



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Facultad de Ingeniería en Electricidad y Computación**

“SISTEMA PARA CONTROL DE CALIFICACIONES  
ESCOLARES”

**INFORME DE PROYECTO INTEGRADOR**

Previo a la obtención del Título de:

**INGENIERA EN CIENCIAS COMPUTACIONALES**  
**ORIENTACIÓN SISTEMAS MULTIMEDIA**

LILIANA JACQUELINE RAMOS MENÉNDEZ

GUAYAQUIL – ECUADOR

AÑO: 2016

## AGRADECIMIENTOS

Mis más sinceros agradecimientos a la PhD. Cristina Abad por su asesoramiento durante el desarrollo de este proyecto

Al Ing. Fernando Brito y la Unidad Educativa “Montepiedra” por permitirme implementar este proyecto y darme retroalimentación cuando fue necesario.

A mi padre.

A mi hermana.

A Harry.

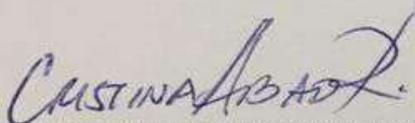
## DEDICATORIA

El presente proyecto lo dedico a mi padre Luis por su apoyo y respaldo incondicional durante mis estudios. Por darme consejos a tiempo e inculcarme valores que llevaré conmigo durante toda mi vida.

A mi hermana Gabriela, por siempre estar cuando la he necesitado y darme un excelente ejemplo a seguir en el día a día.

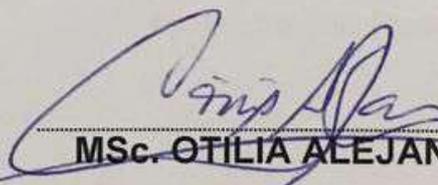
A Harry, por haber estado a mi lado en los momentos de alegrías y fracasos. Por apoyarme, ayudarme y sostenerme durante toda mi vida personal y académica.

## TRIBUNAL DE EVALUACIÓN



PhD. CRISTINA ABAD R.

PROFESOR EVALUADOR

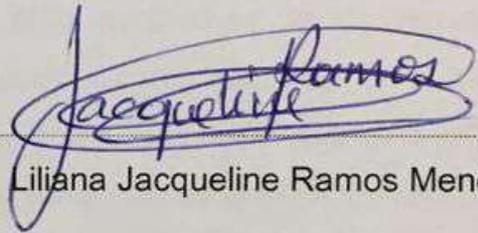


MSc. OTILIA ALEJANDRO M.

PROFESOR EVALUADOR

## DECLARACIÓN EXPRESA

"La responsabilidad y la autoría del contenido de este Trabajo de Titulación, me corresponde exclusivamente; y doy mi consentimiento para que la ESPOL realice la comunicación pública de la obra por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual"

A handwritten signature in blue ink, written over a horizontal dotted line. The signature is stylized and appears to read 'Liliana Jacqueline Ramos Menéndez'.

Liliana Jacqueline Ramos Menéndez

## RESUMEN

El presente documento describe los problemas encontrados en el sistema académico de la unidad educativa “Montepiedra” y se propone la implementación de un sistema nuevo para solucionarlos. Dicho sistema, debe cumplir con las especificaciones del ministerio de educación y la institución para generar libretas. Por esta razón, el sistema está comprendido por módulos como *año lectivo*, *estudiante*, *conducta*, *notas*, *personal* e *informes*.

El sistema propuesto denominado “Sistema para control de calificaciones escolares” tiene como finalidad construir una solución para problemas de modernización, redondeo, seguridad, mantenimiento, procesamiento e implementación de un sistema de escritorio con tecnologías deprecadas como Visual Basic V6, MySql 4 y Crystal Reports 8.5. El sistema es una aplicación web y para esto se usaron las tecnologías Java EE 7, JavaServer Faces, MySql 5.7.3, Spring Security y JasperReports.

Debido a que la base de datos del sistema académico actual está completamente desnormalizada, se hizo un rediseño y se contemplaron las nuevas necesidades que indica el ministerio de educación para generar libretas parciales y quimestrales, el cual es el objetivo final de este proyecto.

Para poder comprobar la generación de estas libretas, se hizo una migración de datos por medio de un ETL creado con Data Integration de Pentaho. El ETL tuvo un diseño exhaustivo y fue exitoso en la carga de datos como *año lectivo*, *cursos*, *paralelos* y *materias*. Cuando se intentó cargar las notas parciales, se notaron las inconsistencias debido a la desnormalización de la base origen. Se plantea una posible solución en la sección de conclusiones y recomendaciones.

