



**ESCUELA SUPERIOR POLITÉCNICA AGROPECUARIA DE
MANABÍ MANUEL FÉLIX LÓPEZ**

CARRERA INFORMÁTICA

**TESIS PREVIA A LA OBTENCIÓN DEL TÍTULO DE:
INGENIERO EN INFORMÁTICA**

TEMA:

**SISTEMA INFORMÁTICO DE REGISTRO DE MATRICULACIÓN
Y CALIFICACIONES EN LA UNIDAD EDUCATIVA ARISTOS DE
LA CIUDAD DE CALCETA DEL CANTÓN BOLÍVAR**

AUTORES:

**MARTHA GABRIELA CRUZ LÓPEZ
JUAN CARLOS MUÑOZ MENDOZA**

TUTOR:

ING. JORGE ANTONIO PÁRRAGA ÁLAVA

CALCETA, OCTUBRE 2012

DERECHO DE AUTORÍA

Cruz López Martha Gabriela y Juan Carlos Muñoz Mendoza, declaramos bajo juramento que el trabajo aquí descrito es de su autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos los derechos de propiedad intelectual a la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López, según lo establecido por la Ley de Propiedad Intelectual y su Reglamento.

MARTHA G. CRUZ LÓPEZ

JUAN C. MUÑOZ MENDOZA

CERTIFICACIÓN DE TUTOR

Jorge Antonio Párraga Álava certifica haber tutelado la tesis **SISTEMA INFORMÁTICO DE REGISTRO DE MATRICULACIÓN Y CALIFICACIONES EN LA UNIDAD EDUCATIVA ARISTOS DE LA CIUDAD DE CALCETA DEL CANTÓN BOLÍVAR**, que ha sido desarrollada por Martha Gabriela Cruz López y Juan Carlos Muñoz Mendoza, previa a la obtención del título de Ingeniero en Informática, de acuerdo al **REGLAMENTO PARA LA ELABORACIÓN DE TESIS DE GRADO DE TERCER NIVEL** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

ING. JORGE ANTONIO PÁRRAGA ÁLAVA

APROBACIÓN DEL TRIBUNAL

Los suscritos miembros del tribunal correspondiente, declaramos haber **APROBADO** la tesis titulada **SISTEMA INFORMÁTICO DE REGISTRO DE MATRICULACIÓN Y CALIFICACIONES EN LA UNIDAD EDUCATIVA ARISTOS DE LA CIUDAD DE CALCETA DEL CANTÓN BOLÍVAR**, que ha sido propuesta, desarrollada y sustentada por Martha Gabriela Cruz López y Juan Carlos Muñoz Mendoza, previa a la obtención del título de Ingeniero en Informática, de acuerdo al **REGLAMENTO PARA LA ELABORACIÓN DE TESIS DE GRADO DE TERCER NIVEL** de la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

ING. ÁNGEL VÉLEZ MERO
MIEMBRO

ING. KELLY LUCAS SALDARRIAGA
MIEMBRO

ING. GUSTAVO MOLINA GARZÓN
PRESIDENTE

AGRADECIMIENTO

En primer lugar agradezco a Dios por permitirme estar aquí, el que está sobre todo y ante todo ya que en ningún momento me desampara, dándome fuerza y valor para salir adelante en cada etapa de mi vida.

En la vida hay momentos de cambios donde hay que tomar decisiones para emprender nuevos propósitos. Yo estoy en ese instante de cambio, evolución, aprendizaje y superación gracias a mis padres y amigos, ellos me brindaron todo el apoyo y la paciencia necesaria para culminar mi carrera.

También un eterno agradecimiento a esta prestigiosa universidad la cual abre sus puertas a jóvenes como nosotros, preparándome para un futuro competitivo y formándome como persona de bien por medio de docentes altamente capacitados.

A nuestro tutor el Ing. Jorge Párraga Álava quien con sus instrucciones y paciencia se convirtió en un pilar importante para la culminación exitosa de este trabajo.

De manera especial a los directivos de la Unidad Educativa ARISTOS por depositar su confianza para desarrollar la aplicación informática.

MARTHA G. CRUZ LÓPEZ

AGRADECIMIENTO

Agradezco a Dios por sobre todas las cosas, por brindarme sabiduría y fortaleza para poder sobrellevar los obstáculos que se me presentan en el camino.

A mis amados padres, por haber estado siempre a mi lado demostrándome su apoyo y confianza en todo momento.

A mis profesores quienes con sus conocimientos y enseñanzas me inculcaron todos los valores morales, éticos y profesionales los cuales los pondré en práctica durante toda mi vida.

A el Ing. Jorge Antonio Párraga Álava, mi tutor de Tesis, quien ha sido parte fundamental de este proyecto guiándome en los análisis y comentarios suscitados en el desarrollo de la tesis.

Así mismo a todas las personas que han contribuido colaborando de la manera más humilde y desinteresada.

JUAN C. MUÑOZ MENDOZA

DEDICATORIA

Dedico este trabajo a Dios y a mis padres. A Dios porque ha estado conmigo en cada paso que doy, cuidándome y dándome fortaleza para continuar. A mis padres, quienes a lo largo de mi vida han velado por mi bienestar y educación siendo mi apoyo en todo momento.

Por eso deseo expresar que tengan siempre presente que no hay obstáculo capaz de imponerse; si queremos podemos llegar más lejos, si queremos podemos llegar más alto, si queremos podemos hacer lo que sea solo hay que proponérselo.

MARTHA G. CRUZ LÓPEZ

DEDICATORIA

A Dios, ser supremo quien me ha iluminado desde el cielo despejando cualquier duda y regalándome un poquito de sabiduría para poder terminar con éxito este proyecto.

A mis padres Sr. Leonardo Muñoz y Sra. Teresa Mendoza, quienes con esfuerzo y sacrificio han sabido compartir mis momentos más difíciles dándome el apoyo incondicional en todos los aspectos.

JUAN C. MUÑOZ MENDOZA

CONTENIDO GENERAL

DERECHO DE AUTORÍA	II
CERTIFICACIÓN DE TUTOR	III
AGRADECIMIENTO.....	V
AGRADECIMIENTO.....	VI
DEDICATORIA.....	VII
DEDICATORIA.....	VIII
CONTENIDO GENERAL.....	IX
CONTENIDO DE CUADROS Y FIGURAS	XII
RESUMEN.....	XIV
PALABRAS CLAVES	XIV
ABSTRACT	XV
KEYWORDS.....	XV
CAPÍTULO I. ANTECEDENTES	1
1.1 PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA.....	1
1.2 JUSTIFICACIÓN.....	3
1.3 OBJETIVOS.....	5
1.3.1 OBJETIVO GENERAL	5
1.3.2 OBJETIVOS ESPECÍFICOS.....	5
1.4 HIPÓTESIS, PREMISAS Y/O IDEAS A DEFENDER	6
CAPÍTULO II. MARCO TEÓRICO.....	7
2.1 UNIDAD EDUCATIVA ARISTOS-CALCETA Y SU TRAYECTORIA EN EL ENTORNO	7
2.1.1 ANTECEDENTES	7
2.2 MISIÓN Y VISIÓN.....	8
2.2.1 MISIÓN.....	8
2.2.2 VISIÓN	8
2.3 FILOSOFÍA	8
2.4 UBICACIÓN GEOGRÁFICA.....	9
2.5 GENERALIDADES DE INGENIERÍA DE SOFTWARE	10
2.5.1 TIPOS DE SOFTWARE	11
2.5.1.1 SOFTWARE DE SISTEMAS	11
2.5.1.2 SOFTWARE DE APLICACIÓN	12
2.5.1.3 SOFTWARE DE GESTIÓN	12
2.5.1.4 SOFTWARE DE PROGRAMACIÓN	13
2.5.2 MODELOS DE DESARROLLO DE SOFTWARE	13
2.5.2.1 RUP (MODELO RACIONAL).....	14
2.5.3 MODELADO UML	19
2.5.3.1 TIPOS DE DIAGRAMAS UML.....	20

2.5.4 PRUEBAS DE SOFTWARE.....	24
2.5.4.1 TIPOS DE PRUEBAS DE SOFTWARE	25
2.6 TIPOS DE PROGRAMACIÓN	29
2.6.1 PROGRAMACIÓN POR CAPAS	30
2.6.1.1 CARACTERÍSTICAS DE LA PROGRAMACIÓN EN CAPAS	30
2.6.1.2 VENTAJAS	33
2.7 ASPECTOS RELEVANTES DE UNA BASE DE DATOS.....	33
2.7.1 MODELOS DE BASE DE DATOS	34
2.7.2 SISTEMA DE GESTIÓN DE BASE DE DATOS (SGBD)	35
2.7.2.1 ABSTRACCIÓN DE LA INFORMACIÓN	35
2.7.2.2 INDEPENDENCIA	35
2.7.2.3 REDUNDANCIA MÍNIMA	36
2.7.2.4 CONSISTENCIA.....	36
2.7.2.5 SEGURIDAD	36
2.7.2.6 INTEGRIDAD.....	36
2.7.2.7 RESPALDO Y RECUPERACIÓN.....	37
2.7.2.8 CONTROL DE LA CONCURRENCIA	37
2.7.2.9 TIEMPO DE RESPUESTA	37
2.8 MODELO ENTIDAD-RELACIÓN.....	37
2.8.1 ENTIDAD.....	38
2.8.2 ATRIBUTOS.....	38
2.8.3 RELACIÓN.....	38
2.8.3.1 CARDINALIDAD DE LAS RELACIONES.....	39
2.9 HERRAMIENTAS DE DESARROLLO DE SOFTWARE.....	39
2.9.1 MICROSOFT SQL SERVER 2008 R2.....	39
2.9.1.1 CARACTERISTICAS GENERALES DE SQL SERVER 2008 R2.....	40
2.9.2 PLATAFORMA .NET.....	41
2.9.2.1 VISUAL BASIC.NET 2010.....	42
CAPÍTULO III. DESARROLLO METODOLÓGICO.....	47
3.1 MÉTODO INFORMÁTICO.....	47
3.1.1 FASE DE INICIO.....	47
3.1.1.1 MODELADO DEL NEGOCIO	47
3.1.1.2 REQUISITOS.....	53
3.1.2 FASE DE ELABORACIÓN.....	55
3.1.2.1 DIAGRAMA UML CASOS DE USO	55
3.1.2.2 DIAGRAMA DE FLUJOS DE DATOS DEL PROCESO DE REGISTRO DE MATRICULACIÓN.....	57
3.1.2.3 DIAGRAMA DE FLUJOS DE DATOS DEL PROCESO DE REGISTRO DE CALIFICACIONES	58
3.1.2.4 DISEÑO DE LA ESTRUCTURA DE LA BASE DE DATOS	59

3.1.2.5 DISEÑO DE INTERFACES	71
3.1.3 FASE DE CONSTRUCCIÓN	74
3.1.3.1 IMPLEMENTACIÓN	74
3.1.3.2 PRUEBAS.....	75
3.1.4 FASE DE TRANSICIÓN.....	77
3.2 TÉCNICAS.....	78
3.3 RECURSOS	78
3.3.1 TALENTOS HUMANOS.....	78
3.3.2 MATERIALES	78
3.3.3 TECNOLÓGICOS	78
3.3.3.1 HARDWARE.....	78
3.3.3.2 SOFTWARE	79
3.3.4 PRESUPUESTO Y FUENTES DE FINANCIAMIENTO.....	79
3.3.4.1 PRESUPUESTO	79
3.3.4.2 FUENTES DE FINANCIAMIENTO.....	80
CAPITULO IV. RESULTADOS Y DISCUSIÓN.....	81
4.1 RESULTADOS	81
4.1.1 COMPARATIVO ENTRE LA OBSERVACIÓN PREVIA Y DESPUES DE LA EJECUCIÓN DE LA APLICACIÓN.	82
4.2 DISCUSIÓN.....	85
CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES	87
5.1 CONCLUSIONES.....	87
5.2 RECOMENDACIONES.....	88
BILIOGRAFÍA.....	89
ANEXOS.....	94

CONTENIDO DE CUADROS Y FIGURAS

Figura 03.01: Casos de Uso del proceso de registro de matriculación	55
Figura 03.02: Caso de uso del proceso de registro de calificaciones	56
Figura 03.03: Diagrama de flujo de datos del proceso de registro de matriculación	58
Figura 03.04: Diagrama de flujo de datos del proceso de registro de calificaciones....	59
Figura 03.05: Estructura de la base de datos.....	53
Figura 03.06: Ingreso al sistema.....	72
Figura 03.07: Pantalla principal.....	72
Figura 03.08: Formulario de registro de matriculación	73
Figura 03.09: Formulario de registro de calificaciones	73
Figura 03.10: Esquema de la arquitectura de programación en tres capas.....	74
Tabla 03.01: Listado de las materias impartidas en la Unidad ARISTOS.....	49
Tabla 03.02: Registro de calificaciones de los estudiantes.	48
Tabla 03.03: Proceso de registro de matriculación.	50
Tabla 03.04: Requisitos funcionales del software	53
Tabla 03.05: Requisitos no funcionales del software.....	54
Tabla 03.06: Requisitos de implementación del software.....	54
Tabla 03.07: Descripción del caso de uso del proceso de registro de matriculación ...	56
Tabla 03.08: Descripción del caso de uso de registro de calificaciones	57
Tabla 03.09: Diccionario de datos de la tabla alumnos	63
Tabla 03.10: Diccionario de datos de la tabla representante.....	64
Tabla 03.11: Diccionario de datos de la tabla persona.....	64
Tabla 03.12: Diccionario de datos de la tabla docente.....	65
Tabla 03.13: Diccionario de datos de la tabla distribución.....	65
Tabla 03.14: Diccionario de datos de la tabla periodo lectivo.....	66
Tabla 03.15: Diccionario de datos de la tabla quimestre	66
Tabla 03.16: Diccionario de datos de la tabla nivel	66
Tabla 03.17: Diccionario de datos de la tabla bimestre	67
Tabla 03.18: Diccionario de datos de la tabla AEB	67
Tabla 03.19: Diccionario de datos de la tabla Asignatura.....	67
Tabla 03.20: Diccionario de datos de la tabla Usuario	68
Tabla 03.21: Diccionario de datos de la tabla parroquia.....	68
Tabla 03.22: Diccionario de datos de la tabla matricula	68

Tabla 03.23: Diccionario de datos de la tabla provincia	69
Tabla 03.24: Diccionario de datos de la tabla Cantón	69
Tabla 03.25: Diccionario de datos de la tabla asignatura matricula.....	69
Tabla 03.26: Diccionario de datos de la tabla calificaciones.....	70
Tabla 03.27: Diccionario de datos de la tabla configuración.....	70
Tabla 03.28: Presupuesto.....	79
Tabla 04.01: Cuadro comparativo entre la observación previa y después de la ejecución de la aplicación (Registro de matriculación)	82
Tabla 04.02: Cuadro comparativo entre la observación previa y después de la ejecución de la aplicación (Registro de calificaciones).....	83
Tabla 04.03: Cuadro comparativo entre la observación previa y después de la ejecución de la aplicación (Consulta de datos)	83
Tabla 04.04: Cuadro comparativo entre la observación previa y después de la ejecución de la aplicación (Adquisición de reportes)	84

RESUMEN

El objetivo primordial de esta investigación fue crear un sistema informático que permita el registro de matriculación y de calificaciones en la unidad educativa Aristos de la ciudad de Calceta del Cantón Bolívar, con el firme propósito de agilizar los procesos y así facilitar las tareas en el departamento de Secretaría. El modelo de software que se empleó para la ejecución de la aplicación fue el Modelo RUP (Rational Unified Process). El cual permitió la recopilación de información para definir los requisitos y la arquitectura del sistema. Se empleó el paradigma de programación tres capas, se elaboraron los casos de uso y los diagramas de flujos de datos que ayudaron a describir lo que el sistema debe hacer. Luego se procedió a elaborar la base de datos en Microsoft SQL Server 2008 R2 que se acopló al trato y manipulación de la información. Seguido a esto se elaboró la interfaz gráfica del sistema y se incorporó la lógica de la aplicación mediante la codificación, para el efecto se utilizó la plataforma de desarrollo Microsoft Visual Basic 2010. Una vez validada la aplicación se logró probar el producto en su entorno de ejecución final.

PALABRAS CLAVES

Sistema Informático, Matriculación, Calificaciones, Unidad Educativa, Aplicación.

ABSTRACT

The primary objective of this research was to create a computer system that allows the vehicle register and control skills in the educational unit Aristos Calceta city of Canton Bolívar, to the effect model was used software development model RUP (Rational Unified Process). It began with the gathering of information to define the requirements and architecture of the system through use cases and data flow diagrams. Was used programming paradigm in three layers and the tool Visual Basic 2010, and then proceeded to develop the database in Microsoft SQL Server 2008 R2. Immediately drew the GUI system and joined the application logic thereto. With the completion of these activities, the respective tests were performed in the system execution environment end and checked for optimum performance confirming that the registration process streamlined enrollment and grades in the institution.

KEYWORDS

Information System, registration, qualifications, Educational Unit, Application.

CAPÍTULO I. ANTECEDENTES

1.1 PLANTEAMIENTO Y FORMULACIÓN DEL PROBLEMA

Con el transcurso del tiempo la tecnología avanza y los centros educativos necesitan adquirirla y aplicarla, para agilizar sus actividades de forma eficiente y suplir las deficiencias que se observaban en las distintas instituciones, ya que es común el uso exagerado de papeles, carpetas y otros materiales de oficina que se utilizan para archivar datos de suma importancia, lo que origina un exceso de hojas impidiendo que la información sea obtenida de manera rápida y oportuna, para lo cual requiere de herramientas tecnológicas que les ayude a realizar los procesos de una manera eficiente y eficaz. Actualmente existen diferentes aplicaciones que sirven para que muchos procesos aumenten su eficiencia, y si una organización, establecimiento o empresa requiere ofrecer a sus clientes la calidad en el servicio que otorga, se hace emergente la búsqueda de soluciones que faciliten los procesos.

No está exenta de esta necesidad la Unidad Educativa Aristos Calceta, institución que ha formado a la juventud y a la niñez con calidad-calidez, en la parte académica; no obstante en el área administrativa, específicamente en secretaría general se observa que los procesos registro de matriculación y calificaciones se los realiza manualmente en hojas con un formato específico realizado en Microsoft Office Word 2007 para el ingreso y registro de matriculación de los estudiantes y otro formato realizado en Microsoft Office Excel 2007 para el registro de las calificaciones, convirtiéndose en un proceso tedioso al momento de requerir información de los estudiantes.

Con estos antecedentes los autores del presente proyecto plantean la siguiente interrogante:

¿De qué manera optimizar los procesos de registro de matriculación y de calificaciones de la Unidad Educativa Aristos, de la ciudad de Calceta?

1.2 JUSTIFICACIÓN

La Unidad Educativa Aristos Calceta, es una institución que tiene como propósito formar a niños-jóvenes con valores y con gran capacidad de conocimientos para desenvolverse profesionalmente en el futuro, estando al día con los avances tecnológicos para mejorar la calidad de educación. En esta institución se determinó que con el aumento de estudiantes en los últimos años se registró una gran cantidad de información por parte del departamento de secretaría, debido a que los procesos de registro de matriculación y de calificaciones de los estudiantes se los realiza de forma manual en hojas cuyo diseño se elaboran en Microsoft Excel y Microsoft Word 2007, sin mecanismos adecuados que agilicen, garanticen y organicen los datos de manera segura y sin riesgo de pérdida o alteración de la información de los estudiantes.

Tomando como referencia la necesidad de contar con una herramienta tecnológica se optó por crear en el departamento de secretaría un sistema que agilice los procesos de registro de matriculación y de calificaciones que permita almacenar, organizar y manipular la información de forma veraz y oportuna. Por esta razón el sistema de registro de matriculación y de calificaciones que se llevará a efecto en la Unidad Educativa Aristos Calceta mejorará sus servicios y brindará la información de forma rápida moderna y actualizada que conlleve a un mejoramiento institucional y así elevar estándares de calidad, beneficiando a padres de familia, estudiantes y específicamente a el área de secretaría.

Con la culminación del sistema se ahorró tiempo y dinero; al ingresar la información se evitó la realización de reportes y cálculos manuales, disminuyendo el uso exagerado de papel y demás instrumentos de oficina, que nos ayudan a contribuir con la salvaguarda del planeta, el mismo que se encuentra en serias dificultades actualmente debido a la contaminación provocada por el hombre.

La creación de esta investigación además se justifica de acuerdo al reglamento para la elaboración de tesis de tercer nivel de la ESPAM MFL que en su Art. 2 manifiesta: “Todo tema de tesis de grado estará relacionado con las líneas de investigación de la carrera del postulante, enmarcado en las áreas y prioridades de investigación establecidas por la ESPAM MFL en concordancia con el Plan Nacional para el Buen Vivir”.

Y de acuerdo lo que decreta en su Art. 7: “Las tesis de investigación laboral son realizados por el postulante(s) en centros académicos, de investigación, producción o servicio local, regional, nacional o internacional afín a su formación profesional.

1.3 OBJETIVOS

1.3.1 OBJETIVO GENERAL

- Crear un Sistema Informático en la Unidad Educativa Aristos Calceta con la finalidad de optimizar los procesos de registro matriculación y de calificaciones.

1.3.2 OBJETIVOS ESPECÍFICOS

- Recopilar información de los procesos de registro de matriculación y de calificaciones realizados en la institución.
- Diseñar la base de datos optimizada para la manipulación de información.
- Desarrollar la aplicación, basándose en los requerimientos del usuario.
- Validar el correcto funcionamiento de la aplicación.
- Implementar el sistema informático en la institución.

