana”

1. ¿Los procesos que realiza la institución en cuanto a calificación, registro de atrasos, faltas, supervisión de convivencia, se están cumpliendo en el tiempo mínimo requerido?

Si No No sabe

2. ¿Si se automatiza los procesos del punto anterior ¿cree Ud. que sería factible cumplir con el tiempo requerido?

Si No No sabe

3. Si se automatizan los procesos del primer punto, ¿cree Ud. que disminuiría el número de errores al procesar la información?

Si No No sabe

4. ¿Qué criterios toma en cuenta para la selección de la plataforma en que se desarrollara la aplicación web?

Licencia libre Se ejecuta en todo tipo de servidor
Ambos

5. ¿Cual motor o manejador de bases de datos prefiere manejar para el proyecto? (Puede escoger más de uno si así es el caso)

SQL Server PostgreSQL MySQL
Otro_____

3.16.2 Instalación del motor de base de datos postgres-ql en windows

Esta instalación se realizará bajo Windows, al momento de tratarse del diseño y pruebas desde servidor local (localhost).

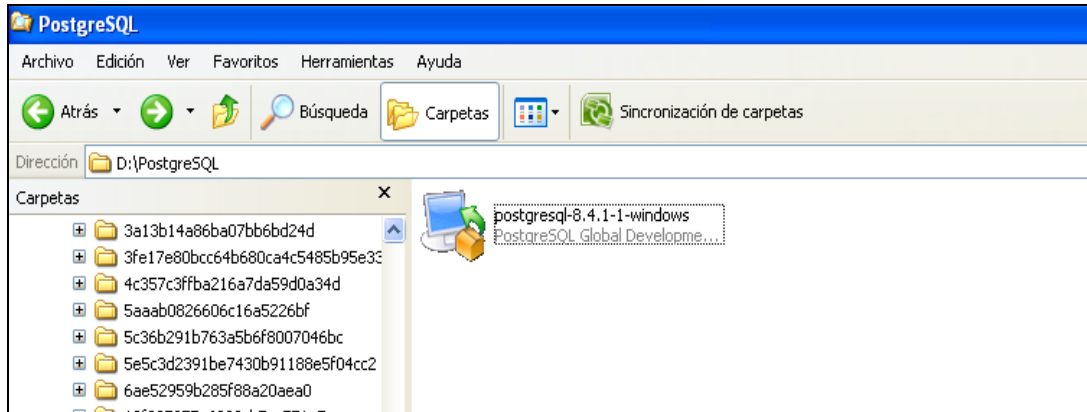


Figura #58: Instalador de postgresql.

Fuente: Los autores.

Luego se procede a ejecutar el archivo del instalador, postgresql-8.4.1-1-windows y obtenemos la siguiente pantalla de bienvenida.



Figura #58: Primer ventana que se muestra durante la instalación de PostgreSQL.

Fuente: Los autores.

Continuar, damos clic en el botón “Siguiente”.

En la siguiente pantalla, se muestra el directorio donde se instalará PostgreSQL, y para continuar damos click en ‘Next’.

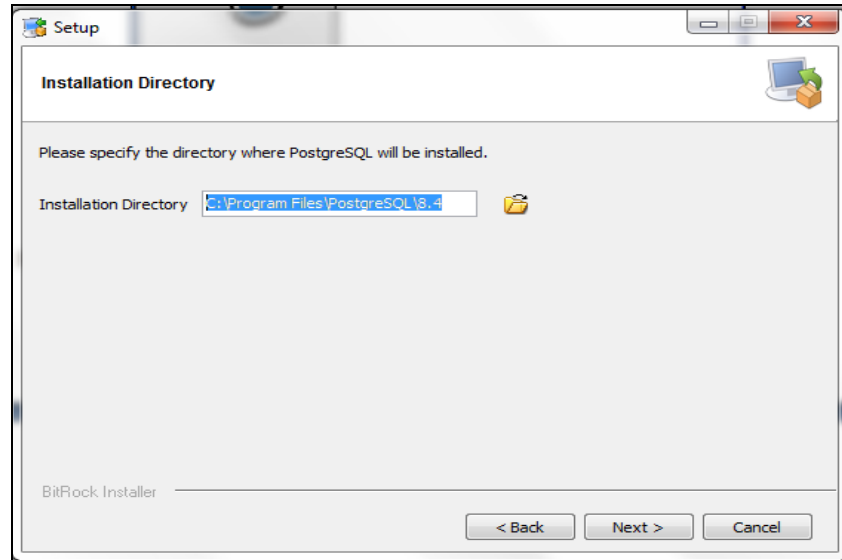


Figura #59: Imagen que muestra la ruta donde se instalará PostgreSQL.