## ÍNDICE GENERAL

AGRADECIMIENTOS.....	ii
DEDICATORIA .....	iii
TRIBUNAL DE EVALUACIÓN .....	iv
DECLARACIÓN EXPRESA.....	v
RESUMEN.....	vi
ÍNDICE GENERAL.....	vii
CAPÍTULO 1 .....	1
1. DESCRIPCIÓN GENERAL DEL SISTEMA ACADÉMICO ACTUAL .....	1
<b>1.1 Características del sistema actual y el sistema educativo del país.....</b>	<b>1</b>
<b>1.2 Características del sistema actual con respecto a su implementación .....</b>	<b>2</b>
<b>1.2.1 Tecnologías.....</b>	<b>2</b>
<b>1.2.2 Características de implementación .....</b>	<b>2</b>
<b>1.3 Descripción de los problemas del sistema actual.....</b>	<b>3</b>
<b>1.3.1 Redondeo.....</b>	<b>3</b>
<b>1.3.2 Seguridad.....</b>	<b>3</b>
<b>1.3.3 Procesamiento e implementación.....</b>	<b>4</b>
<b>1.3.4 Mantenimiento .....</b>	<b>4</b>
<b>1.3.5 Modernización .....</b>	<b>5</b>
CAPÍTULO 2.....	6
2. IMPLEMENTACIÓN DE LA SOLUCIÓN PROPUESTA .....	6
<b>2.1 Metodología SCRUM .....</b>	<b>6</b>
<b>2.1.1 <i>Sprint 0</i> .....</b>	<b>7</b>
<b>2.1.2 <i>Sprint 1, 2 y 3</i> .....</b>	<b>10</b>
<b>2.2 Tecnologías.....</b>	<b>11</b>
<b>2.3 Arquitectura y patrón de diseño <i>facade</i>.....</b>	<b>11</b>
<b>2.4 Módulos.....</b>	<b>12</b>

<b>2.5 Seguridad</b> .....	13
<b>2.6 Reportes</b> .....	15
CAPÍTULO 3 .....	16
3. DETALLES DE LA SOLUCIÓN Y MIGRACIÓN DE DATOS .....	16
<b>3.1 Redondeo</b> .....	16
<b>3.2 Seguridad</b> .....	16
<b>3.3 Procesamiento e implementación</b> .....	17
<b>3.4 Mantenimiento</b> .....	17
<b>3.5 Migración de datos</b> .....	17
CONCLUSIONES Y RECOMENDACIONES .....	19
BIBLIOGRAFÍA .....	21

## **CAPÍTULO 1**

### **1. DESCRIPCIÓN GENERAL DEL SISTEMA ACADÉMICO ACTUAL**

La Unidad Educativa “Montepiedra” cuenta con un sistema académico de escritorio el cual tiene problemas críticos como: el redondeo en el cálculo de promedios, la seguridad, y la consistencia de los datos. Además de esto, se han visto en la necesidad de actualizarse y estar a la par con otras instituciones educativas que ofrecen servicios como el envío de tareas por medio de sus sistemas en la web.

Esta problemática da pie a que se considere una implementación desde cero, empezando por su sistema académico, y solucionando primero los problemas críticos descritos anteriormente.

En este capítulo se describe al sistema académico que existe en la actualidad, con sus características tecnológicas y de implementación. A su vez se podrá leer más a fondo sobre los problemas que tiene.

#### **1.1 Características del sistema actual y el sistema educativo del país**

El sistema académico que posee la unidad educativa refleja el comportamiento que existe en el sistema educativo del país en la actualidad. El año escolar está compuesto por dos quimestres con tres parciales cada uno, teniendo un promedio general para cada ocasión en las libretas correspondientes. Debido a que el sistema académico es de una unidad educativa, se debe generar libretas para cada estudiante en todos los cursos del ciclo básico (primer año básico a décimo año básico) y de bachillerato (primero de bachillerato a tercero de bachillerato)[2][3]. Cada uno con sus propias materias definidas por el ministerio de educación y la institución, creando promedios que solo competen al ministerio de educación llamado “promedio mec”.

Una vez finalizado el año escolar, se generan actas por curso y certificados de promoción de los estudiantes. En cada uno de estos documentos, se reflejan los

promedios de las materias y el promedio general mec que fue obtenido por el estudiante durante todo el año escolar.

## **1.2 Características del sistema actual con respecto a su implementación**

### **1.2.1 Tecnologías**

El sistema académico que la unidad educativa posee actualmente, está implementado en Visual Basic 6.0 [4], cuenta con una versión deprecada de MySql [5] y usa Crystal Reports [5] versión 8.5.

### **1.2.2 Características de implementación**

El sistema actual tiene más de 10 años en funcionamiento y cuenta con varios parches que se han implementado debido a los cambios en el sistema educativo del país. Además, genera aproximadamente 600 reportes, de los cuales ya no se usan más de la mitad.

La principal característica del sistema académico actual es que es de escritorio. Se encuentra alojado en el servidor que tiene la unidad educativa y mediante carpetas compartidas se puede acceder a él. A su vez, el sistema académico está conectado con dos sistemas adicionales: el financiero y el de control de profesores. Esta conexión solo existe a nivel de base de datos.