1.4 HIPÓTESIS, PREMISAS Y/O IDEAS A DEFENDER

- Necesidad de crear un Sistema Informático de registro de matriculación y de calificaciones en la Unidad Educativa Aristos Calceta.

- La implementación del Sistema Informático de registro de matriculación y de calificaciones en la Unidad Educativa Aristos Calceta optimizará los procesos.

CAPÍTULO II. MARCO TEÓRICO

2.1 UNIDAD EDUCATIVA ARISTOS-CALCETA Y SU TRAYECTORIA EN EL ENTORNO

2.1.1 ANTECEDENTES

Para entender la propuesta educativa de Aristos-Calceta se centra la atención en la persona de Josefina Viteri Mendoza, su fundadora, y estudiar en este aspecto sus inquietudes como educadora: La constante preocupación por la educación de los niños y jóvenes.

Aristos, nombre de procedencia griega que significa EL MEJOR y de aristocracia que proyecta CALIDAD. Tiene como símbolo ÉL ÁGUILA, ave REINA indiscutible de los cielos; ave rapaz más bella y majestuosa que representa la agudeza visual, capaz de avizorar más de tres kilómetros de distancia, tiene potencia en el vuelo y desafía los rigores atmosféricos.

El águila a través de los tiempos se la ha utilizado como emblema. Es un presagio de altura, calidad, poder y grandeza en las empresas de proyección. Por esto. Aristos, la seleccionó para forjar su excelencia educativa con la idea de liderar la educación de calidad en Calceta y sus áreas de influencia.

La institución Aristos fue creada en el año 2005, debido a la gran demanda estudiantil existente en la región. La idea de fundarla surgió en una reunión familiar, donde se trataba de las falencias de la educación en el Cantón Bolívar, de repente nació la idea de incrementar con el funcionamiento de una nueva institución educativa, el campo educacional de la localidad.

Se realizó una encuesta aproximadamente a 250 domicilios en lo que concierne a la educación de sus hijos, para satisfacción la colectividad requería

que se mejore la calidad de la formación para sus vástagos. Se realizó el proyecto y se presentó, el mismo que fue aprobado por la Dirección de Educación de Manabí, legalizando su funcionamiento con los Acuerdos #066-DP-DECM-2006 Y # 080-DP-DECM/2006

2.2 MISIÓN Y VISIÓN

2.2.1 MISIÓN

Educar, formar y capacitar a los /las estudiantes del Nivel Inicial y General Básico, desarrollando destrezas integradas en competencias para fortalecer la autonomía, el pensamiento crítico, la capacidad para resolver los problemas de la vida cotidiana y enfrentar retos para alcanzar la meta propuesta dentro de un ambiente de calidez, respeto, armonía y democracia.

2.2.2 VISIÓN

En los próximos cinco años, “ARISTOS-CALCETA” propenderá a la formación holística, orientada al ser, convivir y emprender, a la consecución de infraestructura adecuada, funcional con tecnología de punta en servicios educativos y a la creación del bachillerato internacional afianzando la formación científica para liderar como unidad educativa en el Cantón Bolívar con una educación de calidad (Consejo Directivo).

2.3 FILOSOFÍA

a. DE ÍNDOLE SOCIAL

Equidad de Género, Interculturalidad y Desarrollo Personal: A lo largo del proceso educativo, y teniendo en cuenta la edad y maduración de los alumnos nos proponemos orientar y acompañar a los niños y jóvenes en el

descubrimiento y realización de un proyecto de vida realista y orientado hacia los demás. Porque el descubrimiento de la propia llamada, la opción libre y generosa por la misma y el proyecto del propio futuro, constituyen la meta de todo el proceso de maduración humana integral.

b. DE ÍNDOLE PEDAGÓGICO

Los postulados de la UNESCO:

- Aprender a conocer
- Aprender a ser
- Aprender a hacer
- Aprender a convivir
- Aprender a emprender

Nuestro modelo de educación exige que Aristos-Calceta sea una auténtica Comunidad Educativa. A este fin, el conjunto de estamentos y personas que la forman se integran armónicamente poniendo en común ideales, objetivos y realizaciones en el Marco de un Proyecto educativo.

c. DE ÍNDOLE HUMANO

Directivos, docentes, personal administrativo, de servicios, se preparan en el Modelo Educativo.

Forman parte de la comunidad educativa Aristos-Calceta: Los directivos que forman parte de la entidad titular, los profesores, los alumnos. Los padres de familia, el personal no docente y padrinos becarios

2.4 UBICACIÓN GEOGRÁFICA

La escuela Particular de Educación General Básica Aristos-Calceta, funciona en la ciudad de Calceta, Cantón Bolívar Provincia de Manabí. La ciudad de

Calceta tiene una geografía irregular por la presencia de colinas y cerros como son: Mil Pesos, Las Montañas de Camote, Bejuco, Membrillo y Quiroga. Esta ciudad está atravesada por el Río Carrizal con sus afluentes: Bejuco, Pechichal y Bejuco.

En educación básica de la zona rural de la ciudad de Calceta hay solo dos unidades educativas completas: Mercedes y Pablo IV. Existen colegios fiscales desde octavo años hasta el bachillerato, como son: El 13 de Octubre, Monserrate Álava de González y los colegios Particulares: Homero José Ardila, Wenceslao Rijavec.

Hay dos jardines de infantes independientes fiscales: Lídice y Amado Quiroz. Funcionan ocho escuelas fiscales de primero a Séptimo año de E.G.B. y cuatro escuelas particulares de Primero a Séptimo año de E.G.B.

2.5 GENERALIDADES DE INGENIERÍA DE SOFTWARE

Según la definición del IEEE, citada por (Lewis, 1994) "software es la suma total de los programas de computadora, procedimientos, reglas, la documentación asociada y los datos que pertenecen a un sistema de cómputo". Según el mismo autor, "un producto de software es un producto diseñado para un usuario". En este contexto, la Ingeniería de Software es un enfoque sistemático del desarrollo, operación, mantenimiento y retiro del software", que en palabras más llanas, se considera que "la Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas a los problemas de desarrollo de software", es decir, "permite elaborar consistentemente productos correctos, utilizables y costo-efectivos" (Cota, 1994).

Desde otro punto de vista (Jacobson, 1998) argumenta que el proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de

un producto de software de calidad. "El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo". Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo" (Jacobson, 1998). Por otro lado (Zelkowitz, 1978) concreta que es el estudio de los principios y metodologías para desarrollo y mantenimiento de sistemas de software.

Una vez analizados cada uno de los conceptos de ingeniería de software descritos por los autores anteriormente mencionados, concluimos que la Ingeniería de Software es una disciplina que integra el proceso, los métodos, y las herramientas para el desarrollo de software de computadora (Los autores).

2.5.1 TIPOS DE SOFTWARE

El término software (partes suaves o blandas en castellano) hace alusión a la sumatoria de aquellas reglas, programas, datos, documentación e instrucciones que permiten la ejecución de múltiples tareas en un ordenador. Es su parte lógica e intangible y actúa como nexo entre el usuario y el hardware (partes duras), es decir, la parte tangible de la computadora (Alegre, M. 2010).

2.5.1.1 SOFTWARE DE SISTEMAS

Es un conjunto de programas cuyo objeto es facilitar el uso del computador (aísla de la complejidad de cada dispositivo, y presenta al exterior un modelo común de sistema de manejo para todos los dispositivos) y conseguir que se use eficientemente (ejemplo: realizar operaciones mientras se ejecuta un programa). Administra y asigna los recursos del sistema (hard-ware).

Está formado por todos aquellos programas cuya finalidad es servir al desarrollo o al funcionamiento de otros programas. Estos programas son muy variados: editores, compiladores, sistemas operativos, entornos gráficos, programas de telecomunicaciones, etc. pero se caracterizan por estar muy próximos al hardware, por ser utilizados concurrentemente por numerosos usuarios y por tratarse de programas de amplia difusión, no estando diseñados normalmente a medida. Esto permite un mayor esfuerzo en su diseño y optimización, pero también les obliga a ser muy fiables, cumpliendo estrictamente las especificaciones para las que fueron creados. Un ejemplo de este tipo de software son los sistemas operativos, como Windows y Unix (Laudon, et al. 2004).

2.5.1.2 SOFTWARE DE APLICACIÓN

Son los programas que controlan y optimización la operación de la máquina, establecen una relación básica y fundamental entre el usuario y el computador, hacen que el usuario pueda usar en forma cómoda y amigable complejos sistemas hardware, realizan funciones que para el usuario serían engorrosas o incluso imposibles, y actúan como intermediario entre el usuario y el hardware.

Son escritos / modificados por programadores de sistemas que han de tener profundos conocimientos acerca de cómo trabaja realmente la máquina. Se proporciona normalmente como parte integrante de la máquina por el proveedor (Escala, E 2008).

2.5.1.3 SOFTWARE DE GESTIÓN

El procesamiento de información de gestión constituye, casi desde los inicios de la informática la mayor de las áreas de aplicación de los ordenadores. Estos programas utilizan grandes cantidades de información almacenadas en bases

de datos con objeto de facilitar las transacciones comerciales o la toma de decisiones. Además de las tareas convencionales de procesamiento de datos, en las que el tiempo de procesamiento no es crítico y los errores pueden ser corregidos a posteriori, incluyen programas interactivos que sirven de soporte a transacciones comerciales (Flores, P 2008).

2.5.1.4 SOFTWARE DE PROGRAMACIÓN

Son aquellas herramientas que un programador utiliza para poder desarrollar programas informáticos. Para esto, el programador se vale de distintos lenguajes de programación. Como ejemplo se pueden tomar compiladores, programas de diseño asistido por computador, paquetes integrados, editores de texto, enlazadores, depuradores, intérpretes, entre otros (Vergara, K 2007).

2.5.2 MODELOS DE DESARROLLO DE SOFTWARE

Para el desarrollo de cualquier producto de software se realizan una serie de tareas entre la idea inicial y el producto final, un modelo de desarrollo establece el orden en el que se harán las cosas en el proyecto, provee de requisitos de entrada y de salida para cada una de las actividades, por ello es necesario el modelo de desarrollo.

Dado que cada proyecto es único, no existe un modelo que se aplique al 100% a todos los proyectos de una organización. Una organización puede contar con uno o más modelos de desarrollo para ser utilizados dependiendo del tipo de proyecto (Pérez, A 2008).

Con el concepto general que estipula Ángel Pérez Cortés sobre los Modelos de Desarrollo de Software, los autores deducen que es imprescindible la utilización de un modelo al momento de desarrollar un software, debido a que nos permitió tener un proceso lineal y organizado durante todo el proceso de elaboración (Los Autores).

La ingeniería de software tiene varios modelos, paradigmas o filosofías de desarrollo en los cuales se puede apoyar para la realización de software, de los cuales podemos destacar a éstos por ser los más utilizados y los más completos:

- Modelo en espiral (modelo evolutivo)
- Modelo de prototipos
- Modelo por etapas
- Modelo Incremental
- RUP (Modelo Racional)

2.5.2.1 RUP (MODELO RACIONAL)

El Proceso Unificado Racional, Rational Unified Process en inglés, y sus siglas RUP, es un proceso de desarrollo de software y junto con el Lenguaje Unificado de **Modelado UML**, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. El RUP no es un sistema con pasos firmemente establecidos, sino que trata de un conjunto de metodologías adaptables al contexto y necesidades de cada organización, donde el software es organizado como una colección de unidades atómicas llamados objetos, constituidos por datos y funciones, que interactúan entre sí. RUP es un proceso para el desarrollo de un proyecto de un software que define claramente quien, cómo, cuándo y qué debe hacerse en el proyecto (Kruchten, P 2002).

FASES DE DESARROLLO DEL RUP

- a) Inicio
- b) Elaboración
- c) Construcción
- d) Transición

a) FASE DE INICIO

Se hace un plan de fases, donde se identifican los principales casos de uso y se identifican los riesgos. Se concreta la idea, la visión del producto, como se enmarca en el negocio, el alcance del proyecto. El objetivo en esta etapa es determinar la visión del proyecto.

• MODELADO DEL NEGOCIO

En esta fase el equipo se familiarizará más al funcionamiento de la empresa, sobre conocer sus procesos.

- Entender la estructura y la dinámica de la organización para la cual el sistema va ser desarrollado.
- Entender el problema actual en la organización objetivo e identificar potenciales mejoras.
- Asegurar que clientes, usuarios finales y desarrolladores tengan un entendimiento común de la organización objetivo.

• REQUISITOS

En esta línea los requisitos son el contrato que se debe cumplir, de modo que los usuarios finales tienen que comprender y aceptar los requisitos que especifiquemos.

- Establecer y mantener un acuerdo entre clientes y otros stakeholders sobre lo que el sistema podría hacer.
- Proveer a los desarrolladores un mejor entendimiento de los requisitos del sistema.
- Definir el ámbito del sistema.
- Proveer una base para estimar costos y tiempo de desarrollo del sistema.

- Definir una interfaz de usuarios para el sistema, enfocada a las necesidades y metas del usuario.

b) FASE DE ELABORACIÓN

Se realiza el plan de proyecto, donde se completan los casos de uso y se mitigan los riesgos.

Planificar las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura. En esta etapa el objetivo es determinar la arquitectura Óptima.

• ANÁLISIS Y DISEÑO

En esta actividad se especifican los requerimientos y se describen sobre cómo se van a implementar en el sistema.

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

c) FASE DE CONSTRUCCIÓN

Se basa en la elaboración de un producto totalmente operativo y en la elaboración del manual de usuario. Construir el producto, la arquitectura y los planes, hasta que el producto está listo para ser enviado a la comunidad de usuarios. En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.

- **IMPLEMENTACIÓN**

Se implementan las clases y objetos en ficheros fuente, binarios, ejecutables y demás. El resultado final es un sistema ejecutable.

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

- **PRUEBAS**

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

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

d) FASE DE TRANSICIÓN

El objetivo es llegar a obtener el reléase del proyecto. Se realiza la instalación del producto en el cliente y se procede al entrenamiento de los usuarios. Realizar la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto, hasta que el cliente quede satisfecho, por tanto en esta fase suelen ocurrir cambios.

- **DESPLIEGUE**

Esta actividad tiene como objetivo producir con éxito distribuciones del producto y distribuirlo a los usuarios. Las actividades implicadas incluyen:

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

Cada una de estas fases es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

A medida que se avanza en el proyecto, es decir, cuando se va pasando de una fase a otra, la importancia relativa de cada uno de los Flujos de Trabajo va cambiando. Así, en las iteraciones de la Fase de Inicio el trabajo se centra principalmente en el Modelamiento del Negocio y en la captura y especificación de requisitos. Pero en la fase de Construcción el desarrollo está enfocado en la Implementación (codificación) y, en menor medida, en el Diseño (Sánchez, S 2003).

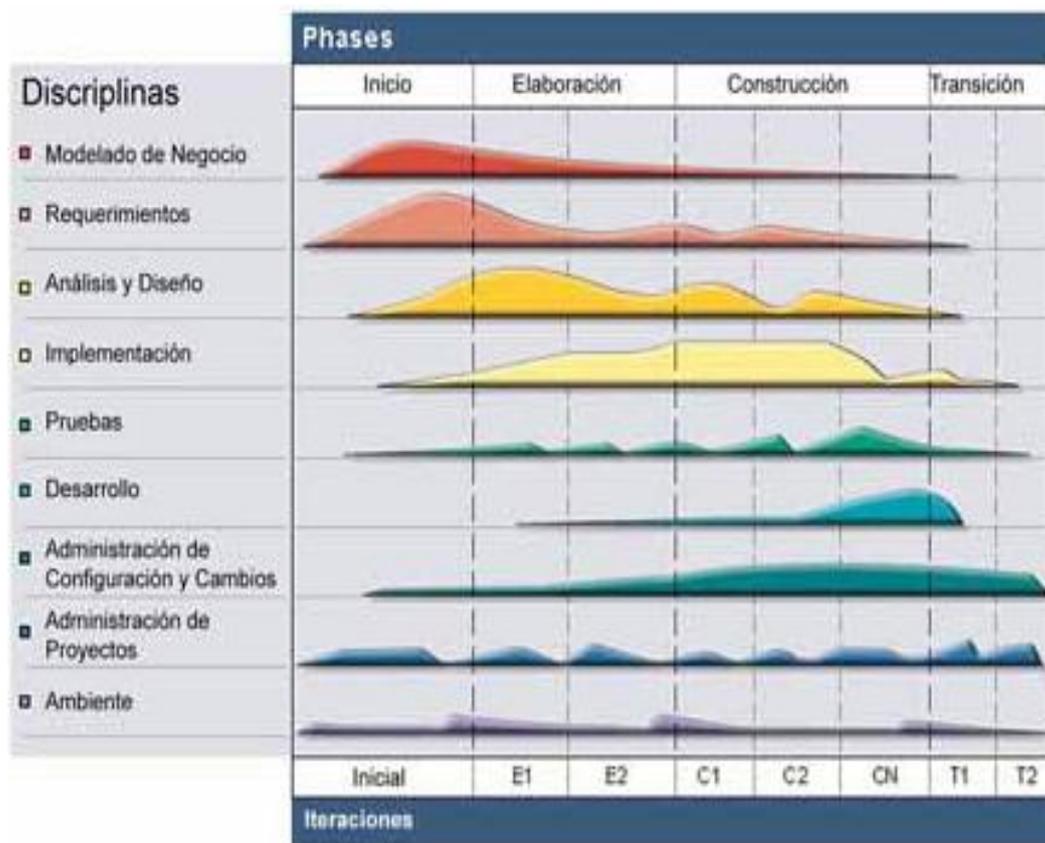


Figura 02.01: Fases de desarrollo del modelo RUP

Fuente: (Mendoza, M. 2004)

2.5.3 MODELADO UML

Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Es importante resaltar que UML es un "lenguaje de modelado" para especificar o para describir métodos o procesos. Se utiliza para definir un sistema, para detallar los artefactos en el sistema y para documentar y construir. En otras palabras, es el lenguaje en el que está descrito el modelo. Se puede aplicar en el desarrollo de software gran variedad de formas para dar soporte a una metodología de desarrollo de software (tal como el Proceso Unificado Racional o RUP), pero no especifica en sí mismo qué metodología o proceso usar.

UML no puede compararse con la programación estructurada, pues UML significa Lenguaje Unificado de Modelado, no es programación, solo se diagrama la realidad de una utilización en un requerimiento. Mientras que, programación estructurada, es una forma de programar como lo es la orientación a objetos, sin embargo, la programación orientada a objetos viene siendo un complemento perfecto de UML, pero no por eso se toma UML sólo para lenguajes orientados a objetos (Gracia, J 2010).

2.5.3.1 TIPOS DE DIAGRAMAS UML

○ DIAGRAMA DE CLASES

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro (Zapata, M 2006).

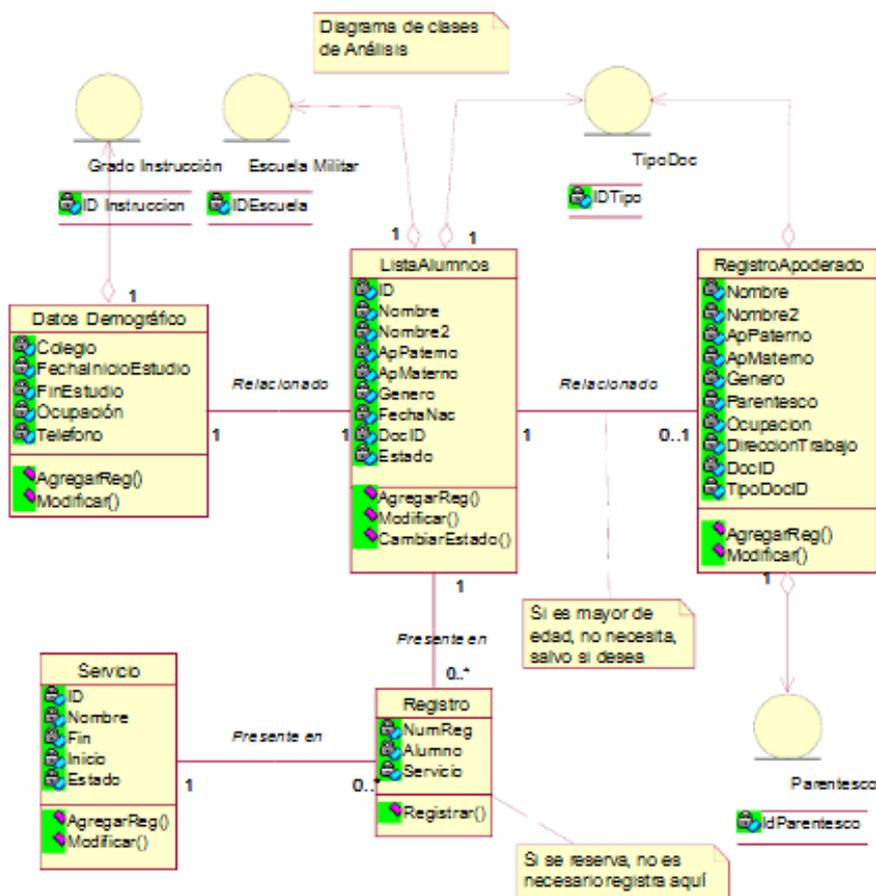


Figura 02.02: Ejemplo de diagrama de clases.

Fuente: (Mesa, E. 2010).

○ DIAGRAMA DE OBJETOS

Los diagramas de objetos son utilizados durante el proceso de Análisis y Diseño de los sistemas informáticos en la metodología UML.