Fuente: Los autores.

Luego tenemos la siguiente pantalla:

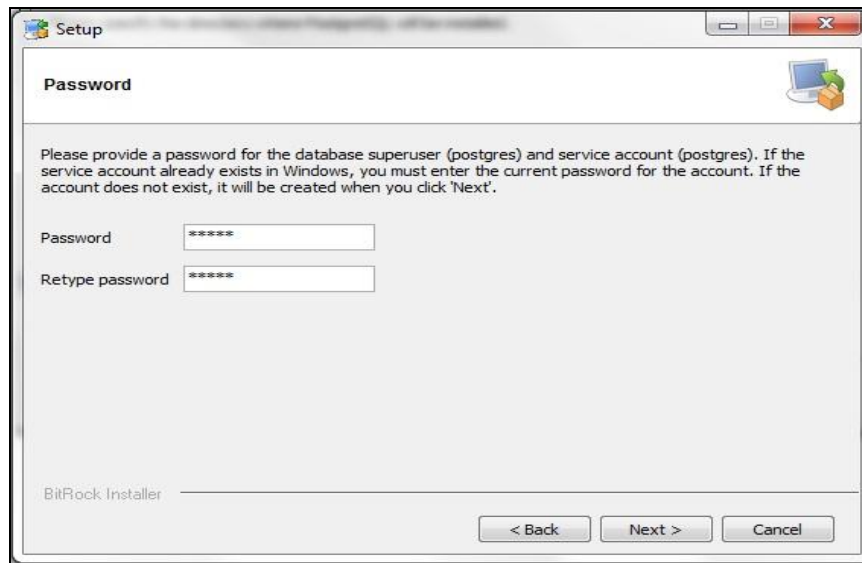


Figura #60: Nos solicita el usuario y contraseña por defecto.

Fuente: Los autores.

En la pantalla mostrada, nos indica que por defecto, el usuario principal se llamaría 'Postgres', al cual debemos darle una contraseña, por ejemplo 'admin'. Para continuar damos click en 'Next'.

Después se muestra una pantalla donde nos muestra el número de puerto con el que trabajará Postgres, el mismo que por predeterminación es 5432.

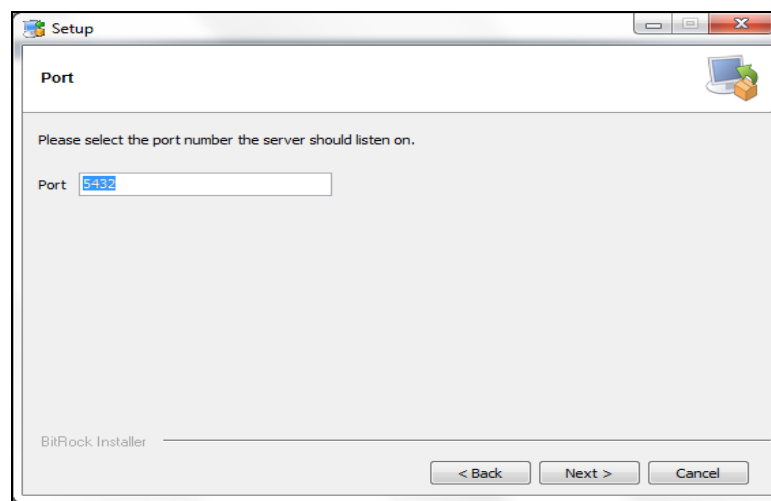


Figura #61: Nos muestra la ventana con el puerto predeterminado de instalación.

Fuente: Los autores.

En las demás pantallas asimismo, damos click en ‘Next’ hasta mostrar el progreso de la instalación.