La información que refleja el sistema solamente puede ser accedida por medio de usuario y contraseña, la cual no se encuentra cifrada con ningún algoritmo. Esta información tiene reglas asociadas a los roles que se manejan, los cuales son: *administrador*, *profesor*, *profesor escuela*, *secretario*, *subdirector*, *rector* e *inspector*. Cada rol tiene un identificador y para administrar los permisos, se debe ingresar el identificador, mediante código, cuando se realiza mantenimiento en el sistema.

## 1.3 Descripción de los problemas del sistema actual

### 1.3.1 Redondeo

El sistema académico actual tiene serios problemas de redondeo cuando se calculan los promedios finales de los estudiantes. Este problema se debe básicamente al lenguaje utilizado. Visual Basic usa en su función *Round()* el redondeo del banquero [1], esto quiere decir que cuando se obtiene un número .5, se lo redondeará al valor par más cercano. Ejemplo: si se tiene 7.5 y 8.5, ambos se redondearan a 8. Esto en la práctica, no es lo que se necesita, puesto que lo que se busca es generar promedios con redondeo aritmético, en el cual, si se tiene 7.5 se redondea a 8 y 8.5 a 9.

El redondeo aritmético en Visual Basic, no está implementado.

### 1.3.2 Seguridad

Existen varios problemas de seguridad en el sistema actual como se describen a continuación:

- Carpeta compartida: el ejecutable del sistema y los reportes se encuentran en una carpeta compartida en el servidor. Esto puede generar que los usuarios cambien los reportes si así lo desean y manipulen la información existente en la base datos. Así mismo, pueden aplicar ingeniería inversa a la aplicación.
- Conexión a la base de datos: Cada computador de la institución tiene creada la conexión a la base de datos por medio de la estructura de datos DSN (Data Source Name) [7]. Este problema va de la mano con la carpeta compartida, puesto que al tener acceso a la base de datos y a los reportes, se puede observar la información tal y como está escrita. Un ejemplo claro es que se visualizan los usuarios con sus respectivas contraseñas y se puede cambiar la información de estudiantes, notas, etc.

- No existe registro de eventos: Se ha conocido que han habido casos en que las notas fueron modificadas antes de generar las libretas de los estudiantes. Este tipo de situación no se puede verificar puesto que no existe un historial de los cambios de las notas en el sistema. Las notas son lo más crítico para este tipo de aplicación.

### **1.3.3 Procesamiento e implementación**

Las libretas que genera el sistema académico actual realiza mucho procesamiento cada vez que se necesita imprimirlas, esto es debido a que el cálculo de los promedios se realiza en cada iteración. Esto va de la mano con el problema de redondeo, puesto que el promedio no se guarda en ningún lado de la base de datos, creando nuevos promedios dependiendo de donde se lo calcule, en la herramienta de reportes o en la aplicación como tal.

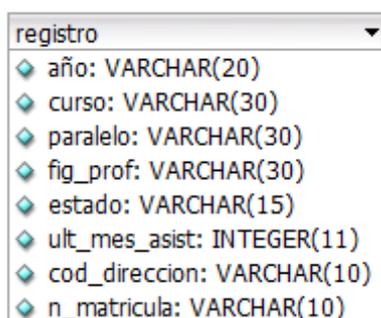
La implementación para el cálculo del promedio también crea conflictos, porque no se debe considerar las materias extracurriculares en el promedio mec. Esto no se refleja en el diseño de la base de datos, y está implementado de tal manera que se debe escribir manualmente, en el código fuente, el nombre de la materia. Aquí se crean conflictos cada vez que se cambia de nombre a la materia o se crea nuevas materias de este tipo.

### **1.3.4 Mantenimiento**

Debido a que el sistema ha estado en funcionamiento por más de 10 años, se hace difícil dar mantenimiento, especialmente cuando se quiere cambiar alguna característica de un año lectivo a otro. Por ejemplo: para el año lectivo 2016 – 2017, la distribución de paralelos cambió en el bachillerato. La distribución de paralelos ya no se realiza por especialización, sino simplemente por paralelos A, B, etc.

Este problema podría ser sencillo de resolver si se tuviera todo en la base de datos, pero los cursos y los paralelos son datos quemados en el

código de la aplicación. Adicionalmente, el diseño de la base de datos es primordial para este tipo de aplicación, pero la base de datos se encuentra completamente desnormalizada y hay ocasiones en que se crean inconsistencias en los datos, debido a que todo es ingresado por el usuario en todas las tablas sin crearse identificadores o claves foráneas.



registro	
◆	año: VARCHAR(20)
◆	curso: VARCHAR(30)
◆	paralelo: VARCHAR(30)
◆	fig_prof: VARCHAR(30)
◆	estado: VARCHAR(15)
◆	ult_mes_asist: INTEGER(11)
◆	cod_direccion: VARCHAR(10)
◆	n_matricula: VARCHAR(10)

**Figura 1.1: Tabla de registro de estudiantes**

En la Figura 1.1 se observa una de las muchas tablas que tiene la base de datos del sistema académico actual. Esta es la tabla registro. Se puede notar como el año lectivo, el curso y el paralelo están guardados como *varchar* y no representan un identificador puesto que las tablas *año lectivo* y *paralelo* no existen. Estos datos son guardados porque están quemados en el código.