Se puede considerar un caso especial de un diagrama de clases en el que se muestran instancias específicas de clases (objetos) en un momento particular del sistema. Los diagramas de objetos utilizan un subconjunto de los elementos de un diagrama de clase. Los diagramas de objetos no muestran la multiplicidad ni los roles, aunque su notación es similar a los diagramas de clase (Rumbaugh, J 2007).

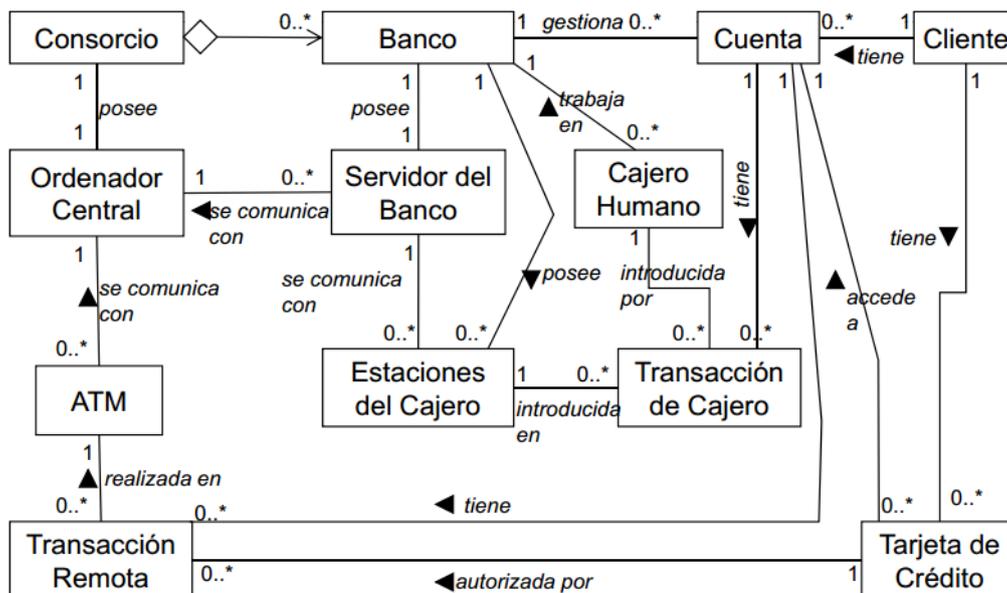


Figura 02.03: Ejemplo de diagrama de objeto

Fuente: (Riesco, D s.f).

○ DIAGRAMA DE CASOS DE USO

Los Casos de Uso no son parte del diseño, sino parte del análisis. De forma que al ser parte del análisis nos ayudan a describir qué es lo que el sistema debe hacer. Los Casos de Uso son qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario.

Los diagramas de casos de uso describen las relaciones y las dependencias entre un grupo de casos de uso y los actores participantes en el proceso.

Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema, y con el cliente, y resultan especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen qué es lo que debe hacer el sistema, pero no cómo (Campderrich, B 2003).

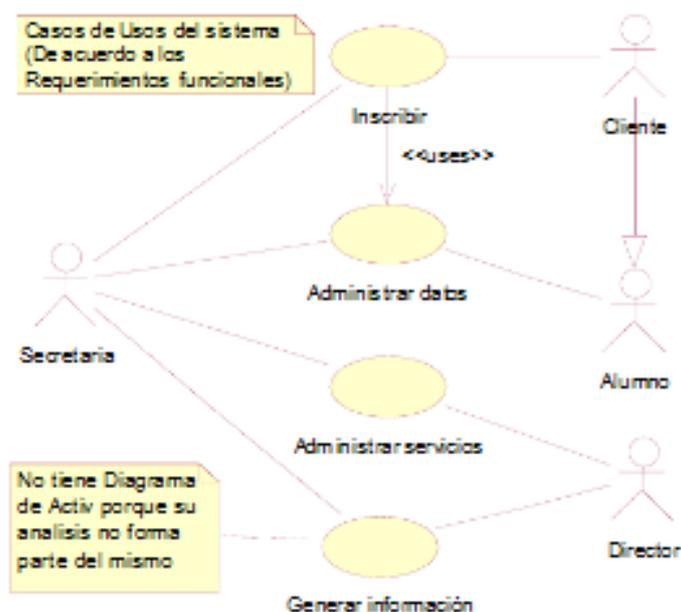


Figura 02.04: Ejemplo de diagrama de caso de uso

Fuente: (Landaca, K. 2010)

ELEMENTOS BÁSICOS

○ ACTORES

Los actores representan un tipo de usuario del sistema. Se entiende como usuario cualquier cosa externa que interactúa con el sistema. No tiene por qué ser un ser humano, puede ser otro sistema informático o unidades organizativas o empresas. Siempre hay que intentar independizar los actores de la forma en que se interactúa con el sistema. Por ejemplo un teclado no es un actor en la mayor parte de los casos, sólo un medio para introducir información al sistema. Suele ser útil mantener una lista de los usuarios reales para cada actor. Un actor en un diagrama de casos de uso representa un rol que alguien puede estar jugando, no un individuo particular por lo tanto puede haber personas particulares que puedan estar usando el sistema de formas diferentes en diferentes ocasiones: socio de biblioteca y bibliotecario.

- **CASO DE USO**

Es una tarea que debe poder llevarse a cabo con el apoyo del sistema que se está desarrollando. Se representan mediante un óvulo. Cada caso de uso debe detallarse, habitualmente mediante una descripción textual.

- **ASOCIACIONES**

Hay una asociación entre un actor y un caso de uso si el actor interactúa con el sistema para llevar a cabo el caso de uso. Un caso de uso debe especificar un comportamiento deseado, pero no imponer cómo se llevará a cabo ese comportamiento, es decir, debe decir QUÉ pero no CÓMO. Esto se realiza utilizando escenarios.

- **UN ESCENARIO**

Es una interacción entre el sistema y los actores, que puede ser descrito mediante una secuencia de mensajes. Un caso de uso es una generalización de un escenario (Cáceres, T 2010).

2.5.4 PRUEBAS DE SOFTWARE

Las Pruebas de Software, o "Testing" es una investigación empírica y técnica cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto bajo pruebas a la parte interesada o Stakeholder. Las Pruebas de Software son una actividad más en el proceso de "Aseguramiento de la Calidad"

Las Pruebas son básicamente un conjunto de actividades dentro del desarrollo de software. Dependiendo del tipo de pruebas, estas actividades podrán ser implementadas en cualquier momento de dicho proceso de desarrollo (Abad, J 2005).

2.5.4.1 TIPOS DE PRUEBAS DE SOFTWARE

a) PRUEBA DE INTEGRACIÓN

OBJETIVO DE LA PRUEBA

- Identificar errores introducidos por la combinación de programas probados unitariamente.
- Determina cómo la base de datos de prueba será cargada.
- Verificar que las interfaces entre las entidades externas (usuarios) y las aplicaciones funcionan correctamente.
- Verificar que las especificaciones de diseño sean alcanzadas.
- Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente.

DESCRIPCIÓN DE LA PRUEBA

- Describe cómo verificar que las interfaces entre las componentes de software funcionan correctamente.
- Determina cómo la base de datos de prueba será cargada.
- Determina el enfoque para avanzar desde un nivel de integración de las componentes al siguiente.
- Decide qué acciones tomar cuando se descubren problemas.
- Por cada Caso de Prueba ejecutado: Compara el resultado esperado con el resultado obtenido.

TÉCNICA

- Utilizar la técnica top-down. Se empieza con los módulos de nivel superior, y se verifica que los módulos de nivel superior llaman a los de nivel inferior de manera correcta, con los parámetros correctos.

- Utilizar la técnica down-top. Se empieza con los módulos de nivel inferior, y se verifica que los módulos de nivel inferior llaman a los de nivel superior de manera correcta, con los parámetros correctos.

CRITERIO DE COMPLETITUD

- Todas las pruebas planeadas han sido ejecutadas.
- Todos los defectos que se identificaron han sido tenidos en cuenta (Martínez, L 2005).

b) PRUEBAS DE CARGA

OBJETIVO DE LA PRUEBA

- Verificar el tiempo de respuesta del sistema para transacciones o casos de uso de negocios, bajo diferentes condiciones de carga.

DESCRIPCIÓN DE LA PRUEBA

- Las pruebas de carga miden la capacidad del sistema para continuar funcionando apropiadamente bajo diferentes condiciones de carga.
- La meta de las pruebas de carga es determinar y asegurar que el sistema funciona apropiadamente aún más allá de la carga de trabajo máxima esperada. Adicionalmente, las pruebas de carga evalúan las características de desempeño (tiempos de respuesta, tasas de transacciones y otros aspectos sensibles al tiempo).

TÉCNICA

- Use los scripts desarrolladas para Pruebas del Negocio.

- Modifique archivos de datos (para incrementar el número de transacciones o veces que cada transacción ocurre).

CRITERIO DE COMPLETITUD

- Múltiples transacciones, múltiples usuarios. Se completaron las pruebas de los scripts sin ninguna falla y dentro del tiempo esperado.

CONSIDERACIONES ESPECIALES

- Las pruebas de carga deben ser ejecutadas en una máquina dedicada o en un tiempo dedicado. Esto permite control total y medidas precisas.
- La Base de datos utilizada para pruebas de desempeño debe ser de un tamaño real o proporcionalmente más grande que la diseñada (Barranco, J sf).

c) PRUEBAS DE STRESS

OBJETIVO DE LA PRUEBA:

Verificar que el sistema funciona apropiadamente y sin errores, bajo estas condiciones de stress:

- Memoria baja o no disponible en el servidor.
- Máximo número de clientes conectados o simulados (actuales o físicamente posibles)
- Múltiples usuarios desempeñando la misma transacción con los mismos datos.
- El peor caso de volumen de transacciones (ver pruebas de desempeño).

NOTAS: La meta de las pruebas de stress también es identificar y documentar las condiciones bajo las cuales el sistema FALLA.

DESCRIPCIÓN DE LA PRUEBA:

Las pruebas de stress se proponen encontrar errores debidos a recursos bajos o completitud de recursos. Poca memoria o espacio en disco puede revelar defectos en el sistema que no son aparentes bajo condiciones normales. Otros defectos pueden resultar de incluir recursos compartidos, como bloqueos de base de datos o ancho de banda de la red. Las pruebas de stress identifican la carga máxima que el sistema puede manejar.

El objetivo de esta prueba es investigar el comportamiento del sistema bajo condiciones que sobrecargan sus recursos. No debe confundirse con las pruebas de volumen: un esfuerzo grande es un pico de volumen de datos que se presenta en un corto período de tiempo.

Puesto que la prueba de esfuerzo involucra un elemento de tiempo, no resulta aplicable a muchos programas, por ejemplo a un compilador o a una rutina de pagos.

Es aplicable, sin embargo, a programas que trabajan bajo cargas variables, interactivas, de tiempo real y de control de proceso.

Aunque muchas pruebas de esfuerzo representan condiciones que el programa encontrará realmente durante su utilización, muchas otras serán en verdad situaciones que nunca ocurrirán en la realidad. Esto no implica, sin embargo, que estas pruebas no sean útiles.

Si se detectan errores durante estas condiciones “imposibles”, la prueba es valiosa porque es de esperar que los mismos errores puedan presentarse en situaciones reales, algo menos exigentes.

TÉCNICA:

- Use los scripts utilizados en las pruebas de desempeño.
- Para probar recursos limitados, las pruebas se deben correr en un servidor con configuración reducida (o limitada).
- Para las pruebas de stress restantes, deben utilizarse múltiples clientes, ya sea corriendo los mismos scripts o scripts complementarios para producir el peor caso de volumen de transacciones.

CRITERIO DE COMPLETITUD:

Todas las pruebas planeadas han sido ejecutadas y excedidas sin que el sistema falle.

(O si las condiciones en que el sistema falle ocurren por fuera de las condiciones especificadas).

CONSIDERACIONES ESPECIALES

Producir stress en la red puede requerir herramientas de red para sobrecargarla de tráfico.

El espacio en disco utilizado para el sistema debe ser reducido temporalmente para limitar el espacio disponible para el crecimiento de la Base de datos.

Sincronización de varios clientes accediendo simultáneamente los mismos registros (Lemus, G 2012).

2.6 TIPOS DE PROGRAMACIÓN

Entre los tipos de programación más importante que existen tenemos:

- Programación estructurada
- Programación orientada a objetos
- Programación en capas.

2.6.1 PROGRAMACIÓN POR CAPAS

La programación por capas es un estilo de programación en la que el objetivo primordial es la separación de la lógica de negocios de la lógica de diseño, un ejemplo básico de esto es separar la capa de datos de la capa de presentación al usuario.

2.6.1.1 CARACTERÍSTICAS DE LA PROGRAMACIÓN EN CAPAS

La programación por capas es una técnica de ingeniería de software propia de la programación por objetos, éstos se organizan principalmente en 3 capas: la capa de presentación o frontera, la capa de lógica de negocio o control, y la capa de datos. Siguiendo el modelo, el desarrollador se asegura avanzar en la programación del proyecto de una forma ordenada, lo cual beneficia en cuanto a reducción de costos por tiempo, debido a que se podrá avanzar de manera más segura en el desarrollo, al ser dividida la aplicación general en varios módulos y capas que pueden ser tratados de manera independiente y hasta en forma paralela.

Por otra parte, otra característica importante de recalcar es la facilidad para las actualizaciones de la aplicación. En este aspecto, la programación en capas juega un papel de suma importancia ya que sigue un estándar conocido en el ambiente de desarrollo de aplicaciones, lo cual da al programador una guía para hacer mejoras a la aplicación sin que esto sea una tarea tediosa y desgastante, siguiendo el estándar establecido para tal fin y dividiendo las tareas en partes específicas para cada capa del proyecto (Calle, F 2008).

Las principales capas que siempre deben estar en este modelo son:

- **CAPA DE PRESENTACIÓN O FRONTERA**

La presentación del programa ante el usuario, debe manejar interfaces que cumplan con el objetivo principal de este componente, el cual es facilitar al usuario la interacción con la aplicación. Para esto se utilizan patrones predefinidos para cada tipo de aplicación y para cada necesidad del usuario. La interfaz debe ser amigable y fácil de utilizar, ya que el usuario final es el que se va a encargar de utilizar el sistema y de dar retroalimentación al equipo de desarrollo en caso de que haya algo que mejorar.

Las interfaces deben ser consistentes con la información que se requiere, no se deben utilizar más campos de los necesarios, así como la información requerida tiene que ser especificada de manera clara y concisa, no debe haber más que lo necesario en cada formulario y por último, las interfaces deben satisfacerlos requerimientos del usuario, por lo cual no se debe excluir información solicitada por el usuario final y no se debe incluir información no solicitada por el mismo.

Dentro de la parte técnica, la capa de presentación contiene los objetos encargados de comunicar al usuario con el sistema mediante el intercambio de información, capturando y desplegando los datos necesarios para realizar alguna tarea. En esta capa los datos se procesan de manera superficial por ejemplo, para determinar la validez de su formato o para darles algún orden específico. Esta capa se comunica únicamente con la capa de Reglas de Negocio o Control (Mayorga, D 2004).

- **CAPA DE LÓGICA DE NEGOCIO O CONTROL:**

Es llamada capa de reglas de negocio porque en esta se definen todas las reglas que se deben cumplir para una correcta ejecución del programa.

Es aquí donde se encuentra toda la lógica del programa, así como las estructuras de datos y objetos encargados para la manipulación de los datos existentes, así como el procesamiento de la información ingresada o solicitada por el usuario en la capa de presentación.

Representa el corazón de la aplicación ya que se comunica con todas las demás capas para poder llevar a cabo las tareas. Por ejemplo, mediante la capa de presentación obtiene la información ingresada por el usuario, y despliega los resultados. Si la aplicación se comunica con otros sistemas que actúan en conjunto, lo hace mediante esta capa. También se comunica con la capa de datos para obtener información existente o ingresar nuevos datos. Recibe los datos que ingresó el usuario del sistema mediante la capa de presentación, luego los procesa y crea objetos según lo que se necesite hacer con estos datos; esta acción se denomina encapsulamiento.

Al encapsular los datos, el programa asegura mantener la consistencia de los mismos, así como obtener información precisa de las bases de datos e ingresar en las mismas, solamente la información necesaria, asegurando así no tener datos duplicados ni en las bases de datos, ni en los reportes solicitados por el usuario (Martín, J s.f).

- **CAPA DE DATOS:**

Es la encargada de realizar transacciones con bases de datos y con otros sistemas para obtener o ingresar información al sistema. El manejo de los datos debe realizarse de forma tal que haya consistencia en los mismos, de tal forma los datos que se ingresan así como los que se extraen de las bases de datos, deben ser consistentes y precisos. Es en esta capa donde se definen las consultas a realizar en la base de datos, tanto las consultas simples como las consultas complejas para la generación de reportes más específicos. Esta capa envía la información directamente a la capa de reglas de negocio para que sea

procesada e ingresada en objetos según se necesite, esta acción se denomina encapsulamiento (Universidad Sevilla, sf).

2.6.1.2 VENTAJAS

Al implementar este modelo de programación, se asegura un trabajo de forma ordenada y separada, debido a que sigue el principio de “divide y vencerás”. Cada capa está dividida según su funcionalidad cuando se quiere modificar el sistema basta con cambiar un objeto o conjunto de objetos de una capa. Esto se llama modularidad. (Tomás, M 2007).

2.7 ASPECTOS RELEVANTES DE UNA BASE DE DATOS

Es una colección de datos organizados y estructurados según un determinado modelo de información que refleja no solo los datos en sí mismo, sino también las relaciones que existen entre ellos. La definición y descripción de los datos han de ser únicas para minimizar la redundancia y maximizar la independencia en su utilización. En una base de datos, las entidades y atributos del mundo real, se convierten en registros y campos. Estas entidades pueden ser tanto objetos materiales como libros o fotografías, pero también personas e, incluso, conceptos e ideas abstractas. Las entidades poseen atributos y mantienen relaciones entre ellas (Pérez, D 2011).

En base a lo estipulado por Pérez los autores concluyen que las bases de datos se han convertido en un elemento indispensable no sólo para el funcionamiento de los grandes motores de búsqueda y la recuperación de información a lo largo y ancho de la Web, sino también para la creación de sistemas en los que se precisa manejar grandes o pequeños volúmenes de datos. La creación de una base de datos es un factor importante en la realización del sistema para la Unidad Educativa Aristos ya que en ésta se puedan hacer consultas y acceder a la información que interese, pues es una herramienta imprescindible de cualquier sistema informático (Los Autores).

2.7.1 MODELOS DE BASE DE DATOS

Un modelo de datos para las bases de datos es una colección de conceptos que se emplean para describir la estructura de una base de datos. Esa colección de conceptos incluye entidades, atributos y relaciones. La mayoría de los modelos de datos poseen un conjunto de operaciones básicas para especificar consultas y actualizaciones de la base de datos.

Entre los diferentes modelos de base de datos se encuentran:

- a) Base de datos relacional
- b) Base de datos de fichero
- c) Base de datos orientada a objeto
- d) Base de datos jerárquicas.
- e) Base de datos de red

a) BASE DE DATOS RELACIONAL

Las bases de datos relacionales son las más populares actualmente. Su nombre proviene de su gran ventaja sobre las bases de datos de fichero plano: la posibilidad de relacionar varias tablas de datos entre sí, compartiendo información y evitando la duplicidad y los problemas que ello conlleva (espacio de almacenamiento y redundancia). (Stair, R.s.f.)

Gracias al aporte de Stair que nos detalla los diferentes modelos de base de datos, hemos llegado a la determinante que la Base de Datos Relacional nos permite una mayor eficacia, flexibilidad y confianza en el tratamiento de los datos. En este modelo relacional se representa el mundo real mediante tablas relacionadas entre sí por columnas comunes. Las bases de datos que pertenecen a esta categoría se basan en el modelo relaciones, cuya estructura

principal es la relación, es decir una tabla bidimensional compuesta por líneas y columnas (Los Autores).

2.7.2 SISTEMA DE GESTIÓN DE BASE DE DATOS (SGBD)

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

El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos (González, J 2009).

Existen distintos objetivos que deben cumplir los SGBD:

2.7.2.1 ABSTRACCIÓN DE LA INFORMACIÓN

Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Da lo mismo si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.

2.7.2.2 INDEPENDENCIA

La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.

2.7.2.3 REDUNDANCIA MÍNIMA

Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.

2.7.2.4 CONSISTENCIA

En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.

2.7.2.5 SEGURIDAD

La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra asegurada frente a usuarios malintencionados, que intenten leer información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.

2.7.2.6 INTEGRIDAD

Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.

2.7.2.7 RESPALDO Y RECUPERACIÓN

Los SGBD deben proporcionar una forma eficiente de realizar copias de seguridad de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.

2.7.2.8 CONTROL DE LA CONCURRENCIA

En la mayoría de entornos (excepto quizás el doméstico), lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

2.7.2.9 TIEMPO DE RESPUESTA

Lógicamente, es deseable minimizar el tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados (Mingo, M 2009).

2.8 MODELO ENTIDAD-RELACIÓN

Los diagramas o modelos entidad-relación (denominado por su siglas, ERD “DiagramEntityrelationship”) son una herramienta para el modelado de datos de un sistema de información. Estos modelos expresan entidades relevantes para un sistema de información, sus inter-relaciones y propiedades.

El modelo de datos entidad-relación está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos objetos.

2.8.1 ENTIDAD

Representa una "cosa" u "objeto" del mundo real con existencia independiente, es decir, se diferencia unívocamente de otro objeto o cosa, incluso siendo del mismo tipo, o una misma entidad.