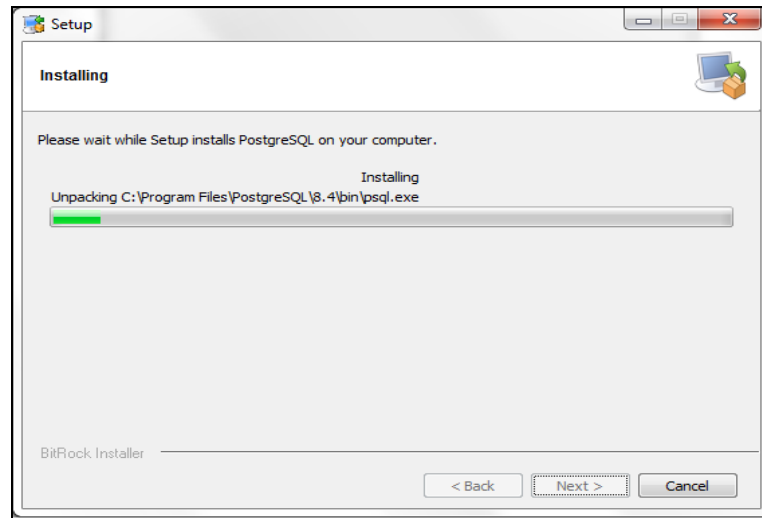


Figura #62: Muestra el progreso y la finalización de instalación de Postgres.

Fuente: Los autores.

Una vez terminada la instalación damos click en ‘Finish’ (ver figura anterior), para así concluir con la instalación de PostgreSQL.

3.16.3 Mapeo de las tablas de la base de datos

Antes de hacer el mapeo debemos hacer la conexión con nuestra base de datos de la siguiente manera:

Nos vamos a prestaciones ahí aparecerá Base de datos – Nueva conexión de base de datos, luego se abre la ventana de conexión en la cual en el combo Nombre de controlador debemos

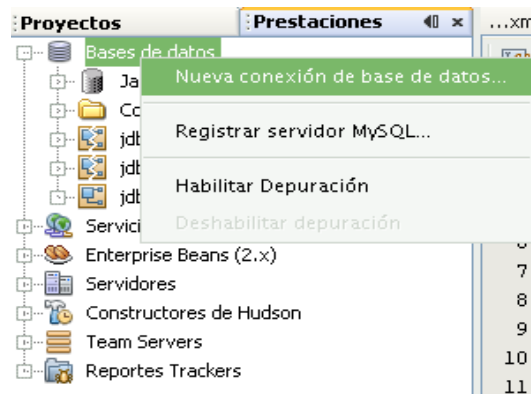
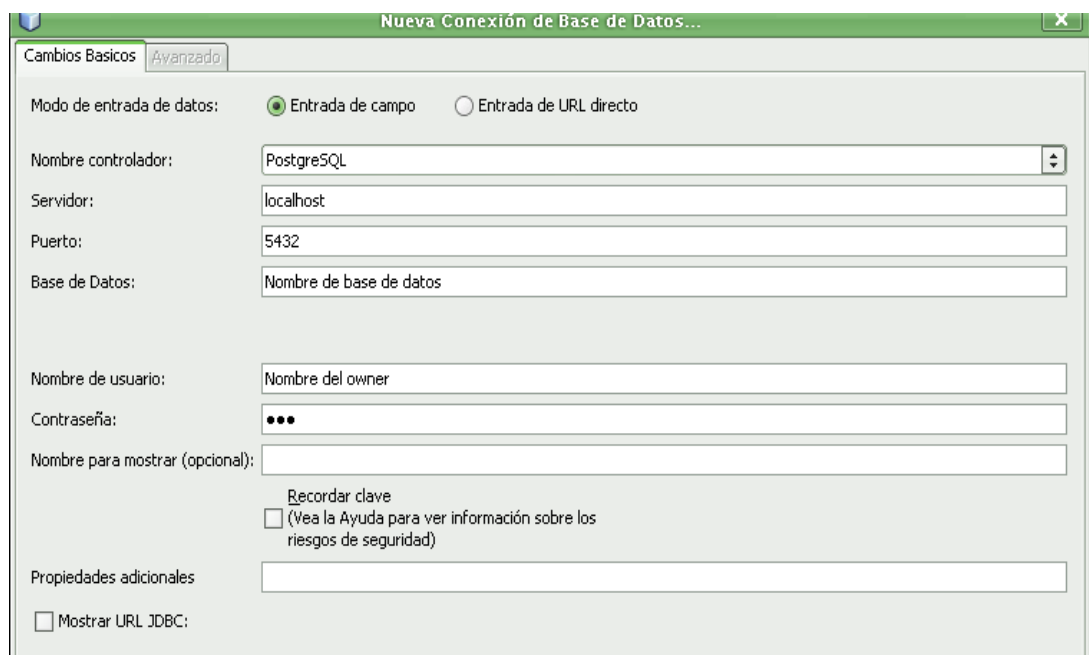


Figura #63: Muestra la opción de conexión de la base de datos con la aplicación.