### 1.3.5 Modernización

Además de todos los problemas descritos anteriormente, la unidad educativa tiene planes de modernización tanto en su infraestructura como en sus sistemas. Se desea brindar servicios a los padres de familia para que puedan estar al tanto de las materias vistas por sus hijos y esto conlleva a crear una aplicación web que tenga acceso a los datos.

## CAPÍTULO 2

### 2. IMPLEMENTACIÓN DE LA SOLUCIÓN PROPUESTA

En el capítulo anterior se presentaron todos los problemas que existen en el sistema académico actual que la unidad educativa utiliza. En este capítulo se detallará cómo se implementó la propuesta para solucionar los problemas críticos como: redondeo, seguridad y consistencia de los datos.

El alcance del proyecto se definió hasta la impresión de libretas parciales y quimestrales. Para esto, se debió crear año lectivo, cursos, paralelos, materias, ingreso de estudiantes, registro de estudiantes en los cursos, ingreso profesores, ingreso de notas parciales y quimestrales, ingreso de notas de conducta parcial y quimestral, cálculo de promedios y generación libretas por medio de alguna herramienta de reportes.

Para la implementación se utilizó la metodología SCRUM [8], en el cual se diseñaron prototipos, diagrama de clases, diagrama entidad-relación, tablas de control de acceso, etc.

En las sub-secciones siguientes, se describe el diseño creado durante la metodología SCRUM y sus historias de usuario, la tecnología usada, la arquitectura y el patrón de diseño *facade*; cómo se implementó la seguridad y los accesos por rol, y cómo se crearon los reportes por medio de la herramienta de reportes.

#### 2.1 Metodología SCRUM

El sistema propuesto fue desarrollado por medio de la metodología ágil SCRUM, en donde cada semana se realizaba stand ups para verificar los avances. El desarrollo fue dividido en 4 *sprints* y cada uno tuvo el número de historias de usuario necesarias para el avance.

### 2.1.1 Sprint 0

El *sprint* 0 tomó lugar del 23 de octubre al 6 de noviembre del 2015. Durante este tiempo se creó el ambiente de desarrollo con la tecnología escogida. Se creó un prototipo con una herramienta *mock-up* [9] y se diseñó el sistema por medio de diagramas de clase y entidad-relación.

Sistema Académico MTP

Año Lectivo: 2016 - 2017

Curso	Paralelo	Código	Materia
Decimo Año	B	D11120	Estudios Sociales
Primero de Bachillerato	Electricidad	CT1155	Instalaciones Eléctricas

Curso: Decimo Año Paralelo: B Código: D11120 Materia: Estudios Sociales

1Q 1P 1Q 2P 1Q 3P 2Q 1P 2Q 2P 2Q 3P

No	Apellidos	Nombres	Nota	Plan	Recomendación
1	Anchundia	Matias	5.0	Leer más	Leer más
2	Anchundia	Matias	7.0		
3	Anchundia	Matias	4.5	Leer más	Leer más
4	Anchundia	Matias	10.0		
5	Anchundia	Matias	10.0		
6	Anchundia	Matias	10.0		
7	Anchundia	Matias	10.0		

Guardar

**Figura 2.1: Mock-up para el ingreso de notas del sistema propuesto**

En la Figura 2.1 se tiene parte del prototipo creado. Se observa la vista para ingreso de notas en donde un profesor con su usuario y contraseña ya ingresada puede calificar al estudiante por materia.

En la Figura 2.2 se observa la parte más importante del diagrama entidad-relación para que el sistema tenga el comportamiento necesario y pueda crear las libretas correspondientes. Dentro de la figura 2.2 se tiene la tabla *estudiante* que se une directamente con la tabla *registro*. Aquí se verifica la información anual donde el *estudiante* está registrado en determinado *curso* durante un *año lectivo* en un respectivo *paralelo*. Cada *curso* debe tener sus propias *materias*, y dependiendo de la especialización, cada *paralelo* también debe tener sus propias *materias*. Se tiene además las *notas de conductas* parciales y quimestrales. A su

vez, se observa el ingreso de *profesores* por medio de *persona* y se les asigna el *rol* correspondiente para que puedan ingresar las *notas* a sus *cursos* asignados.



### 2.1.2 *Sprint* 1, 2 y 3

En los siguientes *sprints* de la fase de desarrollo, se siguieron paso a paso las historias de usuario definidas y se añadieron más conforme se necesitaba para poder completar el objetivo.