Una entidad puede ser un objeto con existencia física como: una persona, un animal, una casa, etc. (entidad concreta); o un objeto con existencia conceptual como: un puesto de trabajo, una asignatura de clases, un nombre, etc. (entidad abstracta).

Una entidad está descrita y se representa por sus características o atributos. Por ejemplo, la entidad Persona las características: Nombre, Apellido, Género, Estatura, Peso, Fecha de nacimiento, etc...

2.8.2 ATRIBUTOS

Los atributos son las características que definen o identifican a una entidad. Estas pueden ser muchas, y el diseñador solo utiliza o implementa las que considere más relevantes. Los atributos son las propiedades que describen a cada entidad en un conjunto de entidades.

En un conjunto de entidades, cada entidad tiene valores específicos asignados para cada uno de sus atributos, de esta forma, es posible su identificación unívoca.

2.8.3 RELACIÓN

Describe cierta dependencia entre entidades o permite la asociación de las mismas o define la relación como una asociación de dos o más entidades. A cada relación se le asigna un nombre para poder distinguirla de las demás y

saber su función dentro del modelo entidad-relación. Otra característica es el grado de relación, siendo las de grado 1 relaciones que solo relacionan una entidad consigo misma. Las de grado 2 son relaciones que asocian dos entidades distintas, y las de grado n que se tratan de relaciones que unen más de dos entidades. (Cuartas, J 2011).

2.8.3.1 CARDINALIDAD DE LAS RELACIONES

El diseño de relaciones entre las tablas de una base de datos puede ser la siguiente:

- **Relaciones de uno a uno:** una instancia de la entidad A se relaciona con una y solamente una de la entidad B.
- **Relaciones de uno a muchos:** cada instancia de la entidad A se relaciona con varias instancias de la entidad B.
- **Relaciones de muchos a muchos:** cualquier instancia de la entidad A se relaciona con cualquier instancia de la entidad B.

2.9 HERRAMIENTAS DE DESARROLLO DE SOFTWARE

2.9.1 MICROSOFT SQL SERVER 2008 R2

Es un sistema de base de datos relacional que utiliza el lenguaje SQL para facilitar al usuario la ejecución de sentencias tanto de manipulación como de selección de datos. SQL SERVER 2008 R2 es un conjunto de objetos eficientemente almacenados. Los objetos donde se almacena la información se denominan tablas, y éstas a su vez están compuestas de filas y columnas. En el centro de SQL Server está el motor de SQL Server, el cual procesa los comandos de la base de datos. Los procesos se ejecutan dentro del sistema operativo y entienden únicamente de conexiones y de sentencias SQL.

SQL Server incluye herramientas para la administración de los recursos que el ordenador nos proporciona y los gestiona para un mejor rendimiento de la base de datos.

Microsoft con su software SQL Server 2008 nos ofrece una plataforma de gestión de datos muy óptima, al cual podemos acceder desde cualquier lugar y en cualquier momento. Con SQL Server 2008 se puede almacenar datos estructurados, semi-estructurados, no estructurados y documentos, tales como las imágenes y más; de forma directamente en el base de datos (Sánchez, D 2009).

2.9.1.1 CARACTERISTICAS GENERALES DE SQL SERVER 2008 R2

- Protección de información valiosa
- Administración extensible de clave
- Espejado mejorado de base de datos
- Recuperación automática de páginas de datos
- Compresión de la corriente de logs
- Respuesta predecible
- Rendimiento predecible de solicitudes
- Compresión de datos
- Administra por políticas
- Colección de datos de rendimiento
- Simplifica el desarrollo de aplicaciones
- Fecha/hora
- Id con jerarquía
- Datos filestream

- Columnas esparcidas
- Tipos definidos de grandes usuarios
- Tipos de datos espaciales
- Compresión de backup
- Paralelismo de tabla particionada
- Optimización de starjoinquery
- Configuración de agrupaciones
- Mejoras en el generador de reportes
- Mergesqlstatement
- Mejoras en sql server integration services (ssis)
- Lookups persistentes en sql server integration services (ssis)
- Escala de análisis y rendimiento
- Reescritura
- Infraestructura para la administración de reportes
- Integración de microsoft office (Torres, L 2008).

2.9.2 PLATAFORMA .NET

.NET es una plataforma para el desarrollo de aplicaciones que integran múltiples tecnologías que han ido apareciendo en los últimos años como ASP.NET, ADO.NET, LINQ, WPF, SILVERLIGHT, etc. Junto con el potente entorno integrado de desarrollo Visual Studio, que permite desarrollar múltiples tipos de aplicaciones (Ceballos et al. 2010)

2.9.2.1 VISUAL BASIC.NET 2010

Visual Basic 2010 ofrece soporte para entorno de ejecución dinámico.

VB 2010 forma parte de Microsoft Silverlight.

Visual Basic .NET (VB.NET) es un lenguaje de programación orientado a objetos que se puede considerar una evolución de Visual Basic implementada sobre el framework .NET. Su introducción resultó muy controvertida, ya que debido a cambios significativos en el lenguaje VB.NET no es compatible hacia atrás con Visual Basic, pero el manejo de las instrucciones es similar a versiones anteriores de Visual Basic, facilitando así el desarrollo de aplicaciones más avanzadas con herramientas modernas.

La gran mayoría de programadores de VB.NET utilizan el entorno de desarrollo integrado Microsoft Visual Studio en alguna de sus versiones (desde el primer Visual Studio .NET hasta Visual Studio .NET 2010, que es la última versión de Visual Studio para la plataforma .NET), aunque existen otras alternativas, como SharpDevelop.

Al igual que con todos los lenguajes de programación basados en .NET, los programas escritos en VB .NET requieren el Framework .NET o Mono para ejecutarse (Groussard, T. 2010).

- **CARACTERÍSTICAS**

Se describen las características nuevas y mejoradas de la versión de lanzamiento original de Visual Basic 2010.

- **PROPIEDADES AUTOIMPLEMENTADAS**

Las propiedades autoimplementadas proporcionan una sintaxis abreviada que permite especificar rápidamente una propiedad de una clase sin tener que escribir el código **Get** y **Set** para la propiedad.

- **INICIALIZADORES DE COLECCIÓN**

Los inicializadores de colección proporcionan una sintaxis abreviada que permite crear una colección y rellenarla con un conjunto inicial de valores. Los inicializadores de colección son útiles cuando se está creando una colección a partir de un conjunto de valores conocidos como, por ejemplo, una lista de opciones de menú o categorías.

- **CONTINUACIÓN DE LÍNEA IMPLÍCITA**

En muchos casos, la continuación de línea implícita permite continuar una instrucción en la línea consecutiva siguiente sin utilizar el carácter de subrayado (`_`). Para obtener una lista de todos los casos en los que se puede omitir un carácter de subrayado

- **EXPRESIONES LAMBDA DE MÚLTIPLES LÍNEAS Y SUBROUTINAS**

La compatibilidad con la expresión lambda se ha expandido para admitir las subrutinas además de las funciones de lambda de múltiples líneas y subrutinas.

- **NUEVA OPCIÓN DE LA LÍNEA DE COMANDOS PARA ESPECIFICAR UNA VERSIÓN DE LENGUAJE**

La opción `/langversion` de la línea de comandos hace que el compilador acepte únicamente la sintaxis que sea válida en la versión especificada de Visual Basic.

- **COMPATIBILIDAD CON LA EQUIVALENCIA DE TIPOS**

Ahora se puede implementar una aplicación que contiene información de tipos incrustada en lugar de información de tipos que se importa desde un ensamblado de interoperabilidad primario (PIA). Con la información de tipos incrustada, la aplicación puede utilizar los tipos en un motor en tiempo de ejecución sin necesidad de una referencia al ensamblado en tiempo de ejecución. Si se publican varias versiones del ensamblado del runtime, la aplicación que contiene la información de tipos incrustada puede funcionar con las diferentes versiones sin que sea necesario volver a compilarla.

- **COMPATIBILIDAD DINÁMICA**

Visual Basic enlaza a los objetos de los lenguajes dinámicos como IronPython e IronRuby.

- **COVARIANZA Y CONTRAVARIANZA**

La covarianza permite usar un tipo más derivado que el especificado por el parámetro genérico, mientras que la contravarianza permite utilizar un tipo menos derivado. Esto permite la conversión implícita de las clases que implementan interfaces variantes y proporciona mayor flexibilidad a la hora de hacer coincidir las firmas de método con tipos de delegado variantes. Se pueden crear interfaces y delegados variantes mediante las nuevas palabras clave `In` y `Out`. .NET Framework también incluye compatibilidad con la varianza para varios delegados e interfaces genéricos existentes, incluidos la interfaz `IEnumerable<T>` y los delegados `Action<T>` y `Func<TResult>`.

- **NAVEGAR A**

Se puede usar la característica **Navegar a** para buscar un símbolo o un archivo en código fuente. Puede buscar palabras clave incluidas en un símbolo concatenadas mediante notación Camel o caracteres de subrayado a fin de dividir dicho símbolo en palabras clave.

- **RESALTAR REFERENCIAS**

Al hacer clic en un símbolo en el código fuente, todas las instancias de ese símbolo se resaltan en el documento.

En muchas estructuras de control, al hacer clic en una palabra clave, se resaltan todas las palabras clave en la estructura. Por ejemplo, al hacer clic en **If** en una construcción **If...Then...Else**, se resaltan todas las instancias de **If**, **Then**, **Elseif**, **ElseyEndIf** de la construcción.

Para desplazarse al siguiente o anterior símbolo resaltado, puede usar **CTRL+MAYÚS+FLECHA ABAJO** o **CTRL+MAYÚS+FLECHA ARRIBA**.

- **GENERAR A PARTIR DEL USO**

La característica **Generar a partir del uso** permite usar clases y miembros antes de definirlos. Puede generar un código auxiliar para cualquier clase, constructor, método, propiedad, campo o enumeración que desee utilizar pero no ha definido todavía. Puede generar nuevos tipos y miembros sin salir de su ubicación actual en el código. De este modo, se minimizan las interrupciones en el flujo de trabajo.

La característica **Generar a partir del uso** admite estilos de programación como el desarrollo de pruebas en primer lugar

- **MODO DE SUGERENCIA DE INTELLISENSE**

IntelliSense proporciona ahora dos alternativas para completar las instrucciones de IntelliSense: el modo de finalización y el modo de sugerencia. El modo de sugerencia se utiliza cuando las clases y los miembros se usan antes de definirlos.

- **APLICACIONES DE EJEMPLO**

Visual Basic incluye nuevas aplicaciones de ejemplo que muestran las características siguientes: propiedades autoimplementadas, continuación de línea implícita, inicializadores de colección, covarianza y contravarianza y expresiones lambda de múltiples líneas y subrutinas (Charte, F. 2011).

CAPÍTULO III. DESARROLLO METODOLÓGICO

3.1 MÉTODO INFORMÁTICO

El método informático que permitió realizar ordenadamente el software para la Unidad Educativa Aristos es el RUP, el mismo contiene cuatro fases de desarrollo:

3.1.1 FASE DE INICIO

3.1.1.1 MODELADO DEL NEGOCIO

En esta fase se efectuó una serie de visitas en la Unidad Educativa Aristos de la ciudad de Calceta, que permitieron obtener conocimientos de los procesos internos realizados en esta dependencia. En entrevistas con el Sr. Jorge Zambrano (Director de la institución) y la Srta. Ana Dolores Santos (Secretaria de la institución) se adquirió información importante de la misma.

La Unidad Educativa Aristos es una institución cuyo objetivo principal es formar líderes para la patria, en la actualidad cuenta con un número de 110 estudiantes provenientes del cantón y de los sitios aledaños a este y con un total de 14 catedráticos. Posee los siguientes años de educación básica:

- **Pre básico:**
 - Inicial 1
 - Inicial 2

- **Nivel básico**
 - Primer año de educación básica.
 - Segundo año de educación básica

- Tercer año de educación básica
- Cuarto año de educación básica
- Quinto año de educación básica
- Sexto año de educación básica
- Séptimo año de educación básica

- **Nivel secundario**

- Octavo
- Noveno
- Décimo

Las asignaturas impartidas varían dependiendo del año de educación básica, excepto las consideradas optativas como son: cultura física, danza, música, deportes, idioma extranjero y computación, cada una de estas tienen asignado un docente específico y son impartidas en todos los niveles. Las materias existentes por año de educación básica se muestran en la siguiente figura:

AÑO DE EDUCACIÓN BÁSICA	ASIGNATURAS BÁSICAS	ASIGNATURAS OPTATIVAS
Pre básica	Lógica matemáticas, lengua, cuento, corporación motora, proyecto.	Computación, idioma extranjero, danza, música, cultura física,
Primer año de educación básica	Lógica matemáticas, educación sexual, entorno, psicomotricidad, cuento, lengua cultura estética,	Danza, deportes, computación, cultura física, música, idioma extranjero.
Segundo año de educación básica	Lógica matemáticas, educación sexual, entorno, cuento, lengua, opción practica cultura estética,	Danza, deportes, computación, cultura física, música, idioma extranjero
Tercer año de educación básica	Matemática, lengua y literatura, educación sexual, entorno natural y social, proyecto, opción práctica, cultura estética,	Danza, idioma extranjero, computación, música, deporte.

Cuarto año de educación básica	Lengua y literatura, matemática, ciencias naturales, opción práctica, estudios sociales, cultura estética.	Danza, idioma extranjero, computación, música, deporte, cultura física.
Quinto año de educación básica	Lengua y literatura, estudios sociales, educación sexual, ciencias naturales, matemáticas, opción práctica, cultura estética,	Computación, danza, cultura física, idioma extranjero, deporte, música.
Sexto año de educación básica	Lengua y literatura, cultura estética, matemática, estudios sociales, ciencias naturales, opción práctica, educación sexual,	Danza, computación, idioma extranjero, deporte, cultura física, música.
Séptimo año de educación básica	Lengua y literatura, ciencias naturales, estudios sociales, educación sexual, matemática, opciones prácticas.	Idioma extranjero, cultura física, danza, deporte, computación.
Octavo	Matemáticas, mecanografía, lengua y literatura, idioma extranjero, ciencias naturales, estudios sociales, tutoría, contabilidad, dibujo técnico, educación sexual.	Computación, danza, deporte, cultura física.
Noveno	Matemáticas, mecanografía, lengua y literatura, idioma extranjero, ciencias naturales, estudios sociales, tutoría, contabilidad, dibujo técnico, educación sexual.	Computación, danza, deporte, cultura física.
Decimo	Matemáticas, mecanografía, lengua y literatura, idioma extranjero, ciencias naturales, estudios sociales, tutoría, contabilidad, dibujo técnico, educación sexual.	Computación, danza, deporte, cultura física.

Tabla 03.01: Listado de las materias impartidas en la Unidad ARISTOS.

Fuente: Ana Dolores Santos Montesdeoca (Secretaria de la Unidad Educativa Aristos)

El departamento de secretaría se encargaba de realizar la distribución de las materias a los docentes con sus respectivos cursos, este proceso se efectuaba en hojas de Microsoft Excel 2010 en tablas previamente elaboradas con un

formato debidamente detallado y seguidamente archivado en folders utilizando la información que se muestra en la tabla 03.01, así mismo se efectuaban los procedimientos de registro de matriculación de calificaciones de los estudiantes. (Vea tabla 03.02 y tabla 03.03).



ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA

"ARISTOS - CALCETA"

Creada mediante Acuerdo Ministerial # 066 DP – DECM

REPORTE DE CALIFICACIONES

AÑO LECTIVO 2012-2013

Apellidos y Nombres:

#	MATERIAS	PRIMER QUIMESTRE				SEGUNDO QUIMESTRE				PUNTAJE TOTALES					
		PRIMERA UNIDAD		SEGUNDA UNIDAD		PRIMERA UNIDAD		SEGUNDA UNIDAD		Aprovechamiento			Disciplina		
		Disciplina	Aprovecha.	Disciplina	Aprovecha.	Disciplina	Aprovecha.	Disciplina	Aprovecha.	1 Q	2Q	PROMEDIO	1Q	2Q	PROMEDIO
1															
2															
3															
4															
5															
6															
7															
8															

OBSERVACIONES:.....

MS= MUY SATISFACTORIO

S= SATISFACTORIO

PS= POCO SATISFACTORIO

DIRECTOR PEDAGÓGICO

DIRIGENTE

SECRETARIA

Tabla 03.02: Registro de calificaciones de los estudiantes.

Fuente: Ana Dolores Santos Montesdeoca (Secretaria de la Unidad Educativa Aristos)



**ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA
"ARISTOS - CALCETA"**
Creada mediante Acuerdo Ministerial # 066 DP – DECM
REPORTE DE REGISTRO DE MATRICULACIÓN
AÑO LECTIVO 2012-2013

Nombres:.....

Apellidos:.....

Representante:.....

Fecha de nacimiento:.....

Colegio procedencia:.....

Dirección:.....

Provincia.....*cantón*.....*parroquia*.....

Año de educación Básica.....*Paralelo*.....*Nivel*.....

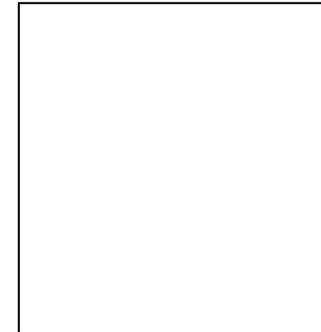


Tabla 03.03: Proceso de registro de matriculación.

Fuente: Ana Dolores Santos Montesdeoca (Secretaria de la Unidad Educativa Aristos)

Los datos obtenidos permitieron constatar que las técnicas utilizadas para ejecutar las actividades académicas, se los realizaban en forma manual, lo cual dificultaba y retrasaba las labores de la Srta. Ana Dolores Santos Montesdeoca, (secretaria de la institución). En esta área se comprobó que el método empleado para registrar y almacenar los datos de los estudiantes era útil, pero que se requería contar con una herramienta tecnológica que permita agilizar los procesos de registro de matriculación y de calificaciones. De esta manera se plasmaron los siguientes requisitos:

3.1.1.2 REQUISITOS

FUNCIONALES
Registrar los datos del estudiante, de los representantes y de los docentes.
Mantener los datos de los estudiantes actualizados
Efectuar las matrículas de los estudiantes en el vigente periodo lectivo y en el correcto grado o curso en su respectivo paralelo.
Ingresar y almacenar calificaciones de los estudiantes
Realizar consultas
Generación de los reportes correspondientes
El software no maneja los procesos de pago de matrículas
Ingreso, almacenamiento y consulta de los docentes con sus respectivas materias

Tabla 03.04: Requisitos funcionales del software

Elaboración: Los autores

NO FUNCIONALES
Confiabilidad: con el sistema la información que se obtendrá a través de consulta y reportes será 100% real y confiable, ya que la seguridad de los datos es uno de los puntos relevantes
Amigable: Las aplicaciones tienen un diseño sencillo, fácil de manejar, con opciones claras y ordenadas.
Seguridad: los datos solo podrán ser manipulados por la persona encargada del departamento de secretaria, mediante un password
Efectividad: se podrán obtener datos reales y confiables
La aplicación será desarrollada en la plataforma Microsoft Visual Studio.Net 2010
El diseño de la base de datos se realizara en Microsoft SQL Server 2008 R2

Tabla 03.05: Requisitos no funcionales del software

Elaboración: Los autores

IMPLEMENTACION
<p>La aplicación deberá funcionar sobre Microsoft</p> <ul style="list-style-type: none"> • Sistema operativo Windows XP o Superior • Memoria RAM 1 GB o superior • Disco Duro de 50 GB o superior • Procesador Pentium IV en adelante. • Resolución de pantalla de 800X600

Tabla 03.06: Requisitos de implementación del software

Elaboración: Los autores

3.1.2 FASE DE ELABORACIÓN

Con los requisitos obtenidos se elaboraron los diagramas UML casos de uso, se procedió a diseñar la estructura de la Base de Datos.

3.1.2.1 DIAGRAMA UML CASOS DE USO

El diagrama de caso de uso nos permite entender los pasos o las actividades que deberán realizarse para llevar a cabo los procesos de registros de matriculación y de calificaciones. Como se muestra en las siguientes figuras:

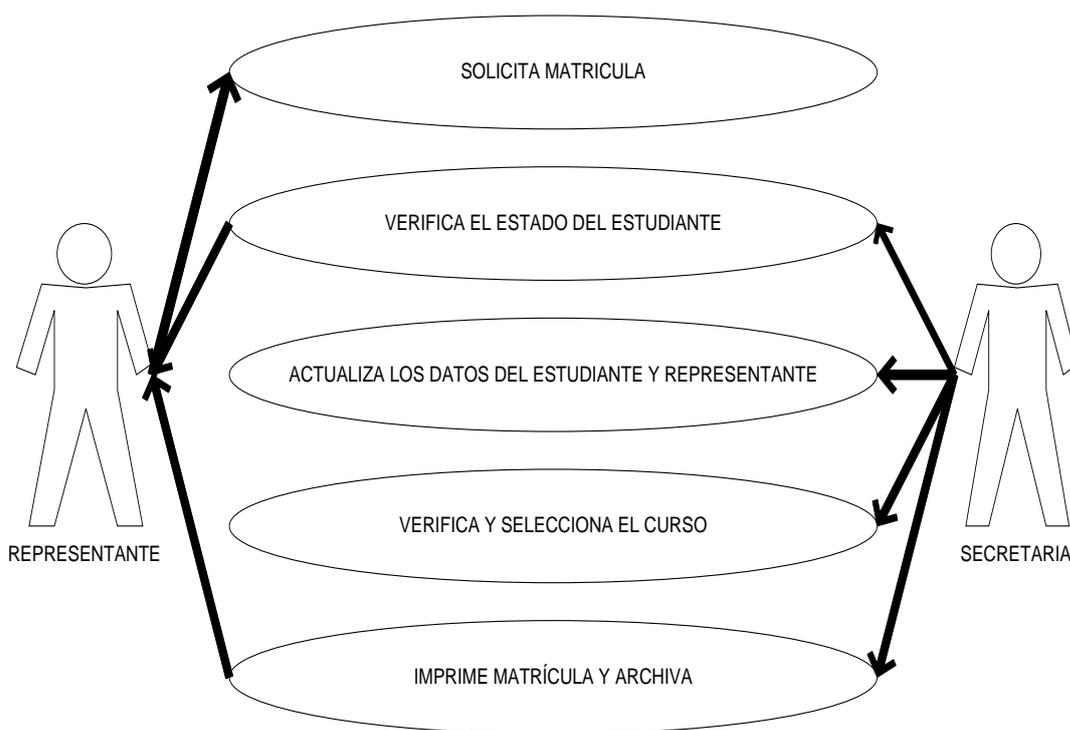


Figura 03.01: Casos de Uso del proceso de registro de matriculación

Elaboración: Los autores

Nº	CASO DE USO	FUNCIÓN
1	Solicitar matrícula	Permite al representante solicitar una matrícula
2	Verificar estado de estudiante	Ayuda a la secretaria a conocer estado del estudiante, si es apto o no para matricularse
3	Actualizar los datos	Una vez obtenidos los datos del estudiante, se procede a actualizarlos.
4	Verificar y seleccionar curso	Permite establecer el curso al que pertenece el estudiante
5	Imprimir y archivar	Permite almacenar la información debidamente e imprimir el reporte de matrícula.