Fuente: Los autores.

Elegir PostgreSQL, Servidor = localhost, Puerto = 5432, Base de datos = Nombre de la base de datos q crearon, Nombre de usuario = Nombre de su owner, Contraseña = la clave q pusieron al crear el owner. Luego click en ACEPTAR



Ahora veremos que ya existe en el árbol de base de datos nuestra conexión

Ahora si podremos hacer el mapeo de nuestras tablas porque ya existe la conexión con ella.

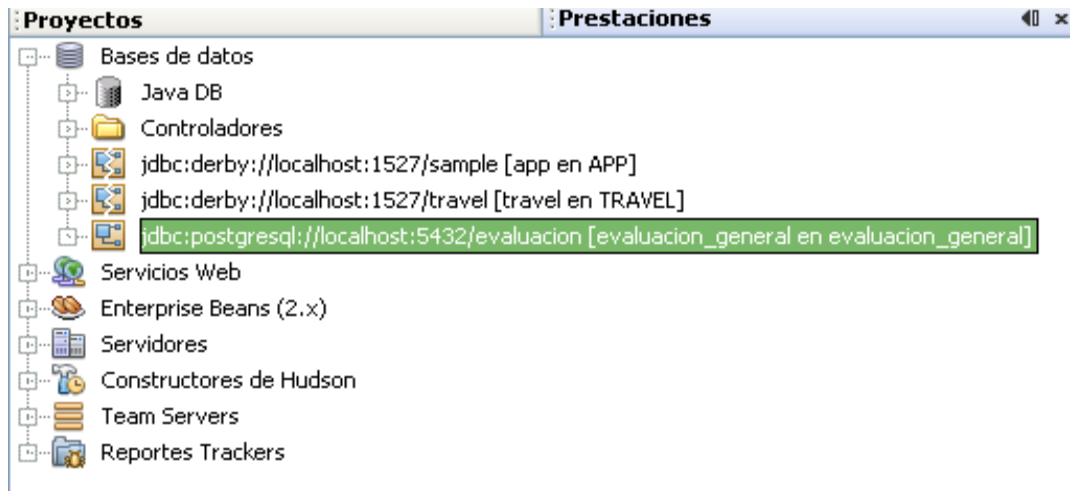


Figura #64: Imagen que muestra la conexión realizada con la base de datos.

Fuente: Los autores.

Todo lo a continuación explicado será manipulado en el proyecto relacional.

Primero creamos un paquete java llamado mapeo como muestra la figura.

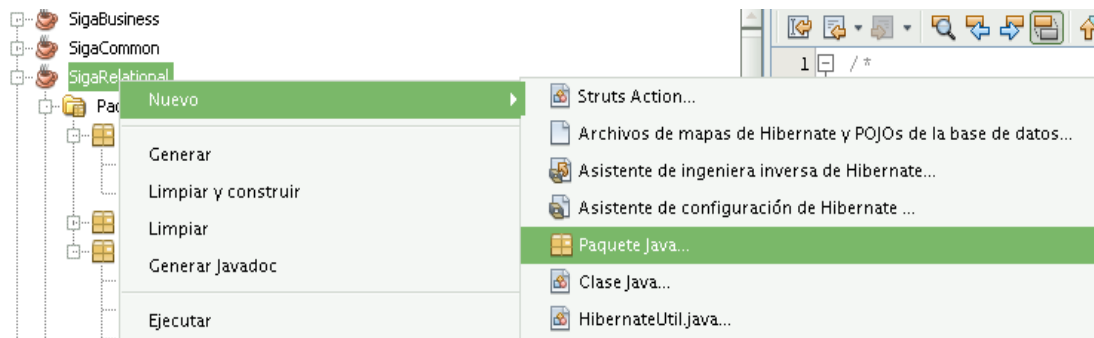


Figura #65: Imagen que muestra el primer paso para la creación de mapeo.

Fuente: Los autores.

Ponemos el nombre y click en terminar.