<b>Sprint</b>	<b>Historia de usuario</b>	<b>Fecha de inicio</b>	<b>Fecha de finalización</b>
<b>1</b>	Creación de ingreso y actualización de datos de un alumno	7/11/2015	4/12/2015
	Generación de matrícula por folio		
	Creación de cursos y paralelos		
	Creación de ingreso de materias con código curso, paralelo, año, mec		
	Creación de ingreso de personal con asignación de rol		
	Creación de asignación de profesores por materia		
	Creación del <i>schema</i>		
	Aplicación de tema para las vistas		
	Creación de año lectivo		
	Creación de cursos y paralelos		
	Registrar alumnos por paralelo		
<b>2</b>	Impresión de libretas parciales y quimestrales	5/12/2015	1/1/2016
	Asignación de materias a grupos		
	Ingreso de notas por quimestre y curso		
	Restricción para asignación de planes y recomendaciones en notas parciales		
	Creación de grupos		
	Creación de planes y recomendaciones		
	Ingreso de records de conducta		
	Ingreso de conducta parcial		
	Ingreso de conducta quimestral		
<b>3</b>	Creación de cierre de sesión	2/01/2016	29/01/2016
	Restricciones por rol		
	Cifrado de contraseña		

**Tabla 1: Historias de usuario por sprint.**

En la Tabla 1 se puede observar cómo se avanzó durante todo el desarrollo del sistema académico propuesto. La tabla presenta el número del *sprint*, la historia de usuario, la fecha de inicio del *sprint* y la fecha de finalización del *sprint*.

Cada historia de usuario estuvo pensada para poder lograr el alcance y cumplir con el objetivo de poder imprimir las libretas parciales y quimestrales cada que se lo requería.

En todas las iteraciones del desarrollo se consideraron funcionalidades extras tales como:

- Retroalimentación del sistema.
- Creación de nuevas tablas para datos constantes.
- Cifrado de contraseña.

## 2.2 Tecnologías

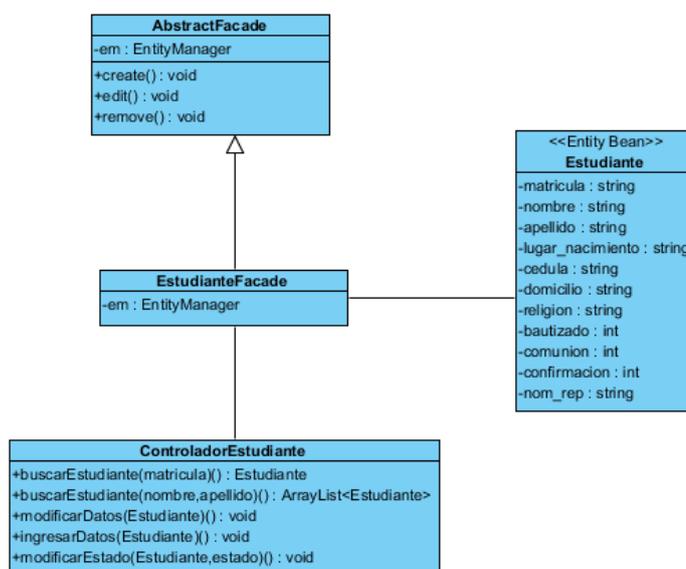
La tecnología escogida para el desarrollo del “Sistema para control de calificaciones escolares” está compuesta por: Java EE 7 [10] con Java 1.8.0, JavaServer Faces (JSF) 2.2.4 [11] con Primefaces 5.3 [12], JasperReports 5.1.2 [13], Spring Security 4.0.3 [14], Maven 3.0.5 [15], Java Persistence API (JPA) 2.1 [16] con su implementación ORM [17]: EclipseLink [18], MySql 5.7.9, Git [19] y Glassfish 4.1 [20].

Mientras que las herramientas fueron Netbeans IDE 8.0.2 [21], JasperSoft Studio 6.2.0 [22], MySql Workbench 6.3 [23] y Git Bash para Windows 2.7.2 [24].

## 2.3 Arquitectura y patrón de diseño *facade*

La arquitectura de la aplicación viene dada por Java EE, la cual está compuesta por 3 capas: la capa de presentación por medio de JSF, la capa de persistencia por medio de la implementación de JPA y la capa lógica mediante clases y controladores.

El diagrama entidad-relación dio paso a generar entidades mediante el ORM que luego se convirtieron en los objetos de la aplicación y tuvieron una conexión abstracta con la base de datos. Para poder acceder a estas entidades, se usó el patrón de diseño *facade*, el cual esconde la complejidad de la implementación.