Tabla 03.07: Descripción del caso de uso del proceso de registro de matriculación

Elaboración: Los autores

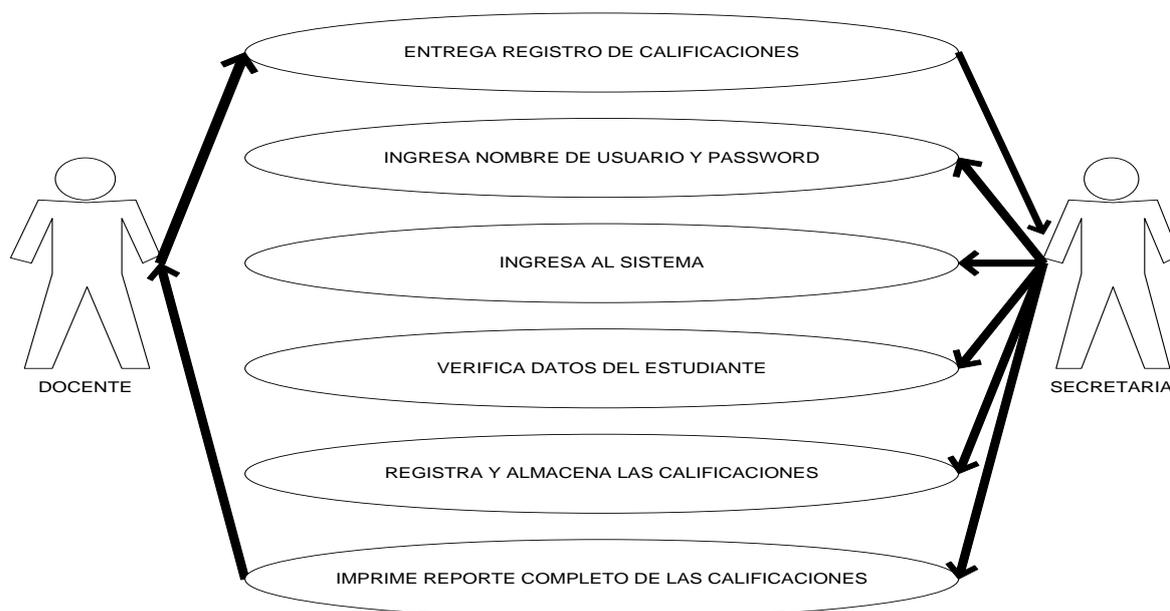


Figura 03.02: Caso de uso del proceso de registro de calificaciones

Elaboración: Los autores

Nº	CASO DE USO	FUNCIÓN
1	Entrega registro de calificaciones	A través de esta acción el docente hace la entrega de los promedios y conductas de los estudiantes, a la secretaria.
2	Ingresa nombre de usuario y contraseña	Permite que la secretaria coloque su nombre de usuario y contraseña respectiva.
3	Ingresa al sistema	Una vez ingresados correctamente los datos de seguridad, la contraseña ingresa al sistema.
4	Verifica datos del estudiante	Permite que la secretaria verifique los datos de los estudiantes
5	Registra y almacena las calificaciones	Registra las calificaciones de los alumnos de un paralelo y periodo seleccionado
6	Imprime reporte completo de las calificaciones	Permite imprimir reporte de calificaciones de los estudiantes.

Tabla 03.08: Descripción del caso de uso de registro de calificaciones

Elaboración: Los autores

3.1.2.2 DIAGRAMA DE FLUJOS DE DATOS DEL PROCESO DE REGISTRO DE MATRICULACIÓN

El representante se acerca al departamento de secretaría y entrega los documentos del estudiante. Si los documentos están correctos la secretaria constata si el estudiante es nuevo o no. Si es nuevo ingresa los datos y los almacena, caso contrario solo se actualiza la información. Luego de haber realizado estos procesos se imprime un reporte de matrícula y se le entrega al representante. El flujo de datos se indica en la figura 03.03

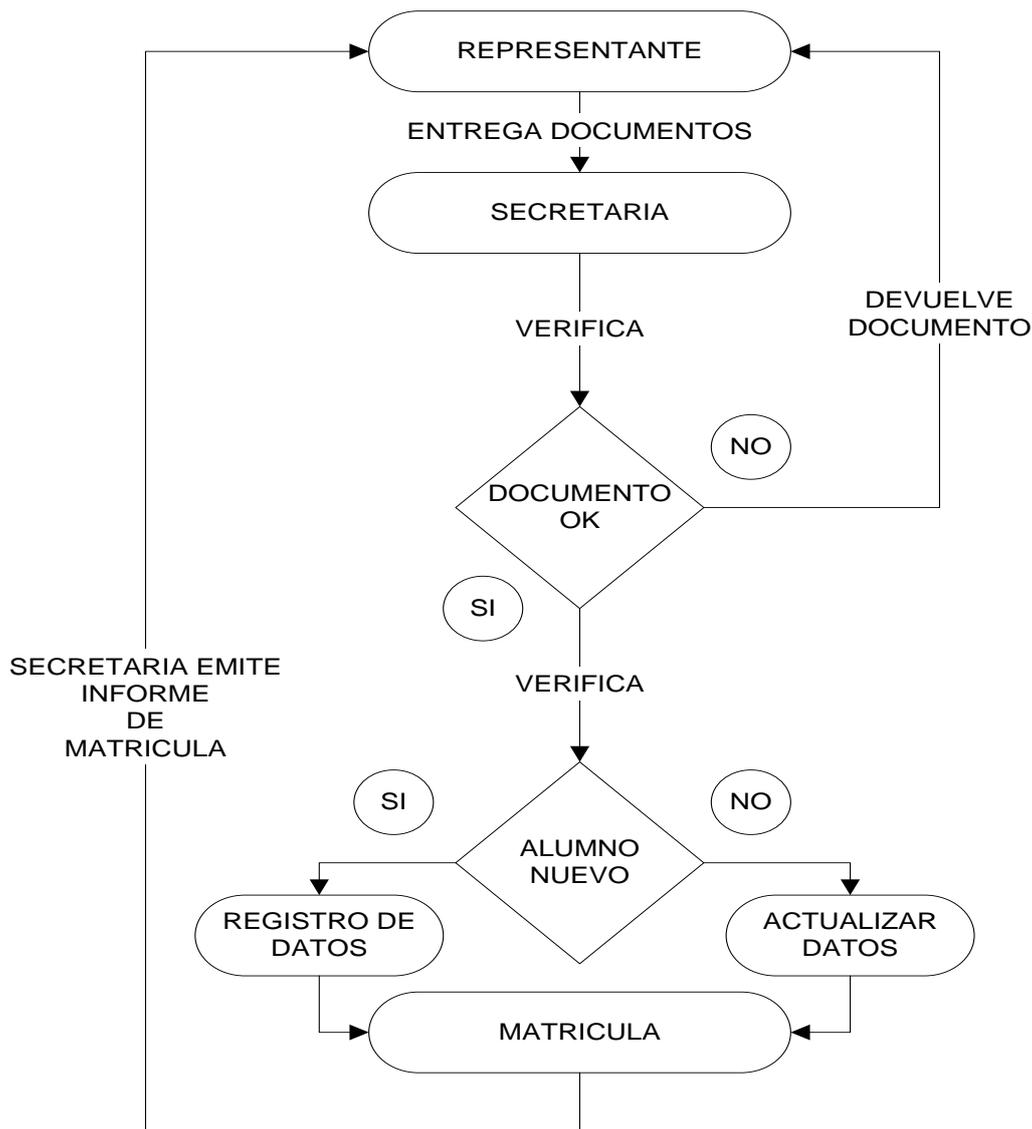


Figura 03.03: Diagrama de flujo de datos del proceso de registro de matriculación

Elaboración: Los autores

3.1.2.3 DIAGRAMA DE FLUJOS DE DATOS DEL PROCESO DE REGISTRO DE CALIFICACIONES

El docente entrega reporte de calificaciones a la secretaria, luego esta corrobora los datos; sino están correctos devuelve los documentos al profesor, caso contrario registra las calificaciones y las almacena. Partiendo de las suma

de las calificaciones se procede a confirmar la aprobación o pérdida del alumno imprimiendo un reporte.

El alumno independientemente puede requerir sus calificaciones finales en el área de secretaria.

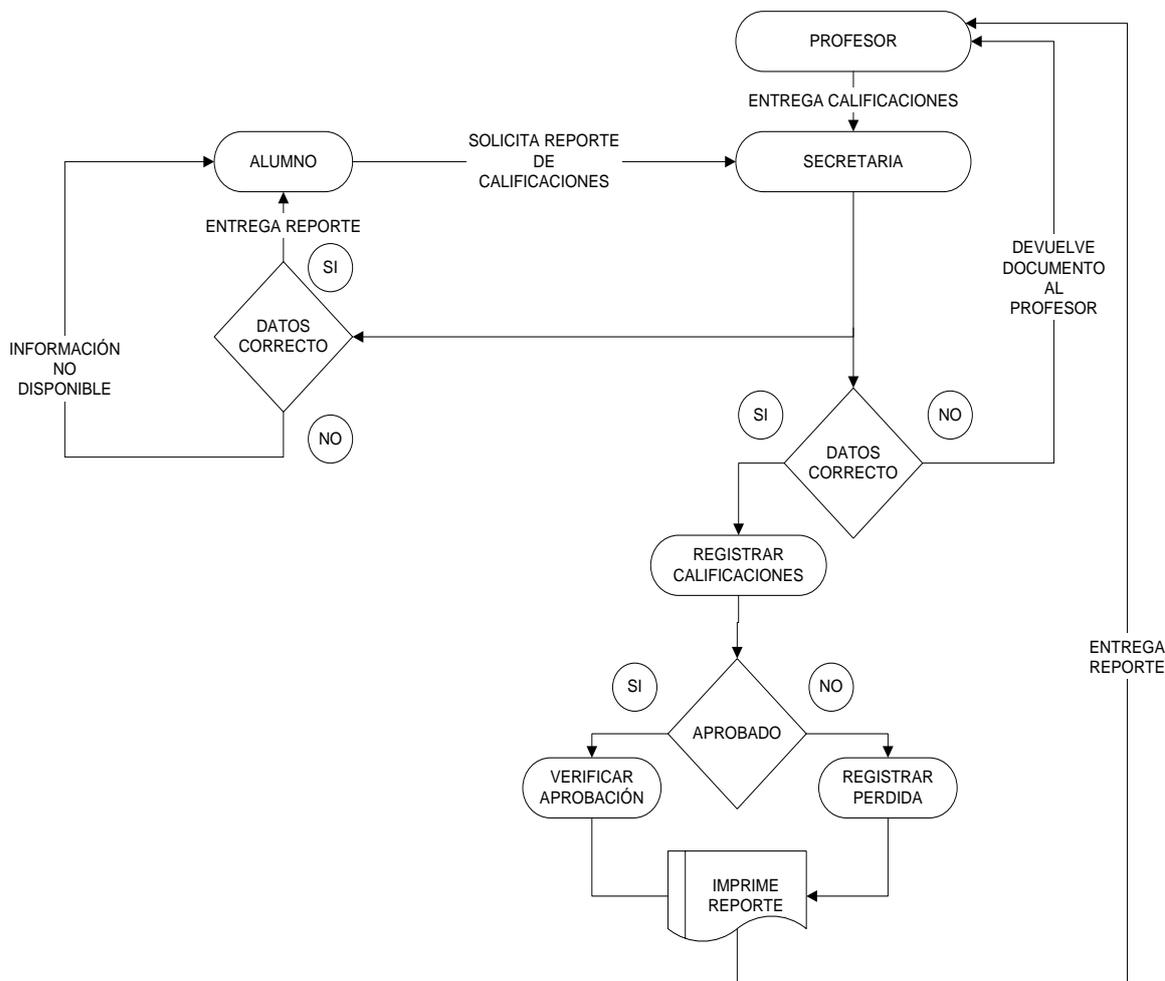


Figura 03.04: Diagrama de flujo de datos del proceso de registro de calificaciones

Elaboración: Los autores

3.1.2.4 DISEÑO DE LA ESTRUCTURA DE LA BASE DE DATOS

La base de datos se elaboró en SQL Server 2008 R2, en la que se crearon diecinueve tablas que contienen información relevante sobre las partes que

intervienen en los procesos de registro de matriculación y de calificaciones en la Unidad Educativa Aristos, las mismas que están debidamente relacionadas para cumplir con los objetivos del sistema, como son las consultas, ingresos, actualizaciones de información entre otros. Entre las tablas creadas se encuentran:

- Alumnos
- Representante
- Docente
- Persona
- Distribución
- Nivel
- Asignatura
- Año de educación básica (AEB)
- Configuración
- Bimestre
- Quimestre
- Periodo lectivo
- calificaciones
- Provincia
- Cantón
- Parroquia
- Matrícula
- Matrícula asignatura
- Usuario

La estructura finalizada de la base de datos, se presenta en la siguiente figura:

A continuación se puntualizan las tablas participantes, las relaciones que existen con su respectiva cardinalidad:

- La relación existente entre la tabla persona y la tabla alumno tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla persona y la tabla representante tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla persona y la tabla docente tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla representante y la tabla alumno tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla alumno y la tabla matricula tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla aeb y la tabla distribución tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla aeb y la tabla matricula tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla nivel y la tabla aeb tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla docente y la tabla distribución tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla asignatura y la tabla distribución tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla periodo lectivo y la tabla distribución tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla periodo lectivo y la tabla matricula tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla periodo lectivo y la tabla quimestre tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla quimestre y la tabla bimestre tiene una cardinalidad de uno a varios.

- La relación existente entre la tabla provincia y la tabla cantón tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla cantón y la tabla parroquia tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla matricula y la tabla matricula_asignatura tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla matricula_asignatura y la tabla calificaciones tiene una cardinalidad de uno a varios.
- La relación existente entre la tabla asignatura y la tabla matricula_asignatura tiene una cardinalidad de uno a varios.
- La relación existente entre la periodo_lectivo y la tabla configuración tiene una cardinalidad de uno a varios.
- La relación existente entre bimestre y la tabla calificaciones tiene una cardinalidad de uno a varios.

DICCIONARIO DE DATOS

TABLA: alumno				
ATRIBUTO O CAMPO	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_alumno	Int		PK	Código del alumno
fecha_inicio	Date			Fecha de nacimiento del alumno
procedencia	nvarchar	(50)		Nombre de la institución de donde proviene el alumno
id_representante	Int		FK	Clave foránea que relaciona con la tabla alumno
id_persona	Int		FK	Clave foránea que relaciona con la tabla alumno
Estado	Bit			Determina si un alumno esta activo o inactivo en la institución.

Tabla 03.09: Diccionario de datos de la tabla alumnos

Elaboración: Los autores

TABLA representante				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_representante	Int		PK	Clave primaria del representante
Teléfono	Nchar	(10)		Teléfono del representante
id_persona	Int		FK	Clave foránea que relaciona con la tabla representante

Tabla 03.10: Diccionario de datos de la tabla representante

Elaboración: Los autores

TABLA persona				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_persona	Int		PK	Clave primaria de la persona
Cedula	nchar	(10)		Cédula de identidad de la persona
Nombre	nvarchar	(50)		Nombres de la persona
Apellido	nvarchar	(50)		Apellidos de la persona
Genero	nvarchar	(50)		Genero de la persona (Masculino o Femenino)
dirección	nvarchar	(50)		Dirección de la persona
nacionalidad	nvarchar	(50)		Nacionalidad de la persona
provincia	Int			Nombre de la provincial a la que pertenece la persona
Canton	nvarchar	(50)		Nombre del cantón al que pertenece la persona
parroquia	nvarchar	(50)		Nombre de la parroquia al que pertenece la persona
fecha_nac	Date			Fecha de nacimiento de la persona
Foto	Image			Insertar una foto

Tabla 03.11: Diccionario de datos de la tabla persona

Elaboración: Los autores

TABLA docente				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_docente	Int		PK	Clave primaria del docente
título	Nchar	(30)		Título del docente con la especialidad
id_persona	Int		FK	Clave foránea que relaciona con la tabla docente
teléfono	nchar	(10)		Teléfono del docente

Tabla 03.12: Diccionario de datos de la tabla docente

Elaboración: Los autores

TABLA distribucion				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_distribucion	Int		PK	Clave primaria de distribución
id_aeb	Int		FK	Clave foránea que relaciona con la tabla distribución
id_docente	Int		FK	Clave foránea que relaciona con la tabla distribución
id_asignatura	Int		FK	Clave foránea que relaciona con la tabla distribución
id_periodo	Int		FK	Clave foránea lectivo que relaciona con la tabla distribución

Tabla 03.13: Diccionario de datos de la tabla distribución

Elaboración: Los autores

TABLA period lectivo				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUDE	CLAVE	DESCRIPCIÓN
id_periodo	Int		PK	Clave primaria de periodo lectivo
anio_inicio	Date			Anio inicio del periodo lectivo
anio_fin	Date			Anio fin del periodo lectivo
nombre_periodo	nvarchar	(50)		Nombre del periodo lectivo
estado_periodo	Bit			Estado del periodo (activo o inactivo)

Tabla 03.14: Diccionario de datos de la tabla periodo lectivo

Elaboración: Los autores

TABLA Quimestre				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_quimestre	Int		PK	Clave primaria del Quimestre
nombre_quimestre	nvarchar	(50)		Nombre del quimestre
id_periodo	Int		FK	Clave foránea que relaciona con la tabla Quimestre

Tabla 03.15: Diccionario de datos de la tabla quimestre

Elaboración: Los autores

TABLA nivel				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_nivel	Int		PK	Clave primaria de nivel
nombre_nivel	nvarchar	(50)		Nombre del nivel

Tabla 03.16: Diccionario de datos de la tabla nivel

Elaboración: Los autores

TABLA bimestre				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_bimestre	Int		PK	Clave primaria de bimestre
nombre_bimestre	nvarchar	(50)		Nombre del bimestre
id_quimestre	Int		FK	Clave foránea que relaciona con la tabla bimestre

Tabla 03.17: Diccionario de datos de la tabla bimestre

Elaboración: Los autores

TABLA aeb				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUDE	CLAVE	DESCRIPCIÓN
id_aeb	Int		Pk	Clave primaria de año de educación básica
Aeb	nvarchar	(50)		Nombre del año de educación básica
paralelo_aeb	Nvarchar	(2)		Nombre del paralelo (A, B, C....)
id_nivel	Int		FK	Clave foránea que relaciona con la tabla año de educación básica
Numero	Int			Seleccionar número dependiendo del año de educación básica

Tabla 03.18: Diccionario de datos de la tabla AEB

Elaboración: Los autores

TABLA asignatura				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_asignatura	Int		PK	Clave primaria de la asignatura
nombre_asignatura	nchar	(20)		Nombre de la asignatura
Optative	Bit			Asignaturas optativas

Tabla 03.19: Diccionario de datos de la tabla Asignatura

Elaboración: Los autores

TABLA usuario				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_usuario	Int		PK	Clave primaria de usuario
nombre_usuario	nvarchar	(50)		Nombre del usuario
contraseña_usuario	nvarchar	(50)		Contraseña de usuario
tipo_usuario	nchar	(20)		Tipo de usuario
m_imagen	bit			
Imagen	image			Asigna imagen

Tabla 03.20: Diccionario de datos de la tabla Usuario

Elaboración: Los autores

TABLA parroquia				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_parroquia	Int		PK	Clave primaria de parroquia
nombre	nvarchar	(50)		Nombre de parroquia
id_canton	Int		FK	Clave foránea que relaciona con la tabla provincia

Tabla 03.21: Diccionario de datos de la tabla parroquia

Elaboración: Los autores

TABLA matrícula				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_matricula	Int		PK	Clave primaria de la matrícula
id_aeb	Int		FK	Clave foránea que relaciona con la tabla Matrícula
id_alumno	Int		FK	Clave foránea que relaciona con la tabla matrícula
fecha_matricula	Date			Fecha de matriculación
id_periodo	Int		FK	Clave foránea que relaciona con la tabla matrícula