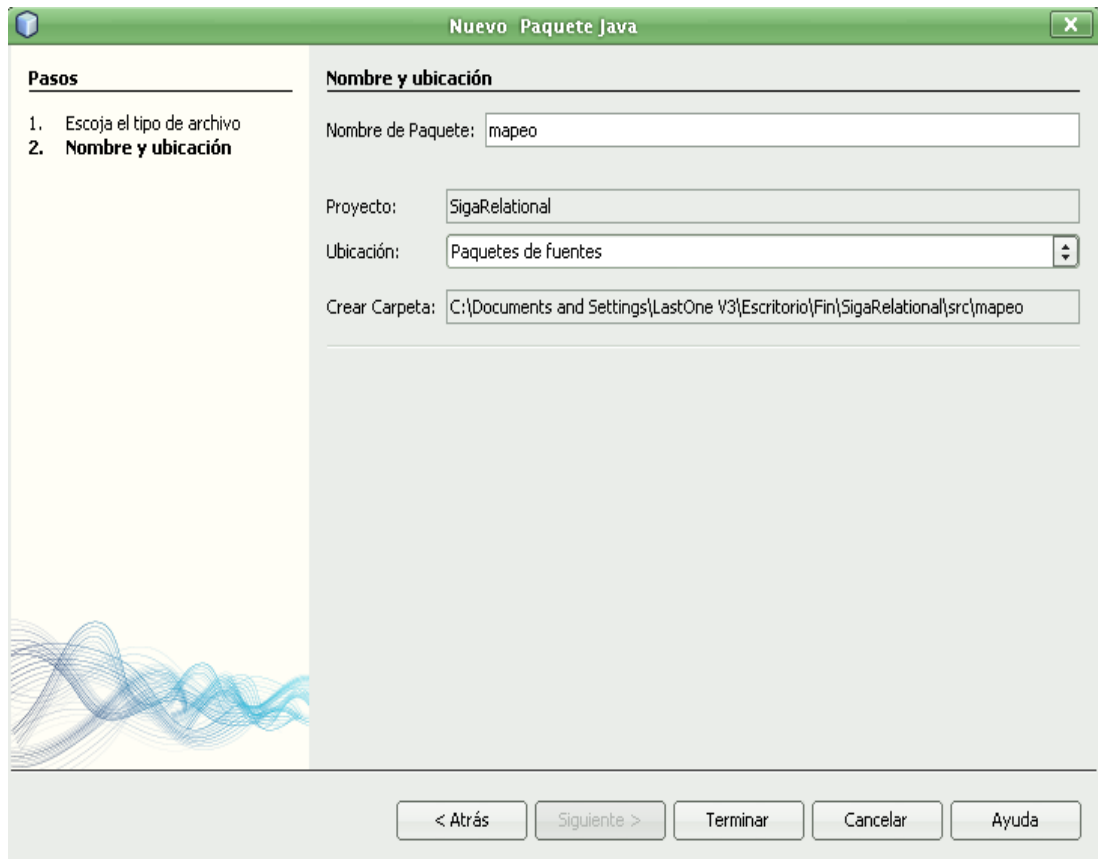
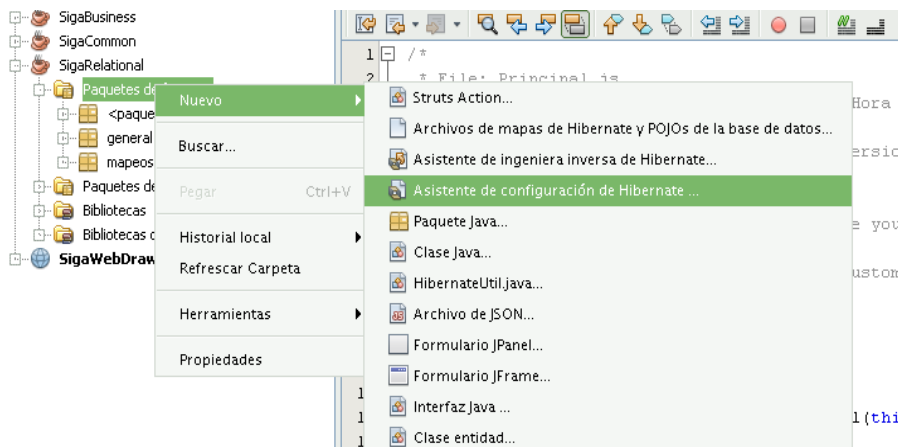


Figura #66: Creando mapeo parte 1.

Fuente: Los autores.

Ahora ya estamos listos para hacer nuestro mapeo...

Click derecho en paquetes fuentes – nuevo – Asistente de configuración de Hibernate; por default dejamos el mismo nombre y damos siguiente, tal como muestran las figuras.