**Figura 2.3: Representación del patrón de diseño *facade* con la entidad *Estudiante***

En la Figura 2.3 se aprecia la implementación del patrón de diseño *facade* con la entidad *Estudiante*. Se observa cómo se abstrae por completo el comportamiento de las funciones *create()*, *edit()* y *remove()* que son las que manejan la información para realizar la respectiva operación en la base de datos.

## 2.4 Módulos

El sistema propuesto se implementó por medio de módulos, tales como: *año lectivo*, *estudiante*, *personal*, *notas*, *conducta* e *informes*. Cada módulo tiene las siguientes vistas:

- Año lectivo
  - Cursos

- Grupos
  - Materias
  - Planes y recomendaciones
  - Subniveles
- Estudiante
  - Registro de estudiante
  - Nuevo estudiante
  - Editar estudiante
- Personal
  - Nuevo usuario
  - Asignación de profesores
- Notas
  - Ingreso de nota parcial
  - Ingreso de examen quimestral
- Conducta
  - Records de conducta
  - Ingreso de conducta
    - Parcial
    - Quimestral
- Informes
  - Impresión de libreta parcial
  - Impresión de libreta quimestral

## 2.5 Seguridad

Para la implementación de las seguridades, se usó el *framework* Spring Security. Aquí se manejaron reglas de acceso, verificación de usuario y contraseña, cifrado de la contraseña y manejo de excepciones como el error 403.

En la Tabla 2 se tiene la información de los accesos de los roles por módulo o vista. En la aplicación se definieron los 3 roles más importantes que son: *administrador*, *profesor* e *inspector*. Cada uno tiene características diferentes y

por eso tienen accesos distintos a la información. Cuando un rol en específico tiene acceso a la vista y la información debe ser limitada, se lo maneja por medio de *queries*, como es el caso de las notas. Un profesor tiene acceso a la vista, pero solamente deberá ingresar o modificar las notas de las materias que dicta.

Módulo	Vista	ADMIN		PROFESOR		INSPECTOR	
		Lectura	Escritura	Lectura	Escritura	Lectura	Escritura
Año lectivo	Cursos	x	x				
	Materias	x	x				
	Grupos	x	x				
	Planes y recomendaciones	x	x				
	Subniveles	x	x				
Personal	Nuevo usuario	x	x				
	Asignación de profesores	x	x				
Estudiante	Nuevo estudiante	x	x				
	Editar estudiante	x	x				
	Registro	x	x				
Notas	Nota parcial	x	x	x	x		
	Examen quimestral	x	x	x	x		
Conducta	Conducta parcial	x	x			x	x
	Conducta quimestral	x	x			x	x
	Records de conducta			x	x	x	x
Informes		x	x				

**Tabla 2: Lista de accesos de roles por módulos o vista**

## **2.6 Reportes**

Para generar las libretas, se usó la herramienta JasperSoft el cual maneja los reportes de JasperReports. Para generar cada libreta, se realizaron funciones en la base de datos que obtienen información de los estudiantes, las materias y las notas que obtuvieron.

## CAPÍTULO 3

### 3. DETALLES DE LA SOLUCIÓN Y MIGRACIÓN DE DATOS

En este capítulo se detalla la solución implementada, y cómo se migraron los datos desde la base actual a la propuesta.

#### 3.1 Redondeo

El problema más crítico del sistema académico actual es el de redondeo como se lo analizó en el capítulo 1. Para solucionar este problema se decidió usar el lenguaje Java y usar el tipo de dato *BigDecimal* para manejar la información de las notas ingresadas. *BigDecimal* maneja todos los tipos de redondeos, siendo el redondeo aritmético el necesario para esta aplicación.

#### 3.2 Seguridad

La seguridad es uno de los principales problemas que tiene el sistema académico actual que posee la unidad educativa. Con el sistema propuesto, se resolverán los problemas de la siguiente manera:

- Ya no existirán carpetas compartidas: al ser una aplicación web hecha en java, se deberá crear el archivo `.war` del sistema propuesto y colocarlo en el servidor de aplicaciones, en este caso Glassfish. Los usuarios no tendrán acceso a la base de datos por medio de los reportes, puesto que no se encuentran compartidos.
- Los usuarios ya no tendrán acceso a la conexión de la base de datos: la conexión solamente existirá en los archivos de configuración de la aplicación y no habrá que configurar los computadores de la institución, puesto que solo necesitan un navegador web.
- Contraseña cifrada: se cifra la contraseña mediante el algoritmo *BCrypt* que lo provee Spring Security.
- Acceso a la información: para que los usuarios puedan tener acceso a la información, se debía quemar en el código fuente los identificadores

permitidos por componente. Ahora se creó una lista de control de acceso (ACL), el cual se divide por módulos o vistas de la aplicación.

### **3.3 Procesamiento e implementación**

El reporte creado en JasperReports no tiene ninguna información almacenada, puesto que así se evita el procesamiento innecesario cada que se las desea imprimir.