Tabla 03.22: Diccionario de datos de la tabla matricula

Elaboración: Los autores

TABLA provincia				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_provincia	Int		PK	Clave primaria de provincial
Nombre	nvarchar	(50)		Nombre de la provincia

Tabla 03.23: Diccionario de datos de la tabla provincia

Elaboración: Los autores

TABLA canton				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_canton	Int		PK	Clave primaria del canton
Nombre	nvarchar	(50)		Nombre del cantón
id_provincia	Int		FK	Clave foránea que relaciona con la tabla provincia

Tabla 03.24: Diccionario de datos de la tabla Cantón

Elaboración: Los autores

TABLA matricula_asignatura				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_mat_asig	Int		PK	Clave primaria de material asignatura
id_matricula	Int		FK	Clave foránea que relaciona con la tabla matrícula asignatura
id_asignatura	Int		FK	Clave foránea que relaciona con la tabla matrícula asignatura
promedio	Float			Promedio
supletorio	Float			Supletorio
Estado	Nchar	(10)		(aprobado o reprobado)

Tabla 03.25: Diccionario de datos de la tabla asignatura matricula

Elaboración: Los autores

TABLA calificaciones				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
Id_mat_asig	int		PK	Clave primaria de calificaciones
Id_bimestre	int		PK	Clave primaria de calificaciones
Apro	decimal (18,2)			Parcial aproximación
Conc	decimal (18,2)			Parcial conceptualización
[proc]	decimal (18,2)			Parcial procesos
Argu	decimal (18,2)			Parcial argumentación
Cpa	decimal (18,2)			Parcial cpa
Suma	decimal (18,2)			Suma de los parciales
Disc	decimal (18,2)			Disciplina
agregado	bit			Estado de calificaciones

Tabla 03.26: Diccionario de datos de la tabla calificaciones

Elaboración: Los autores

TABLA configuracion				
NOMBRE DE COLUMNA	TIPO DE DATOS	LONGITUD	CLAVE	DESCRIPCIÓN
id_configuracion	int		PK	Clave primaria de table configuración
n_minima	decimal (18,2)			Nota minima
n_maxima	decimal (18,2)			Nota máxima
Aprobado	decimal (18,2)			Nota para aprobado
Supletorio	decimal (18,2)			Nota para supletorio
Reprobado	decimal (18,2)			Nota para reprobado
id_periodo	int		FK	Clave foránea

Tabla 03.27: Diccionario de datos de la tabla configuración

Elaboración: Los autores

3.1.2.5 DISEÑO DE INTERFACES

La presentación del programa ante el usuario, se manejó con interfaces que cumplen con uno de los objetivos primordiales, el cual es facilitar al usuario la interacción con la aplicación. Para esto se utilizó patrones predefinidos para cada tipo de aplicación y para cada necesidad del usuario. En esta capa se diseñó la interfaz gráfica del software, las ventanas, los enlaces, y los colores se escogieron de acuerdo a los tonos que representan a la institución. La interfaz en forma general es de sentido amigable para la persona encargada de administrar el sistema, se incluyeron diferentes estilos en la pantalla principal, en la parte superior se ubicó un menú mediante el componente Menústrip, el mismo que sirvió de enlace para acceder a los otros formularios. Cada formulario tenía adjuntados controles básicos como buttons, label, textbox, listbox, combobox, datagridview, Menústrip, entre otros que sirvieron de complemento para manipular con éxito la información.

En la parte derecha de la pantalla principal se adjuntó un panel expandible mediante el componente Expandablesplitter, el cual conjuntamente con la aplicación Donetbar permitió crear un menú dinámico con iconos y animaciones.

Se le adjuntó la opción Status Strip, la cual mediante el control timer muestra la hora en la barra de estado.

La opción ContextMenúStrip que permite visualizar en dos columnas los iconos incorporados en el panel expandible.

Desde la barra de herramientas se aplicó la opción SuperTool, el mismo que aceptó que al momento de pasar el puntero del mouse por un icono, aparezca una vista previa de lo que contiene ese vínculo.

Entre las ventanas principales del sistema de registro de matriculación y de calificaciones se encuentran las siguientes:

ACCESO AL SISTEMA

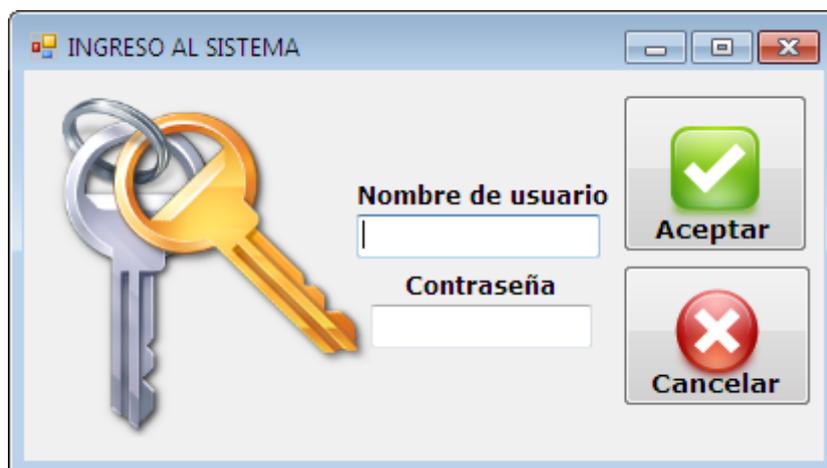


Figura 03.06: Ingreso al sistema

Elaboración: Los autores

VENTANA PRINCIPAL



Figura 03.07: Pantalla principal

Elaboración: Los autores

FORMULARIO DE REGISTRO DE MATRÍCULA

Matrícula

Nivel:

AEB:

Buscar por:

CEDULA	ALUMNO	FECHA MATRÍCULA	AEB	PERIODO LECTIVO
1313136424	JUAN MUÑOZ	22/08/2012	NIVEL A	PERIODO LECTIVO 2012-2013
1313136432	MARTHA CRUZ	22/08/2012	PRIMERO A	PERIODO LECTIVO 2012-2013
1312536483	DOMENICA CECIBEL CARRERRA SOLIS	23/08/2012	PRIMERO A	PERIODO LECTIVO 2012-2013
1313136473	PEDRO LOPEZ	22/08/2012	NIVEL B	PERIODO LECTIVO 2012-2013
1310621964	VIVIANA CLARIBEL GARCIA MENEDEZ	23/08/2012	QUINTO A	PERIODO LECTIVO 2012-2013
1312850496	MARIANO MENDOZA	23/08/2012	NOVENO A	PERIODO LECTIVO 2012-2013

MATRÍCULA

Periodo lectivo *

Fecha *

Alumno *

Nivel *

AEB *

Estado

Figura 03.08: Formulario de registro de matriculación

Elaboración: Los autores

FORMULARIO DE REGISTRO DE CALIFICACIONES

Calificaciones

Periodo lectivo: *

Nivel: * AEB: *

Docente: * Materia: *

Quimestral Total

Quimestre: * Bimestre: *

APPELLIDOS Y NOMBRES	Apro	Conc	Proc	Argu	CPA	SUMA	DISC	INGRESADO
CARRERRA SOLIS DOMENICA CECIBEL	2,00	2,00	2,00	2,00	2,00	10,00	9,00	<input type="checkbox"/>
CRUZ MARTHA	2,00	2,00	2,00	2,00	2,00	10,00	2,00	<input checked="" type="checkbox"/>

Estado

Figura 03.09: Formulario de registro de calificaciones

Elaboración: Los autores

3.1.3 FASE DE CONSTRUCCIÓN

3.1.3.1 IMPLEMENTACIÓN

Este apartado se realizó como una actividad de la fase anterior del modelo de desarrollo de software en tres capas (Ver figura 03.10) debido a que después de haber efectuado el diseño correspondiente en la capa de presentación, (Ver página 71 - 73) se procedió a continuar el desarrollo del software en la capa de lógica de negocio. La plataforma empleada para realizar el sistema fue Visual Basic.NET 2010 porque es una herramienta de desarrollo orientada a objetos y eventos que provee una gran variedad de herramientas lo cual facilitó el trabajo en los procesos.

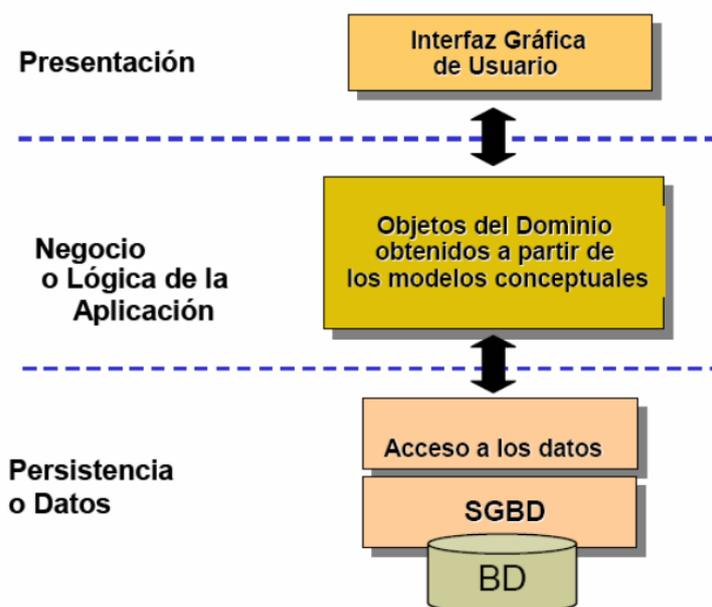


Figura 03.10: Esquema de la arquitectura de programación en tres capas

Fuente: (Maldonado, D. 2010)

En esta fase es donde se encuentra toda la lógica del programa, aquí se procedió a introducir todo el código fuente, se elaboró la estructura de datos y objetos encargados para la manipulación de los datos existentes, así como el procesamiento de la información ingresada o solicitada por el usuario en la capa de presentación.

Es aquí donde se hace el uso de funciones, estas se diseñaron para que interactúen de forma óptima con el código (Ver anexo N° 5).

Siguiendo el orden basado en la arquitectura de programación en tres capas (vea figura 03.10) en la capa de datos se dio paso a realizar transacciones con la base de datos, el manejo y manipulación de estos se los realizo de forma tal que existiera consistencia en los mismos, permitiendo que los datos que se ingresan así como los que se extraen de la base de datos sean consistentes y precisos. Se sirvió utilizar la plataforma SQL Server 2008 R2, en esta aplicación se crearon las tablas pertinentes que permitieron plasmar las diversas consultas que se necesitaban (Ver anexo N° 5, Pág. 108).

3.1.3.2 PRUEBAS

Después de implementar los formularios, las clases y las funciones en el sistema se procedió a realizar las diferentes pruebas de software, empleando el tipo de prueba de integración. Este flujo de trabajo se encargó de evaluar la calidad del producto desarrollado que permitió tener un aseguramiento de la calidad del sistema bajo estas condiciones:

- **IDENTIFICAR ERRORES INTRODUCIDOS POR LA COMBINACIÓN DE PROGRAMAS PROBADOS UNITARIAMENTE.**

En este contexto fue donde se pudo corroborar que la conexión establecida entre la plataforma de desarrollo Visual Basic.net 2010 y Sql Server 2008 R2 había sido correctamente estructurada, la comunicación existente entre ambas permitió tener una interacción y manipulación correcta de la información sin presentarse errores al momento de realizar las diferentes consultas.

- **VERIFICAR QUE LAS INTERFACES Y LAS APLICACIONES FUNCIONEN CORRECTAMENTE.**

Mediante el ingreso y almacenamiento de información ficticia de las entidades participantes en el sistema, se detectaron un sin número de errores que poco a poco se fueron corrigiendo. Entre las tablas alumnos, docente y representante se produjeron inconvenientes al momento de almacenar los datos, puesto que no se había validado correctamente los controles. Al momento de ingresar un alumno, este a su vez podía ser almacenado como un docente o como un representante, y lo mismo sucedía cuando se pretendía guardar los datos del docente o representante. Dicha información era errónea ya que un estudiante no podía ser un profesor y tampoco un estudiante. Y un profesor no podía ser un alumno. Percibido el percance se procedió a validar los controles para que este tipo de inconvenientes no se presente.

Esta misma acción se ejecutó con cada una de las ventanas que conforman el sistema.

- **COMPROBAR QUE LAS ESPECIFICACIONES DE DISEÑO SEAN ALCANZADAS.**

La interfaz gráfica resultó ser amigable y comprensible para los usuarios. Para comprobarlo se optó por buscar a una persona no experta con la finalidad de que manipulara información desde la aplicación, después de capacitarlo de una forma informal y muy rápida. Una vez analizada y evaluada su participación en la conducción del software se llegó a la determinante que sus interfaces resultaron ser de fácil manejo.

- **EVIDENCIAR CUÁLES SON LOS REQUERIMIENTOS DE IMPLEMENTACIÓN PERTINENTES PARA INSTALAR EL SISTEMA.**

En este sentido el objetivo de esta condición fue identificar el comportamiento de la aplicación informática bajo condiciones que sobrecargan sus recursos. Se probó el sistema en un computador con memoria de 512 Mb y disco duro de 80 Gb, con lo cual se constató que el software se ejecutaba, pero con un tiempo considerable de retraso. Por tal motivo se procedió a probar el sistema en un computador con características más avanzadas (ver requisitos funcionales en tabla 03.04). Luego de este análisis se obtuvo el funcionamiento eficaz y rápido del sistema.

Es de suma importancia recalcar que una vez perpetradas todas las pruebas pertinentes, se realizó en esta misma fase el manual de usuario, que se encuentra adjunto en los anexos (Ver anexo N° 6).

3.1.4 FASE DE TRANSICIÓN

Luego de probar el producto y constatar que no existían errores de instalación, errores de sintaxis y errores de conexión a la base de datos, se procedió a efectuar la instalación en la computadora que está ubicada en el área de secretaría. Una vez instalado se empezó a utilizar el sistema, esta etapa requirió en brindar soporte a la Srta. Ana Dolores Santos con la finalidad de que se familiarice y adquiriera destrezas en el manejo de la aplicación. Se le otorgo asistencia, capacitación y la entrega del manual de usuario que le sirvió de guía durante la puesta productiva del software.

Efectuadas las primeras prácticas se verifico que el producto cumplía con las especificaciones entregadas en la fase de inicio, logrando así optimizar y agilizar los procesos de registro de matriculación y de calificaciones de los estudiantes

3.2 TÉCNICAS

- Observación
- Entrevista

3.3 RECURSOS

3.3.1 TALENTOS HUMANOS

Tecnóloga Cruz López Martha Gabriela (Postulante)

Tecnólogo Muñoz Mendoza Juan Carlos (Postulante)

3.3.2 MATERIALES

Los materiales que se utilizaran son los siguientes:

- Hojas tamaño A4
- Lapiceros
- Libros
- Carpetas

3.3.3 TECNOLÓGICOS

3.3.3.1 HARDWARE

- Flash Memory
- Laptop
- Impresoras
- CDs y DVDs
- Camaras

3.3.3.2 SOFTWARE

- Microsoft Visual Basic .NET 2010
- Microsoft SQL Server 2008 R2
- Microsoft Word 2010
- Microsoft Excel 2010
- Microsoft Power Point 2010
- Microsoft Visio 2010
- Adobe Photoshop CS3
- Cristal Reports

3.3.4 PRESUPUESTO Y FUENTES DE FINANCIAMIENTO

3.3.4.1 PRESUPUESTO

ACTIVIDADES	MATERIALES	CANTIDAD	COSTO UNITARIO	TOTAL
Recopilación de información	Hojas	32	0.01	0.32
	Lapiceros	4	0.25	1.00
	Copias	15	0.02	0.30
	Internet	50	0,75	37.50
Denuncia del tema	EspeciesValoradas	1	1.00	1.00
	Impresiones	18	0.20	3.60
	Carpeta	2	0.15	0.30
Desarrollo de la tesis	Software	1	150	250.00
	Internet	300	0.75	225.00
	Otros	-	-	80.00
Elaboración de la tesis	Impresiones	1500	0.20	300.00
	Empastados	4	15.00	60.00
	Otros	-	-	80.00
TOTAL				1069.02

Tabla 03.28: Presupuesto

Elaboración: Los autores

3.3.4.2 FUENTES DE FINANCIAMIENTO

Los autores del proyecto financiaron por sus propios medios los materiales, movilizaciones, y gastos varios en que se incurrieren durante el desarrollo de la tesis y los costos de las licencias de software lo financio la Escuela Superior Politécnica Agropecuaria de Manabí Manuel Félix López.

CAPITULO IV. RESULTADOS Y DISCUSIÓN

4.1 RESULTADOS

Una vez realizada la entrega e instalación del sistema de proceso de registro de matriculación y de calificaciones en la Unidad Educativa Aristos de la ciudad de Calceta del Cantón Bolívar se pudo obtener los siguientes resultados:

- A través de las visitas in situ y después de una serie de entrevistas con la secretaria de la institución se plantearon los requisitos funcionales, no funcionales y de implementación del sistema.
- Con los requisitos obtenidos se crearon los diagramas de caso de uso y los diagramas de flujos de datos de los procesos de registro de matriculación y de calificaciones, los mismos que dieron paso a diseñar la estructura de la base de datos en SQL Server 2008 R2.
- Seguidamente se diseñaron las interfaces graficas del sistema, resultando de sentido amigable para la persona encargada de administrarlo, mediante la inclusión de formularios que sirvieron de complemento para manipular con éxito la información.
- Una vez diseñadas las interfaces se elaboraron patrones de lógica de programación. La construcción del software fue desarrollada en Visual Basic.net 2010 basándose en el diseño de datos donde se utilizaron las instrucciones necesarias para manejar la nueva aplicación y se estableció la conexión con la base de datos realizada en SQL Server 2008 R2 para desarrollar las estructuras de los archivos lógicos en los que se almacenaría la información.

- Después de tener el software terminado se ejecutaron las pruebas en el sistema que consistió en validar la información introducida y de manejar los errores que puedan ocurrir, los mismos que fueron corregidos exitosamente.
- Por medio de la Implementación del Sistema Automatizado se logró una mayor eficiencia y efectividad en los procesos que se realizan en la Unidad educativa Aristos de la ciudad de Calceta. Estos resultados se muestran en la siguiente tabla:

4.1.1 COMPARATIVO ENTRE LA OBSERVACIÓN PREVIA Y DESPUES DE LA EJECUCIÓN DE LA APLICACIÓN.

Orden de Alumno	Proceso	Sin Sistema		Con Sistema		Optimización
		Tiempo Ingreso	Tiempo Acumulado	Tiempo Ingreso	Tiempo Acumulado	
1º Alumno	REGISTRO DE MATRICULACIÓN	0:16:00	0:16:00	0:05:00	0:05:00	69%
2º Alumno		0:14:00	0:30:00	0:05:00	0:10:00	
3º Alumno		0:15:00	0:45:00	0:04:00	0:14:00	
4º Alumno		0:13:00	0:58:00	0:06:00	0:20:00	
5º Alumno		0:09:00	1:07:00	0:02:00	0:22:00	
6º Alumno		0:12:00	1:19:00	0:02:00	0:24:00	
7º Alumno		0:13:45	1:32:45	0:03:00	0:27:00	
8º Alumno		0:13:05	1:45:50	0:07:00	0:34:00	
9º Alumno		0:14:00	1:59:50	0:02:05	0:36:05	
10º Alumno		0:13:00	2:12:50	0:05:05	0:41:10	
TOTAL			12:26:15		3:53:15	
PROMEDIO			1:14:37		0:23:19	

Tabla 04.01: Cuadro comparativo entre la observación previa y después de la ejecución de la aplicación (Registro de matriculación)

Fuente: Entrevista a la Secretaria

Orden de Alumno	Proceso	Sin Sistema		Con Sistema		Optimización
		Tiempo Ingreso	Tiempo Acumulado	Tiempo Ingreso	Tiempo Acumulado	
1º Alumno	REGISTRO DE CALIFICACIONES	0:05:05	0:05:05	0:03:30	0:03:30	45%
2º Alumno		0:06:00	0:11:05	0:03:04	0:06:34	
3º Alumno		0:04:30	0:15:35	0:04:00	0:10:34	
4º Alumno		0:07:00	0:22:35	0:03:00	0:13:34	
5º Alumno		0:05:40	0:28:15	0:02:49	0:16:23	
6º Alumno		0:06:00	0:34:15	0:02:29	0:18:52	
7º Alumno		0:06:00	0:40:15	0:03:00	0:21:52	
8º Alumno		0:08:00	0:48:15	0:04:00	0:25:52	
9º Alumno		0:09:00	0:57:15	0:02:05	0:27:57	
10º Alumno		0:06:00	1:03:15	0:05:05	0:33:02	
	TOTAL		5:25:50		2:58:10	
	PROMEDIO		0:32:35		0:17:49	

Tabla 04.02: Cuadro comparativo entre la observación previa y después de la ejecución de la aplicación (Registro de calificaciones)

Fuente: Entrevista a la Secretaria

Orden de Alumno	Proceso	Sin Sistema		Con Sistema		Optimización
		Tiempo Ingreso	Tiempo Acumulado	Tiempo Ingreso	Tiempo Acumulado	
1º Alumno	CONSULTA DE DATOS	0:20:00	0:20:00	0:05:05	0:05:05	71%
2º Alumno		0:18:00	0:38:00	0:06:00	0:11:05	
3º Alumno		0:21:00	0:59:00	0:04:30	0:15:35	
4º Alumno		0:16:00	1:15:00	0:04:00	0:19:35	
5º Alumno		0:18:00	1:33:00	0:05:40	0:25:15	
6º Alumno		0:20:00	1:53:00	0:06:00	0:31:15	
7º Alumno		0:19:00	2:12:00	0:06:00	0:37:15	
8º Alumno		0:20:00	2:32:00	0:08:00	0:45:15	
9º Alumno		0:22:00	2:54:00	0:06:00	0:51:15	
10º Alumno		0:16:00	3:10:00	0:06:00	0:57:15	
	TOTAL		17:26:00		4:58:50	
	PROMEDIO		1:44:36		0:29:53	

Tabla 04.03: Cuadro comparativo entre la observación previa y después de la ejecución de la aplicación (Consulta de datos)

Fuente: Entrevista a la Secretaria

Orden de Alumno	Proceso	Sin Sistema		Con Sistema		Optimización
		Tiempo Ingreso	Tiempo Acumulado	Tiempo Ingreso	Tiempo Acumulado	
1º Alumno	ADQUISICIÓN DE REPORTE	0:08:00	0:08:00	0:05:05	0:05:05	45%
2º Alumno		0:10:00	0:18:00	0:06:00	0:11:05	
3º Alumno		0:06:00	0:24:00	0:04:30	0:15:35	
4º Alumno		0:12:00	0:36:00	0:04:00	0:19:35	
5º Alumno		0:09:45	0:45:45	0:05:40	0:25:15	
6º Alumno		0:13:00	0:58:45	0:06:00	0:31:15	
7º Alumno		0:10:00	1:08:45	0:06:00	0:37:15	
8º Alumno		0:08:30	1:17:15	0:04:00	0:41:15	
9º Alumno		0:10:00	1:27:15	0:06:00	0:47:15	
10º Alumno		0:09:00	1:36:15	0:06:00	0:53:15	
	TOTAL		8:40:00		4:46:50	
	PROMEDIO		0:52:00		0:28:41	

Tabla 04.04: Cuadro comparativo entre la observación previa y después de la ejecución de la aplicación (Adquisición de reportes)

Fuente: Entrevista a la Secretaria

Para la realización de los cuadros comparativos, se tomaron como muestra 10 alumnos, en los cuales los procesos se realizaron en diferentes niveles de tiempos, debido a que estos varían dependiendo si se trata de un primer ingreso o una actualización de dato. Se creó una columna para calcular el tiempo acumulado y posteriormente se sacó el promedio en lo referente a lo que se tarda la secretaria en realizar los procesos sin el sistema y con el sistema. Estos dos resultados permitieron constatar que los procesos se han optimizado, es decir gracias a la utilización del sistema informático se redujeron los tiempos al realizar los procesos de registro de matriculación y de calificaciones.