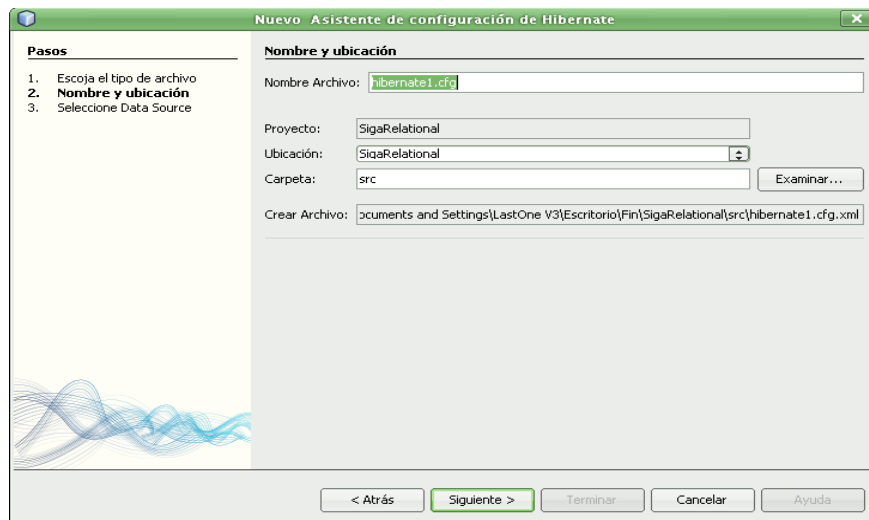


Figura #67: Creando mapeo parte 2.

Fuente: Los autores.

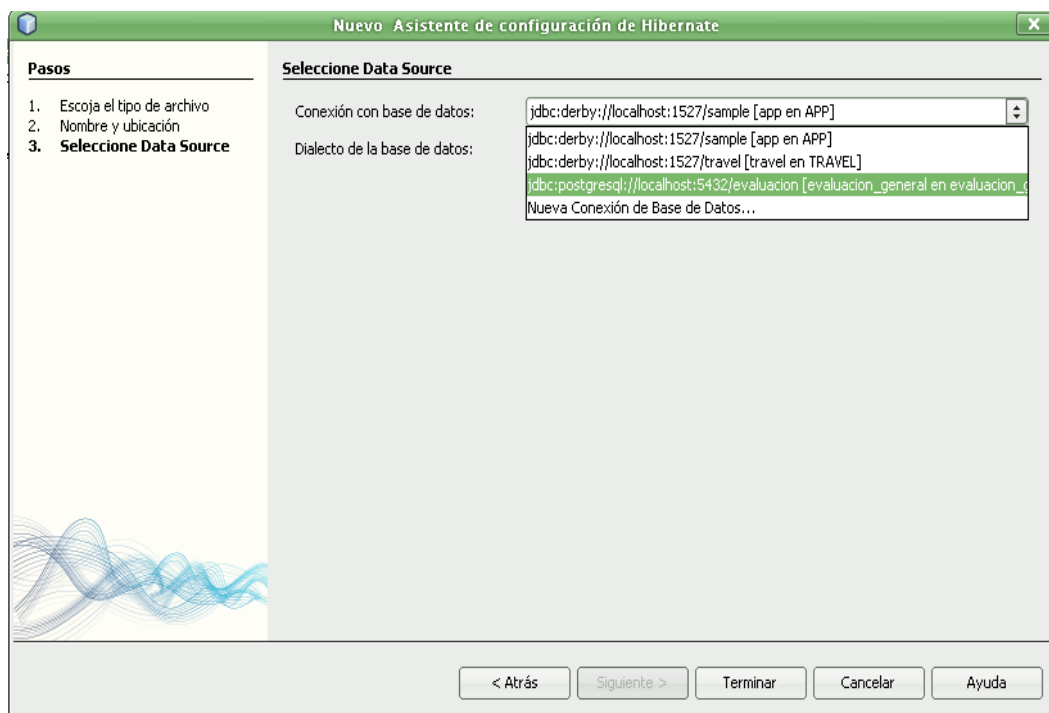


Figura #68: Creando mapeo parte 3.

Fuente: Los autores.

Luego nos aparecerá otra ventana en la cual debemos seleccionar la conexión que hicimos en pasos anteriores en el combo conexión con base de datos.

Ponemos terminar y se debe crear en un <paquete predeterminado> un archivo llamado hibernate.cfg.xml.

1. Click derecho en <paquetes predeterminado> – nuevo – Asistente de Ingeniería inversa de Hibernate; por default dejamos el mismo nombre y damos siguiente, tal como el paso anterior.