En la implementación se evitó calcular los promedios en lugares diferentes. Todos los cálculos están hechos en la aplicación y no en la herramienta de reportes.

### **3.4 Mantenimiento**

Con el rediseño de la base de datos se crean identificadores y claves foráneas que permiten solucionar la inconsistencia de los datos. El programador deberá enfocarse simplemente en la codificación del programa, más no en cómo se guardan los registros siempre que se insertan, modifican o eliminan.

Toda la información que se presenta en el sistema se obtiene de forma dinámica y no existe código quemado innecesario, lo que mejora mucho cuando hay cambios grandes en el sistema para crear una nueva funcionalidad.

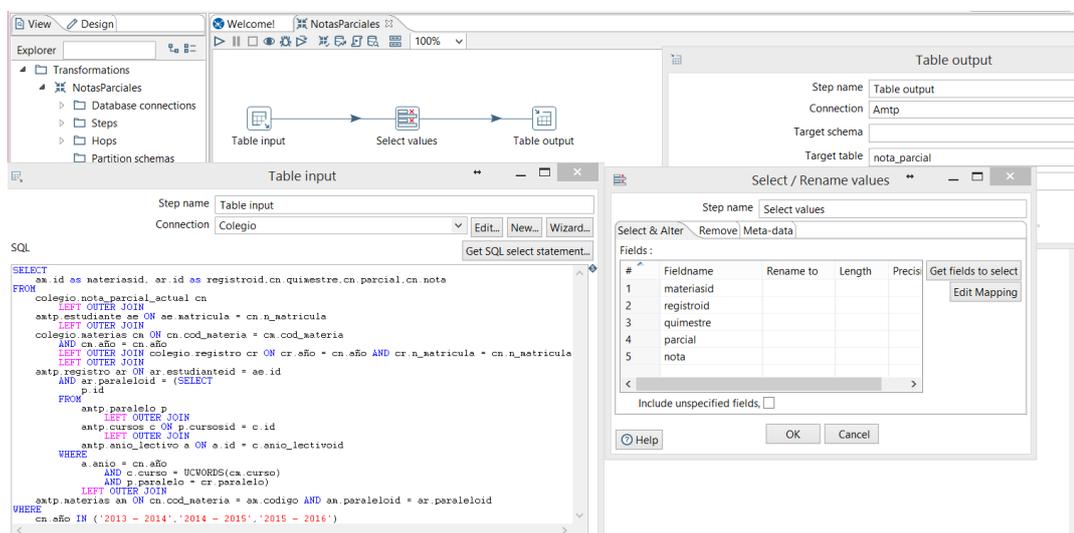
### **3.5 Migración de datos**

Puesto que se tuvo que rediseñar la base de datos para que funcione la solución propuesta, fue necesario probar el sistema con datos reales.

Para esto, se usó el proceso de extracción, transformación y carga (ETL) para la migración de datos. Se utilizó la herramienta Data Integration de Pentaho [25].

En la Figura 3.1 se observa un trabajo para el proceso ETL de las *notas parciales*. Se seleccionó en la tabla origen los datos correspondientes a los años lectivos 2013 – 2014, 2014 – 2015 y 2015 – 2016. Luego se seleccionaron

los campos deseados para posteriormente cargarlos a la tabla destino de la nueva base de datos. Estas tablas en la imagen son: *Table input* y *Table output*.



**Figura 3.1: Proceso ETL de las notas parciales**

Además de *notas parciales*, se cargaron *años lectivos*, *cursos*, *paralelos* y *materias*. Todos tuvieron éxito en el proceso de carga, excepto las *notas parciales*. A pesar de que la base de datos se llenó por completo y aparentemente tenía relaciones creadas, los datos no se presentaban en la aplicación. Esto pudo deberse a que la base de datos actual tiene muy segregados los datos y por estar completamente desnormalizada, crea conflictos a la hora de extraer información. Se deberá hacer un análisis más profundo y verificar cuál es la información que no se está cruzando correctamente en comparación con la base de datos destino. Probablemente, deba crearse algún proceso manual para llenar la base de datos destino antes de cargar los datos originales.

## CONCLUSIONES Y RECOMENDACIONES

### Conclusiones

A continuación, algunas conclusiones basadas en los objetivos definidos en el capítulo 1:

1. El sistema propuesto denominado “Sistema para control de calificaciones escolares” es una solución basada en los problemas que el sistema académico actual tiene.
2. Actualmente el módulo *notas*, refleja el comportamiento que existe en el sistema educativo del país y de la unidad educativa, cumpliendo con el alcance definido inicialmente.
3. Cumple con el requerimiento de modernización de la unidad educativa porque está implementado con las últimas versiones de las tecnologías usadas que le permiten a futuro mejorar la generación de libretas y el mantenimiento del sistema.
4. El sistema propuesto resolvió el problema de redondeo usando el tipo de dato *BigDecimal* que Java ofrece, puesto que implementa el redondeo aritmético común.
5. El sistema propuesto tiene buenas y nuevas políticas de seguridad debido a que todo está abstraído por el *framework* Spring Security. Se crearon roles y el acceso a la información fue definido por medio de un listas de control.