4.2 DISCUSIÓN

En toda institución la información es un recurso estratégico que influye, en gran medida y de modo fundamental, lo que es muy importante tanto que esta información se gestione eficaz y eficientemente para el logro de los objetivos relacionados con la calidad, es decir, que reúnan, almacenen, procesen y proporcionen información relevante. De esta forma, dicha información debe ser accesible y estar disponible para quienes desean utilizarla. Si bien es cierto, en este mundo sumergido en los avances tecnológicos, las aplicaciones informáticas se han convertido en una herramienta de apoyo eficiente, que permiten automatizar y obtener datos exactos en las organizaciones (Sánchez, S. 2003). Haciendo uso de esta tecnología se desarrolló un software informático de ingreso de matrículas y de calificaciones para la escuela fiscal mixta “Portete de Tarqui” de la parroquia colón, ciudad de Portoviejo” (Macías, M. 2010), del mismo se evidenció que los procesos de almacenamiento de datos eran correctos, pero que no se habían incluido herramientas de búsqueda, para que mediante filtros establecidos se obtenga de forma más eficiente la información, otro de los aspectos que se analizaron es que se manejaban las ventanas de forma separada, por ejemplo: Para realizar el ingreso de un estudiante se usaba un formulario, para la modificación del mismo otro, y para la eliminación se efectuaba la misma acción, de esta manera cada vez que se requería realizar una acción diferente era necesario salir de un formulario para tener acceso. De igual manera al tomar como referencia a otro sistema similar desarrollado en la misma universidad (Escobar, et al. 2010) se detectó que el sistema manejaba debidamente los procesos, incluía todos los reportes pertinentes y necesarios que se utilizan en una unidad educativa. Sin embargo las interfaces del sistema eran simples. Otro aspecto es que al momento de llenar los datos de vivienda de un estudiante no se contaba con un control que contenga la información almacenada de las provincias, parroquias y cantones, lo cual sirvió para que los autores tomen como referencia la lista de las provincias con sus respectivas parroquias y cantones. Además el sistema de registro de matriculación y de

calificaciones para la unidad educativa Aristos cuenta con interfaces de sentido amigable, con formularios que permiten que la información se muestre más organizada. Contiene filtros que permiten realizar las búsquedas en poco tiempo y de manera segura. Integra diferentes acciones en un solo formulario, para que todas sean realizadas en el mismo sin la necesidad de salir, con la finalidad manejar la información de forma más exacta y rápida que aporten positivamente en las actividades realizadas por la secretaria.

CAPÍTULO V. CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- La recolección de información sirvió de base para la realización de un análisis exhaustivo, el mismo que permitió plantear los diferentes requerimientos establecidos entre los directivos de la Unidad Educativa Aristos y los Autores (Ver anexo N° 3).
- La elaboración de una base de datos lógicamente organizada y estructurada, facilitó la manipulación de datos permitiendo efectuar consultas, ingresos, actualizaciones, entre otros procesos. (Ver Figura 03.011)
- En el desarrollo de la aplicación se consideraron los requerimientos establecidos por los directivos de la Unidad Educativa Aristos, utilizando una interfaz gráfica amigable (Ver Figura 03.07).
- Al ingresar los datos y al manipular los diferentes controles, se pudieron localizar las falencias y corregirlas acertadamente (Ver paginas 75 - 77).
- La implementación del sistema informático en la Unidad Educativa Aristos, permitió optimizar y agilizar los procesos de registro de matriculación y de calificaciones. (Ver tabla 04.05, tabla 04.02, tabla 04.03, tabla 04.04)

5.2 RECOMENDACIONES

- Que una vez recolectados los datos, se ordenen, clasifiquen y se seleccionen solo los necesarios, para no tener exceso de información que pudiera ser inútil.

- Que en el diseño de la base de datos se empleen tablas y campos relevantes, aplicando políticas de seguridad para conservar su integridad.

- Que al desarrollar la aplicación la interfaz sea lo más entendible posible, para que los NO expertos puedan usarlo sin complicaciones.

- Que se realice las pruebas pertinentes del funcionamiento eficaz del software, antes de su entrega total.

- Que se emplee un computador con los requisitos apropiados que permita el óptimo funcionamiento del sistema.

BIBLIOGRAFÍA

- Abad, J. 2005. Pruebas de Software (En línea). Consultado, el 21 de julio del 2012. Formato (HTML). Disponible en: <http://ing-sw.blogspot.com/>
- Alegre, M. 2010. Sistemas operativos monopuestos: Tipos de software. Carmen Lara Carmona. 1 ed. Madrid, ES. Paraninfo y no editorial Paraninfo. P 13.
- Barranco, J. s.f. metodología del análisis estructurado de sistemas: prueba de carga. Belén roció. Ed. rev. Madrid. ES. Ortega. p 468.
- Cáceres, T. 2010. Diagramas de casos de uso (En línea). Consultado, el 16 de Julio del 2011. Formato (PDF). Disponible en: <http://www2.uah.es/>
- Calle, F. 2008. Características de programación en capas. (En línea). Consultado, el 16 de julio del 2011. Formato (HTML). Disponible en: <http://www.slideshare.net>
- Campderrich, B. 2003. Ingeniería de software: Definición de diagrama UML casos de uso. UOC. 1 Ed. Cataluña. ES. Graficas Rey. p 83.
- Ceballos, J; Gañán, J; Conesa, J y Rius, A.2010. Introducción a .NET: Plataforma.NET. Sonia Poch. 1 ed. Barcelona, ES .UOC. p 15.
- Charte, F. 2011. Visual Basic 2010: Características de Visual Basic 2010. Anaya Multimedia. Ed. Rev. México. ISBN. p 45.
- Cota, 1994. Ingeniería de software, soluciones avanzadas (En línea). Consultado, el 15 de Julio del 2011. Formato (HTML). Disponible en: <http://www.angelfire.com/>

Cuartas, J. 2011. Modelo Entidad Relación (En línea). Consultado, el 21 de julio del 2012. Formato (HTML). Disponible en: <http://www.alegsa.com.ar/>

Escala, E. 2008. Software de aplicación (En línea). Consultado, el 21 de julio del 2012. Formato (HTML). Disponible en: <http://www.slideshare.net/>

Flores, P. 2008. Tipos de software (En línea). Consultado, el 21 de julio del 2012. Formato (HTML). Disponible en <http://www.slideshare.net/>

Gracia, J. 2010. Modelado UML, Tipos de UML (En línea). Consultado, el 16 de Julio del 2011. Formato (HTML). Disponible en: <http://www.ingenierossoftware.com/>

González, J. 2009. SQL Server 2008 R2: publicadas las primeras características (En línea). Consultado, el 21 de julio del 2012. Formato (HTML). Disponible en: <http://jcgonzalezmartin.wordpress.com/>

Groussard, T. 2010. Visual Basic.NET: Visual Basic.NET 2010. Carlos Walzer. Ed. Rev. Barcelona, ES. ENI (Marca comercial registrada de ediciones software). p 159.

Jacobson, 1998. Ingeniería de software (En línea). Consultado, el 15 de Julio del 2011. Formato (HTML). Disponible en: <http://www.angelfire.com/>

Kruchten, P. 2002. "The Rational Unified Process, An Introduction." (En línea). Consultado, el 10 de mayo del 2012. Formato (HTML). Disponible en: <http://www-306.ibm.com/>

Landacay, K. 2010. Diagramas UML casos de uso (En línea). Consultado, el 21 de julio del 2012. Formato (HTML). Disponible en: <http://www.slidershare.net/>

Lemus, G. 2012. Los tipos de pruebas (En línea). Consultado, el 21 de julio del 2012. Formato (HTML). Disponible en: <http://www.slideshare.net/>

Lewis, 1994. Generalidades de ingeniería de software (En línea). Consultado, el 15 de Julio del 2011. Formato (HTML). Disponible en: <http://www.angelfire.com/>

Laudon, K; Laudon, J. 2004. Sistemas de información gerencial: software de sistema. Editor Enrique Quintanar. 8 Ed. México. ISBN 0-13-101498-6. p 193

Martin, J. s.f. Programación en tres capas: Lógica de negocios. (En línea). Consultado, 18 de Julio del 2011. Formato (HTML). Disponible en: www.joseluismartin.info/

Martínez, L. 2005. Introducción a la ingeniería del software: Pruebas de integración. Javier Barbero Rubio. Zaragoza. ES. 1 Ed. Publicaciones DELTA. p 393.

Mayorca, D. 2004. Programación en tres capas: Capa presentación. (En línea). Consultado, 18 de Julio del 2011. Formato (HTML). Disponible en: www.di.uniovi.ec

Mendoza, M. 2004. Grafico del modelo RUP. (En línea). Consultado, 18 de Julio del 2011. Formato (HTML). Disponible en: www.informatizate.net

Mesa, E. 2010. Diagramas UML de clases (En línea). Consultado, el 19 de Julio del 2011. Formato (HTML). Disponible en: uml/mesaesteban.blogspot.com.html

Mingo, M. 2009. Sistema de gestión de base de datos (SGBD) (En línea). Consultado, el 16 de Julio del 2011. Formato (HTML). Disponible en: <http://www.basesdedatos.org/>

Pérez, Á. 2008. Modelos de Desarrollo de Software (En línea). Consultado, el 16 de Julio del 2011. Formato (HTML). Disponible en: <http://www.mitecnologico.com/>

Pérez, D. 2011. ¿Qué son las bases de datos? (En línea). Consultado, el 16 de Julio del 2011. Formato (HTML). Disponible en: <http://www.maestrosdelweb.com/>

Riesco, D. s.f. Ejemplo de diagramas UML de objetos. (En línea). Consultado, el 16 de julio del 2011. Formato (PDF). Disponible en: www.sel.unsl.edu.ar

Rumbagh, J. 2007. Definición de diagrama de objetos. (En línea). Consultado, el 16 de julio del 2011. Formato (HTML). Disponible en: www.sparxsystem.com.ar

Sánchez, D. 2009. Microsoft SQL Server 2008 R2 (En línea). Consultado, el 16 de Julio del 2011. Formato (HTML). Disponible en: <http://www.computing.es/>

Sánchez, S; Chalmeta, R; Coltell, O; Manfor, P; Campos, C. 2003. Ingeniería de proyectos informáticos: Importancia de los sistemas informáticos. Publicaciones de la Universitat Jaume.4 Ed. España, Es. Graphic Group S.A. p 101.

Stair, R. s.f. Principios de sistemas informáticos: Modelo Relacional. Gloria Medina. 4 Ed. México. Thonsom Learning. p 205.

Tomás, M. 2007. Tipos de programación (En línea). Consultado, el 21 de julio del 2012. Formato (HTML). Disponible, en <http://www.elguille.info/>

Torres, L. 2008. SQL Server 2008, nuevas características (En línea). Consultado, el 16 de Julio del 2011. Formato (HTML). Disponible en: <http://www.gravitar.biz/>

Universidad de Sevilla. s.f. Ingeniería de software II: Diseño de la capa de datos. (En línea). Consultado, 18 de Julio del 2011. Formato (HTML). Disponible en: www.lsi.us.es/

Vergara, K. 2007. Concepto y tipos de software: programas, definición. (En línea). Consultado, el 15 de Julio del 2011. Formato (HTML). Disponible en: <http://www.bloginformatico.com/>

Zapata, M. 2006. Definición de diagrama de clases. (En línea). Consultado, el 15 de Julio del 2011. Formato (PDF). Disponible en: ocw.unizar.es

Zelkovitz, 1978. Ingeniería del Software (En línea). Consultado, el 15 de Julio del 2011. Formato (HTML). disponible en: <http://ingsoft.com/>

ANEXOS

ANEXO Nº 1

POSTULANTES DESARROLLANDO LA APLICACIÓN



ANEXO Nº 2

POSTULANTES, TUTOR Y DIRECTIVOS DE LA INSTITUCIÓN EN REUNIÓN

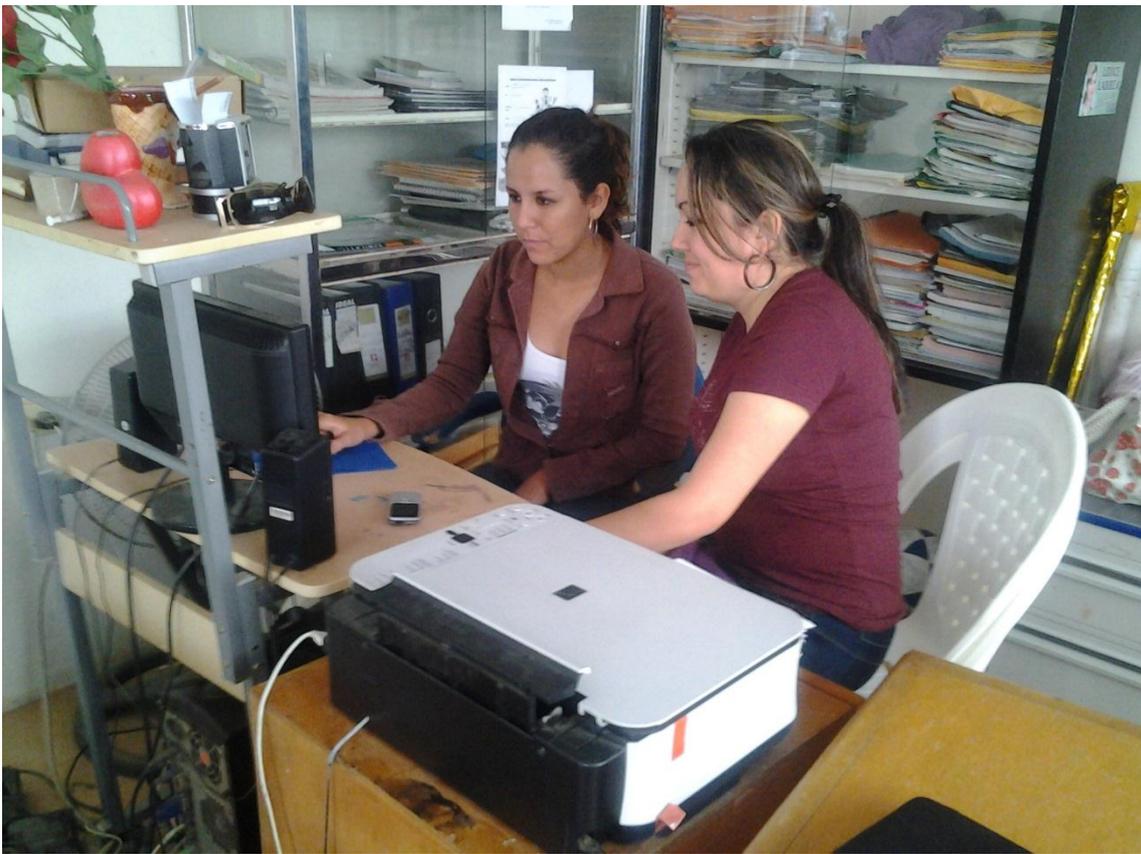


ANEXO Nº 3

VISITA IN SITU EN LA UNIDAD EDUCATIVA ARISTOS



ANEXO Nº 4
POSTULANTES EN LA ENTREGA Y CAPACITACIÓN DEL SISTEMA
INFORMÁTICO



ANEXO Nº 5
MANUAL DEL PROGRAMADOR

FUNCIÓN PARA VALIDAR CÉDULA

```
PublicFunctionvalidar_cedula(ByVal cedula As String) As Boolean
Dim i, sp, si, st, dv As Integer
Ifcedula.Length = 10 Then

For i = 1 To 9
If i Mod 2 = 0 Then
sp = sp + CInt(cedula(i - 1).ToString)
Else
IfCInt(cedula(i - 1).ToString) * 2 > 9 Then
si = si + ((CInt(cedula(i - 1).ToString) * 2) - 9)
Else
si = si + (CInt(cedula(i - 1).ToString) * 2)

EndIf
EndIf
Next
st = sp + si
IfstMod 10 > 0 Then
dv = 10 - (stMod 10)
Else
dv = 0
EndIf
IfNotcedula(9).ToString = dv Then
Return False
Else
Return True
EndIf
Else
Return False
EndIf

EndFunction
```

IMPORTAR LA CLASE SYSTEM.DATA.SQLCLIENT PARA PODER UTILIZAR LAS INSTRUCCIONES DE SQLCONNECTION Y SQLADAPTER

```
Imports System.Data.SqlClient
Public Class frm_matricula
    Dim connectionstring As String = "Data Source=(local);Initial
Catalog=SMCN;Integrated Security=yes"
    Dim conexion, conexion2 As New SqlConnection(connectionstring)
    Dim func As New funciones
    Dim op As String
    Dim datos, datos2, datos3, datos4, datos5 As New DataSet
    Dim base As New SqlDataAdapter("select matricula.id_matricula,
(persona.nombre + ' ' + persona.apellido)as nom,matricula.fecha_matricula,
aeb.aeb + ' ' + paralelo_aeb, periodo_lectivo.nombre_periodo from
matricula,aeb,alumno,persona,periodo_lectivo where
matricula.id_periodo=periodo_lectivo.id_periodo and
aeb.id_aeb=matricula.id_aeb and matricula.id_alumno=alumno.id_alumno and
alumno.id_persona=persona.id_persona and periodo_lectivo.estado_periodo=1
order by aeb.numero, aeb.paralelo_aeb asc", conexion)
    Dim base2 As New SqlDataAdapter("select * from periodo_lectivo where
estado_periodo='TRUE'", conexion)
    Dim base3 As New SqlDataAdapter("select id_aeb, (aeb + ' ' + paralelo_aeb)
as curso from aeb order by numero,paralelo_aeb", conexion)
    Dim base4 As New SqlDataAdapter("select alumno.id_alumno,
(persona.apellido + ' ' + persona.nombre)as nom from alumno, persona where
alumno.id_persona=persona.id_persona", conexion)
    Dim base5 As New SqlDataAdapter("select id_aeb, (aeb + ' ' + paralelo_aeb)
as curso from aeb order by numero,paralelo_aeb", conexion)
    Dim sw As Boolean
    Dim p, a As Integer
```

FUNCIÓN PARA AGREGAR LA MATRÍCULA A LA BASE DE DATOS.

```
Public Function insertar_matricula(ByVal id_matricula As Integer, ByVal nombre
As Integer, ByVal aeb As Integer, ByVal fecha As Date, ByVal periodo As
Integer) As Boolean
    Try
        Conectado()
```

```

    Dim cmd As New SqlCommand("insert into
matricula(id_matricula,id_alumno,id_aeb,fecha_matricula,id_periodo)values(@i
d_matricula,@id_alumno,@id_aeb,@fecha,@id_periodo)")
    With cmd
        .Parameters.Add(New SqlParameter("@id_matricula",
SqlDbType.Int)).Value = id_matricula
        .Parameters.Add(New SqlParameter("@id_alumno",
SqlDbType.Int)).Value = nombre
        .Parameters.Add(New SqlParameter("@id_aeb",
SqlDbType.Int)).Value = aeb
        .Parameters.Add(New SqlParameter("@fecha",
SqlDbType.Date)).Value = fecha
        .Parameters.Add(New SqlParameter("@id_periodo",
SqlDbType.Int)).Value = periodo
    End With
    cmd.Connection = cnn
    Dim dr As SqlDataReader
    dr = cmd.ExecuteReader
    Return True
Catch ex As Exception
    MsgBox(ex.Message)
    Return False
Finally
    Desconectado()
End Try
End Function