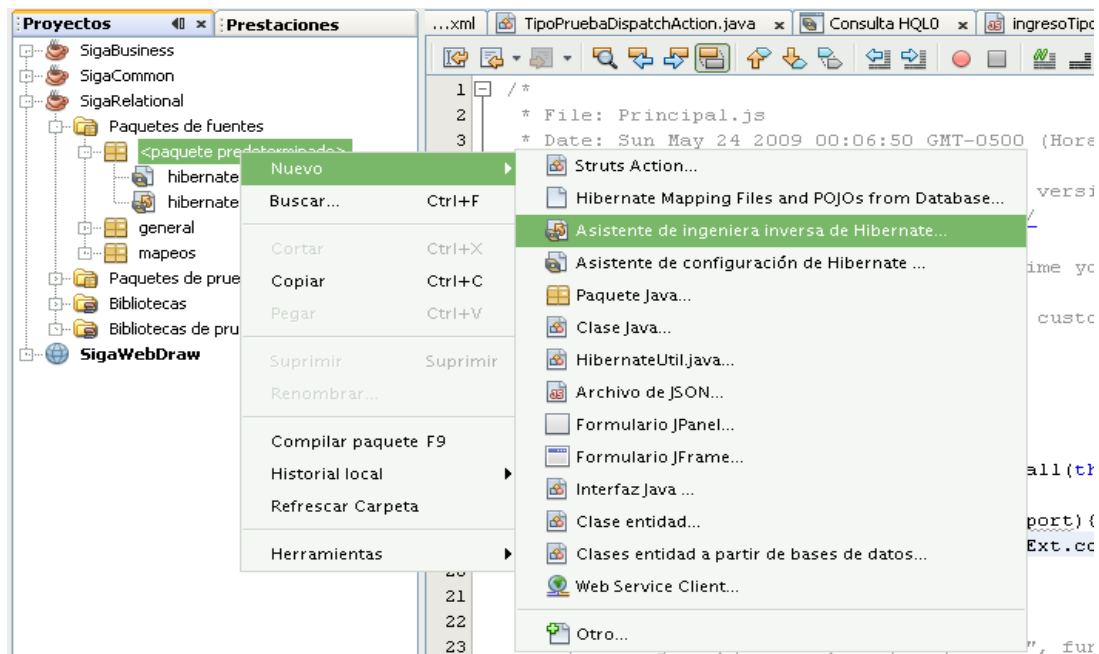


Figura #69: Creando mapeo parte 4.

Fuente: Los autores.

2. Luego se nos cargarán todas las tablas y ponemos Agregar todo>> y por ultimo click en el botón terminar.

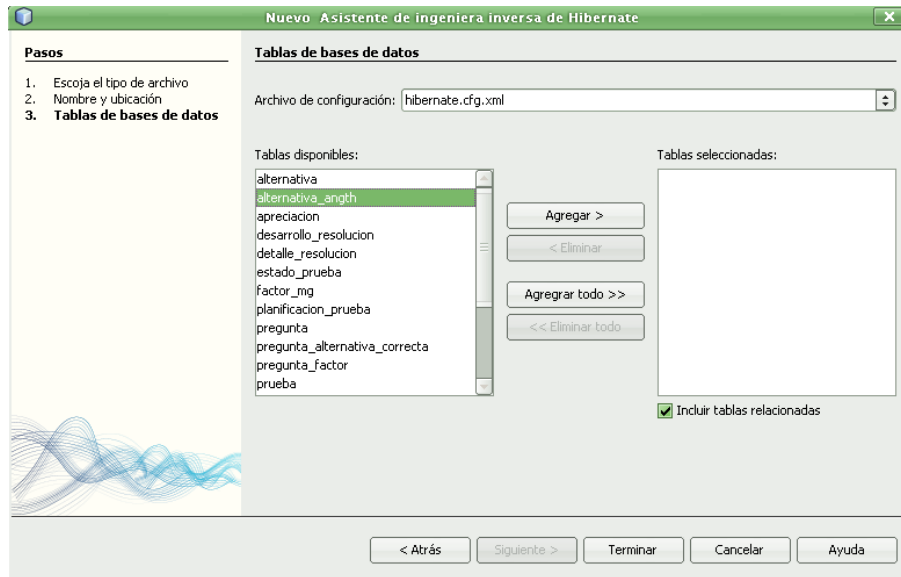


Figura #70: Creando mapeo parte 5.

Fuente: Los autores.