La seguridad de la aplicación y de la base de datos también mejoró porque ya no existirán carpetas compartidas y conexiones de base de datos en computadores individuales.

6. El diseño de la base de datos mejoró mucho, puesto que ahora soporta materias extracurriculares y excepciones en materias para que no formen parte del promedio mec. Esto ya no se lo hace quemando código, sino mediante *queries* y funciones en la base de datos.

Adicionalmente, la base de datos ya no se encuentra desnormalizada lo que conlleva a un mejor mantenimiento y mejora la calidad de programación.

### **Recomendaciones**

1. Debido a que en la planificación de los *sprints* no se agregó el registro de eventos, una mejora muy importante sería crear tablas en la base de datos para guardar estos eventos, especialmente el cambio de notas parciales o quimestrales.
2. El sistema propuesto a futuro deberá integrarse con un nuevo sistema financiero para poder realizar las operaciones necesarias como matrículas o pago de pensiones. Esto es algo que el sistema académico actual tiene y al ser reemplazado, el sistema propuesto deberá tener la misma habilidad.
3. Para solucionar el problema de la migración de los datos, será necesario crear un proceso manual. El proceso deberá estar compuesto por la selección de notas parciales por cada quimestre y paralelo. A pesar de ser un trabajo tedioso, es necesario para poder encontrar inconsistencias o duplicados.

## BIBLIOGRAFÍA

- [1] Microsoft. How to implement custom rounding procedures [en línea]. Disponible en: <https://support.microsoft.com/en-us/kb/196652>.
- [2] Ministerio de Educación del Ecuador. Educación general básica [en línea]. Disponible en: <http://educacion.gob.ec/educacion-general-basica>.
- [3] Ministerio de Educación del Ecuador. Bachillerato general unificado [en línea]. Disponible en: <http://educacion.gob.ec/bachillerato-general-unificado/>.
- [4] Microsoft. Visual Basic 6.0 [en línea]. Disponible en: <https://msdn.microsoft.com/en-us/vstudio/ms788229.aspx>.
- [5] MySQL. MySQL [online]. Disponible en: <https://www.mysql.com/>.
- [6] Crystal Reports. Crystal Reports [en línea]. Disponible en: <http://www.crystalreports.com/>.
- [7] Kane, (2011, marzo). Data Source Name (DSN) [en línea]. Disponible en: <http://whatis.techtarget.com/definition/data-source-name-DSN>.
- [8] Sowon. Scrum [online]. Disponible en: <http://scrummethodology.com/>.
- [9] H. Keisser, (2014, mayo). What is a Mock-up? [en línea]. Disponible en: <https://experience.sap.com/basics/what-is-a-mock-up/>
- [10] Oracle. Java EE [en línea]. Disponible en: <http://www.oracle.com/technetwork/java/javaee/overview/index.html>
- [11] Oracle. JavaServer Faces Technology [en línea]. Disponible en: <http://www.oracle.com/technetwork/java/javaee/jspserverfaces-139869.html>
- [12] Primefaces. Primefaces [online]. Disponible en: <http://www.primefaces.org/>
- [13] JasperSoft Community. JasperReports Library [en línea]. Disponible en: <http://community.jaspersoft.com/project/jasperreports-library>
- [14] Spring. Spring Security [online]. Disponible en: <http://projects.spring.io/spring-security/>
- [15] Apache Maven. Apache Maven Project [en línea]. Disponible en: <https://maven.apache.org/>
- [16] Oracle. Java Persistence API [en línea]. Disponible en: <http://www.oracle.com/technetwork/java/javaee/tech/persistence-jsp-140049.html>
- [17] Hibernate. What is Object/Relational Mapping? [en línea]. Disponible en: <http://hibernate.org/orm/what-is-an-orm/>

- [18] Eclipse. EclipseLink [en línea]. Disponible en: <http://www.eclipse.org/eclipselink/>
- [19] Git. Git [en línea]. Disponible en: <https://git-scm.com/>
- [20] Oracle. Glassfish [en línea]. Disponible en: <https://glassfish.java.net/>
- [21] Oracle. Netbeans [en línea]. Disponible en: <https://netbeans.org/>
- [22] JasperSoft. JasperSoft [en línea]. Disponible en: <http://www.jaspersoft.com/>
- [23] MySQL. MySQL Workbench [en línea]. Disponible en: <https://www.mysql.com/products/workbench/>
- [24] Git. Git Bash para Windows [en línea]. Disponible en: <https://git-scm.com/download/win>
- [25] Pentaho. Data Integration [en línea]. Disponible en: <http://www.pentaho.com/product/data-integration>