Una vez que hacemos eso ya tenemos los 2 archivos:

- ✓ hibernate.cfg.xml
- ✓ hibernate.reveng.xml

3. Por último le damos click derecho en nuestro paquete de mapeos – nuevo – Archivos de mapas de Hibernate y POJOs de la base de datos.

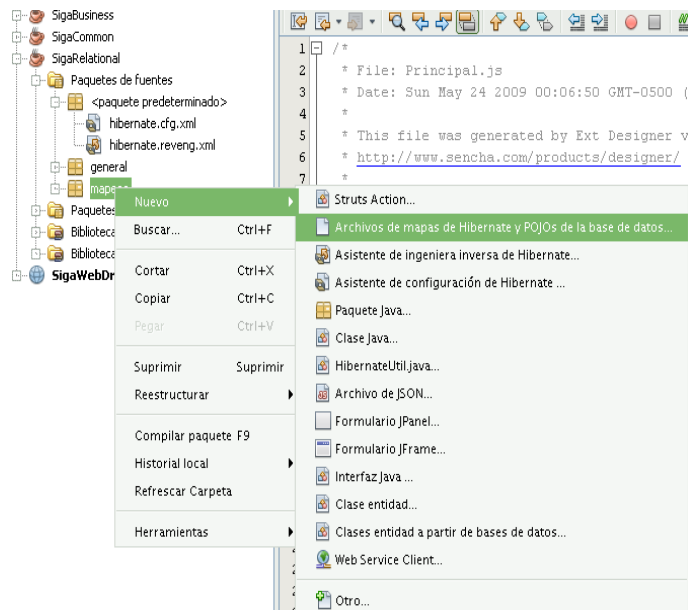
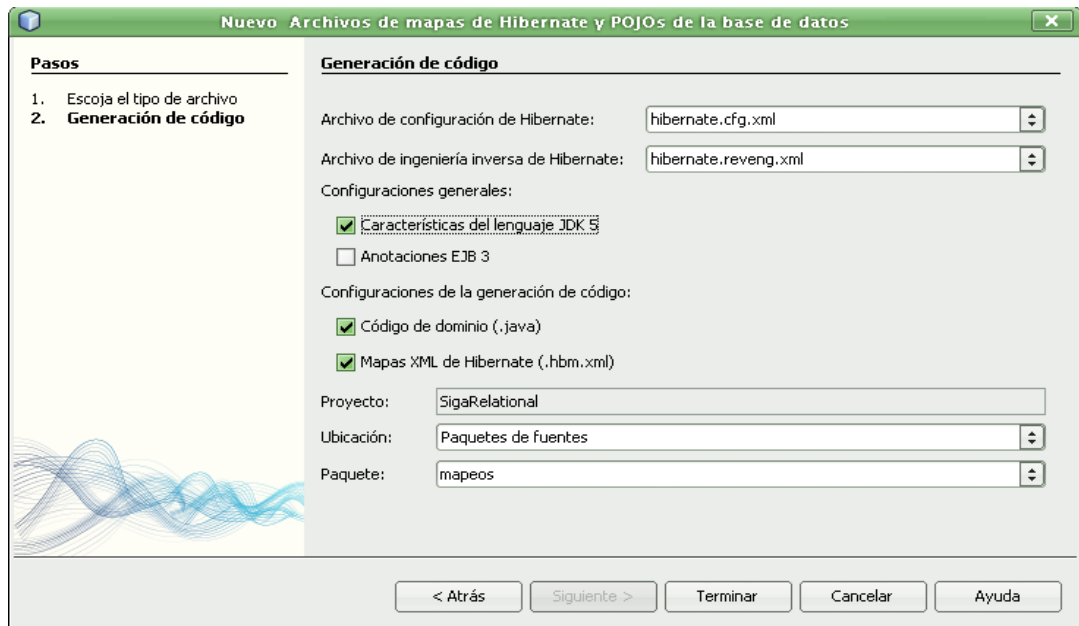


Figura # 71: Creando mapeo parte 6.

Fuente: Los autores.

Una vez seleccionado nos aparece la ventana y seleccionamos el check de características del lenguaje JDK 5 y click en el botón terminar.



Una vez dado en terminar revisamos en nuestro paquete de mapeos y deberán aparecer todos los nombres de sus tablas, cada tabla trae 2 archivos, ejemplo:

- ✓ Tabla.java
- ✓ Tabla.hbm.xml



Figura #72: Creando mapeo parte 7.

Fuente: Los autores.