```

FUNCIÓN PARA MODIFICAR LA MATRÍCULA EN LA BASE DE DATOS.

```

Public Function editar_matricula(ByVal id_matricula As Integer, ByVal
nombre As Integer, ByVal aeb As Integer, ByVal fecha As Date, ByVal periodo
As Integer) As Boolean
    Conectado()
    Dim cmd As New SqlCommand("Update matricula set
id_alumno=@alumno,id_aeb=@aeb,id_periodo=@periodo,fecha_matricula=@f
echa where id_matricula=" & id_matricula & """)
    With cmd
        .Parameters.Add(New SqlParameter("@alumno", SqlDbType.Int)).Value
= nombre
        .Parameters.Add(New SqlParameter("@aeb", SqlDbType.Int)).Value =
aeb

```

```

        .Parameters.Add(New SqlParameter("@periodo", SqlDbType.Int)).Value
= periodo
        .Parameters.Add(New SqlParameter("@fecha",
SqlDbType.Date)).Value = fecha
    End With
    cmd.Connection = cnn
    Try
        Dim dr As SqlDataReader
        dr = cmd.ExecuteReader
        Return True
    Catch ex As Exception
        MsgBox(ex.Message)
        Return False
    Finally
        Desconectado()
    End Try
End Function

```

FUNCIÓN PARA ELIMINAR LA MATRÍCULA EN LA BASE DE DATOS.

```

Public Function eliminar_matricula(ByVal id_matricula As String) As Boolean
    Try
        Conectado()
        cmd = New SqlCommand("Delete from matricula where id_matricula="
& id_matricula & """)
        cmd.CommandType = CommandType.Text
        cmd.Connection = cnn
        cmd.ExecuteNonQuery()
        MessageBox.Show("La matrícula fue eliminada con exito.",
"Eliminado..", MessageBoxButtons.OK, MessageBoxIcon.Information)
        Return True
    Catch ex As Exception
        MessageBox.Show("El periodo se encuantra en uso, no se puede
eliminar.", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Exclamation)
        Return False
    Finally
        Desconectado()
    End Try
End Function

```

CLASE CONEXIÓN CON LA BASE DE DATOS

```
Imports System.Data.SqlClient
Imports System.Data
Public Class conexion
    Protected cnn As New SqlConnection
    Protected Function Conectado()
        Try
            cnn = New SqlConnection("Data Source=(local);Initial
Catalog=SMCN;Integrated Security=yes")
            cnn.Open()
            Return True
        Catch ex As Exception
            MsgBox(ex.Message)
            Return False
        End Try
    End Function
    Protected Function Desconectado()
        Try
            If cnn.State = ConnectionState.Open Then
                cnn.Close()
                Return True
            Else
                Return False
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
            Return False
        End Try
    End Function
End Class
```

ANEXO Nº 6
MANUAL DEL USUARIO



Manual del Usuario

SISTEMA INFORMÁTICO DE
REGISTRO DE MATRICULACIÓN Y
CALIFICACIONES

INDICE

1. GUÍA DE USUARIO	112
2. INTRODUCCIÓN	115
3. INTRODUCCION AL SISTEMA	115
3.1. REQUERIMIENTOS	116
4. DESCRIPCIÓN DETALLADA DEL SISTEMA	116
5. ACCESO AL SISTEMA	116
6. COMPONENTES DE LA PANTALLA	118
6.1. MENÚ PRINCIPAL	118
6.2. CONFIGURACION DE REGISTROS	119
6.3. MENÚ MATRICULACIÓN	119
6.4. MENÚ CALIFICACIONES	120
6.5. MENÚ REPORTES	120
6.6. MENÚ HERRAMIENTAS.....	121
6.7. MENÚ AYUDA.....	122
6.8. MENÚ SALIR	122
7. MENÚ LATERAL DE ACCESO RÁPIDO.....	122
8. FORMULARIO DE ALUMNOS.	125
8.1. INGRESO DE UN NUEVO ALUMNO	126
8.2. GUARDAR DATOS DE UN NUEVO ALUMNO	129
8.3. EDITAR DATOS DE UN ESTUDIANTE.....	130
8.4. ELIMINAR DATOS DE UN ESTUDIANTE	131
9. FORMULARIO DE REPRESENTANTES.	131
9.1. INGRESO DE UN NUEVO REPRESENTANTE	131
9.2. GUARDAR, EDITAR Y ELIMINAR UN REPRESENTANTE	132
10.FORMULARIO DE DOCENTES.	132
10.1. INGRESO DE UN NUEVO DOCENTE	132
10.2. GUARDAR, EDITAR Y ELIMINAR UN DOCENTE	133
11.FORMULARIO DE REGISTRO DE MATRICULACIÓN	133
12.FORMULARIO DE INGRESO DE CALIFICACIONES	135
13.FORMULARIO DISTRIBUCIÓN	137
14.FORMULARIO PERIODO LECTIVO.	138
15.FORMULARIO DE ASIGNATURAS	140
16.FORMULARIO AÑO DE EDUCACIÓN BÁSICA	142
17.FORMULARIO QUIMESTRE	144
18.FORMULARIO BIMESTRE.....	145
19.OPCIÓN DE BÚSQUEDAS.	147
20.CUADRO EXPLICATIVO DE LOS BOTONES	148
21. REPORTES	1490

1. GUÍA DE USUARIO

A continuación se describe la guía de usuario, que nos indica los pasos ordenados a seguir para manejar adecuadamente la información, por lo tanto la presente ayudaría en gran porcentaje a la persona encargada de administrar el sistema.

La aplicación manejará principalmente los procesos de registro de matriculación y de calificaciones por lo que es de gran importancia contar con la información previa, para que sea manejada por el sistema.

La aplicación informática posee formularios, los cuales deben ser llenados ordenadamente para tener los resultados esperados.

PASOS:

- Lo primero que debe hacerse es crear la cuenta administrador, ingresando un nombre de usuario y una contraseña, con el firme propósito de que no todo individuo pueda tener acceso al sistema
- Ya realizada la acción anterior, el administrador debe dirigirse a la opción periodo lectivo e ingresar los datos solicitados en ese formulario.
- Seguido a esto, debe abrir el formulario quimestre e ingresar textualmente “primer quimestre”, asignarle el numero uno y guardar. Una vez realizado este paso, en la misma ventana añadimos un “segundo quimestre”, le asignamos el numero dos y guardamos.

- Acción similar realizamos en el formulario bimestre, con la diferencia que en este agregamos lo siguiente: primer, segundo, tercero y cuarto bimestre, asignando los números del uno al cuatro dependiendo del orden de cada bimestre.
- Contando con estos datos almacenados, nos dirigimos al formulario año de educación básica (AEB), para introducir los respectivos cursos con sus paralelos y en el nivel correspondiente. Ejemplos: (PRIMERO A, nivel BÁSICO).

Se cuenta con 3 niveles de educación:

- Pre básica: comprende los años de educación iniciales: NIVEL A Y NIVEL B.
- Básico: Desde PRIMERO a SÉPTIMO
- Secundario: OCTAVO, NOVENO Y DÉCIMO.

Para que la información se muestre ordenada en el datagrid, se incluyó en el formulario un campo denominado número, el mismo contiene los números desde el 1 hasta el 10. Estos se les asignan a cada año de educación básica dependiendo de su orden. Ejemplo: si es Pre básica se le asigna el 1, a básico el 2 y secundario el 3. De esta manera se presentarán ordenados por nivel.

- El siguiente paso es ingresar las asignaturas que se impartirán en el periodo lectivo actual, las mismas que se registrarán de acuerdo al nivel. Ejemplo:
 - Asignaturas de pre básica: Matemáticas
 - Asignaturas de básico : Matemáticas 1
 - Asignaturas de secundario: Matemáticas 2

Al momento de almacenar las asignaturas, procurar que se les asigne el adecuado nivel, basándose en la explicación anterior.

- Seguidamente debemos dirigirnos al formulario de docentes, ingresar la información correctamente y luego almacenarla.
- A continuación abrimos el formulario alumnos e ingresamos los datos correspondientes. El botón que se denomina activo, lo utilizamos cuando un alumno está activo en la institución, en caso que el alumno se retire, debemos dar clic en inactivo. En esta misma ventana, en la pestaña datos de representante, encontraremos un botón de agregar, el mismo que nos llevará al formulario de representantes, para ingresar los datos de éstos en caso de no tener registrado uno.
- Realizada esta acción, procedemos a enlazarnos con el formulario matriculación. Aquí podremos matricular a los alumnos en el periodo correspondiente, para esto debemos llenar los datos correctamente. El switch botón que contiene aprobado o reprobado, se habilitará solo cuando el alumno sea matriculado más de una vez en la unidad educativa. Basándose en el promedio que obtenga el estudiante en el final del periodo lectivo, este botón se pondrá automáticamente aprobado o reprobado. Si se activa reprobado, al momento de matricularse solo le aparecerá el año de educación básica en el que se encontraba. Si se activa aprobado se podrá matricular en el año de educación básica superior al que se encontraba.
- Es importante distribuir las asignaturas con su respectivo docente, para realizar este proceso debemos abrir el formulario distribución.
- Para validar las calificaciones nos dirigimos al menú Herramientas y seleccionamos personalizar y en la pestaña calificaciones procedemos a

detallar el calificación mínima y máxima que permitirá el sistema, así mismo se puntualizaran los valores que determinaran cuando un alumno esta aprobado, cuando esta en supletorio y cuando esta reprobado. Por ultimo, ya contando con todos los datos necesarios, procedemos a llenar las calificaciones en el formulario que lleva este nombre. Las calificaciones se ingresan por quimestre con sus respectivos bimestres. No se podrá ingresar las calificaciones en un quimestre o bimestre superior, sin antes tener llenos todos los datos del quimestre o bimestre inferior. Al guardar debemos estar completamente seguros que las calificaciones estén correctas, debido a que una vez almacenadas, éstas no podrán modificarse bajo ningún concepto.

2. INTRODUCCIÓN

A continuación se presenta un manual, que le permitirá al usuario utilizar el sistema computarizado de matriculación y control de calificaciones de la unidad educativa Aristos.

Se muestra cada una de las opciones que este sistema contiene, así como también la descripción de las funciones que desempeñan cada una de ellas, se detalla claramente los pasos que debe seguir el usuario a fin de realizar cada uno de los procesos de la mejor manera. De igual modo se presentan las pantallas y mensajes que pudieran producirse, según los procesos que se desean llevar a cabo.

3. INTRODUCCION AL SISTEMA

En este apartado se detallaran los requerimientos establecidos de hardware y software, para el correcto funcionamiento del sistema. Además se detallan los elementos del entorno con los que el usuario debe relacionarse para el manejo del sistema.

3.1. REQUERIMIENTOS

- Sistema operativo: XP o superior
- Tipo de procesador: Pentium IV en adelante
- Tamaño de disco duro: 160 Gb de espacio inicial o Superior
- Memoria RAM: 1 GB o Superior
- Monitor: VGA o SVGA
- Unidad de CD: DVD - ROM
- Resolución de pantalla: mínimo 1024 x 768

4. DESCRIPCIÓN DETALLADA DEL SISTEMA

En este manual tenemos los pasos a seguir en el sistema para un mejor y adecuado manejo y de esta manera evitar problemas en el funcionamiento del mismo que a continuación detallamos.

5. ACCESO AL SISTEMA

Los pasos que debe realizar el usuario administrador del software, son los siguientes:

1. En el escritorio se muestra el icono de acceso directo llamado SMCC Aristos:



Seguidamente damos doble clic en el icono nombrado en el paso anterior.

Para tener acceso al sistema, el usuario deberá saber la clave o contraseña del mismo. Esta se le pedirá al iniciar el sistema y le aparecerá una figura similar a la siguiente.

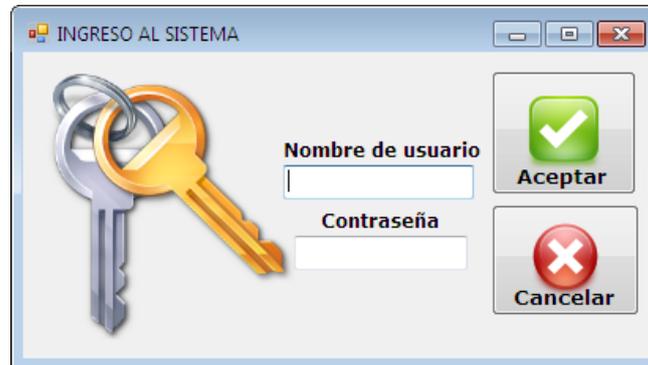


Figura 1: Ingreso al sistema

En caso que la contraseña no sea correcta se presentará un mensaje de error.



Figura 2: Mensaje error de login

2. Si se digitó correctamente la contraseña, le aparecerá la pantalla que indica que el sistema está cargando



Figura 3: Ventana de carga del sistema

3. Seguidamente aparecerá el Menú Principal.



Figura 4: Pantalla principal del sistema

6. COMPONENTES DE LA PANTALLA

6.1. MENÚ PRINCIPAL

El menú principal contiene las opciones:

- Configuración de registros
- Matriculación
- Calificaciones
- Reportes
- Herramientas
- Salir
- Ayuda

Para tener acceso solo dé clic en el menú que desee y rápidamente se desplegarán las opciones que contiene cada uno de ellos

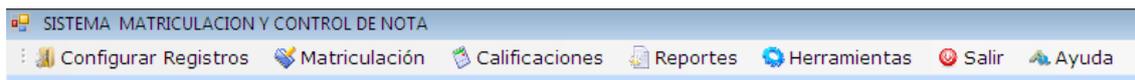


Figura 5: Menú principal del sistema

6.2. CONFIGURACION DE REGISTROS

Al dar clic en está opción nos permitirá acceder a las opciones: periodo lectivo, asignaturas, provincias, docentes, AEB, quimestres, bimestres y usuarios.



Figura 6: Menú configurar registros

6.3. MENÚ MATRICULACIÓN

Se encuentra en el Menú Principal del sistema, dentro de ella se hallan las opciones de registro y registro de matrícula.

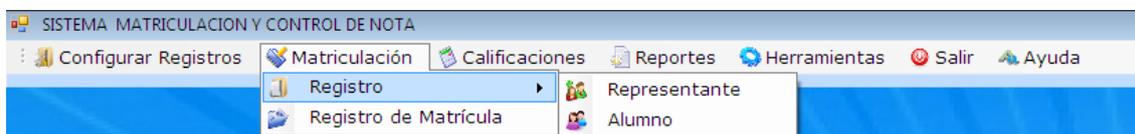


Figura 7: Menú matriculación desplegado

a) OPCIÓN REGISTRO

Al seleccionar esta opción, tendremos acceso a ingresar al formulario de representantes o de alumnos

b) OPCIÓN DE REGISTRO DE MATRÍCULA

Permite realizar el registro de matriculación de un estudiante.

6.4. MENÚ CALIFICACIONES

Esta es la cuarta opción del Menú Principal aquí encontramos opciones de registro de calificaciones.



Figura 8: Menú calificaciones

a) INGRESO DE CALIFICACIONES

Esta opción sirve para ingresar las calificaciones de los estudiantes.

b) REPORTE DE CALIFICACIONES

Al dar clic nos permite visualizar el reporte de calificaciones.

6.5. MENÚ REPORTE

La quinta opción del menú donde nos permitirá acceder a todos los reportes generados por el sistema con dar clic en uno de ellos.

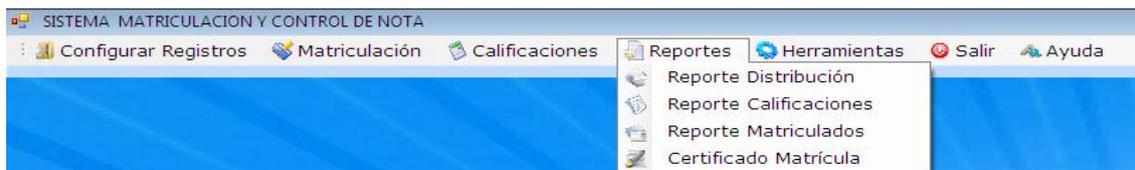


Figura 9: Menú reportes

6.6. MENÚ HERRAMIENTAS

Lo encontraremos en la sexta opción del menú, donde podemos elegir cualquiera de las siguientes opciones:

- a) Base de datos
- b) Personalizar

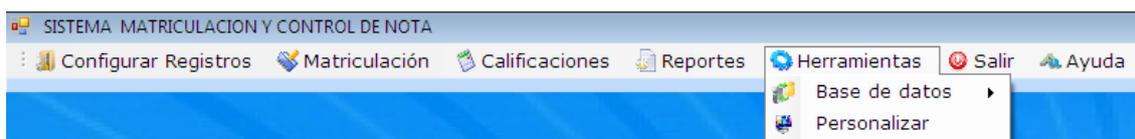


Figura 10: Menú herramientas

a) OPCIÓN BASE DE DATOS

Al dar clic se despliegan dos opciones: realizar copia de seguridad y restaurar: al dar clic en la opción copia de seguridad se va a crear un respaldo de la base de datos del sistema.

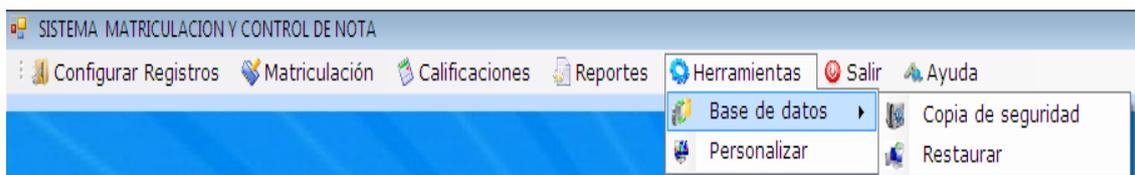


Figura 11: Opción base de datos

b) OPCIÓN PERSONALIZAR

Nos permitirá realizar cambios de apariencia del sistema y también configurar las calificaciones dependiendo de los cambios que se generen.



6.7. MENÚ AYUDA

Al dar clic en esta opción nos mostrara un documento con la guía y el manual del usuario (Ver figura 5).

6.8. MENÚ SALIR

Permite Salir del sistema (Ver figura 5)

7. MENÚ LATERAL DE ACCESO RÁPIDO

En este menú encontramos las siguientes opciones: configurar registros, alumnos, matriculación, docente, calificaciones y distributivo.

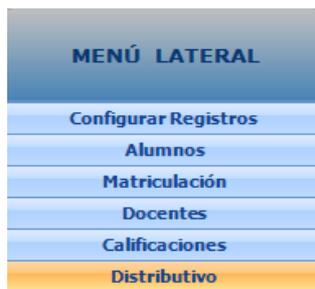


Figura 12: Menú lateral

- Al dar clic en la ficha configurar registros nos aparecerán los iconos de acceso rápido del periodo lectivo, año de educación básica, asignatura y provincia.



Figura 13: Menú lateral - institución

- Al dar clic en la ficha alumnos nos aparecerán los iconos de acceso rápido de alumnos y representantes.



Figura 14: Menú lateral - alumnos

- Al dar clic en la ficha matriculación nos aparecerán los iconos de acceso rápido de registro de matrícula, reporte matriculados y certificados de matrícula.



Figura 15: Menú lateral - matriculación

- Al dar clic en la ficha docente nos aparecerán los iconos de acceso rápido de docente



Figura 16: Menú lateral - docentes

- Al dar clic en la ficha calificaciones nos aparecerán los iconos de acceso rápido de ingreso calificaciones y reporte de calificaciones.



Figura 17: Menú lateral - calificaciones

- Al dar clic en la ficha distributivo nos aparecerán los iconos de acceso rápido de distribución, reporte distribución y detalle distribución.



Figura 18: Menú lateral - distributivo

8. FORMULARIO DE ALUMNOS.

Este formulario contiene toda la información respectiva acerca de los estudiantes.

8.1. INGRESO DE UN NUEVO ALUMNO

El primer paso es abrir el formulario alumno (ver figura 6) o (ver figura 14)

The screenshot shows a web application window titled 'Alumnos'. At the top, there is a search bar with 'Buscar por:' and a dropdown menu for 'Periodo Lectivo' set to 'PERIODO LECTIVO 2012-2013'. Below this is a table listing several students with columns for CEDULA, NOMBRES, GENERO, NACIONALIDAD, DIRECCIÓN, and FECHA NACIMIENTO. The first student is DOMENICA CECIBEL CARRERRA SOLIS.

Below the table, there are tabs for 'Datos Personales', 'Datos Vivienda', and 'Datos Representante'. The 'Datos Personales' tab is active, showing a form with the following fields:

- Cedula *: 1312536483 (with an 'ACTIVO' button)
- Nombres *: DOMENICA CECIBEL
- Apellidos *: CARRERRA SOLIS
- Genero *: FEMENINO
- Colegio Procedencia: MARIANITA DE JESUS
- Fecha de Nacimiento *: domingo, 12 de febrero de 1888

There is a photo of the student and buttons for 'Agregar' and 'Quitar'. At the bottom of the form are buttons for 'Nuevo', 'Guardar', 'Editar', 'Eliminar', and 'Cancelar'.

Figura 19: Formulario alumnos

Para ingresar un nuevo estudiante damos clic en el botón nuevo y se habilitara el campo cédula. Ingresamos un numero de cedula y presionamos (enter) o la tecla (tab). Si la cedula ingresada no es correcta, aparecerá el siguiente mensaje de error:

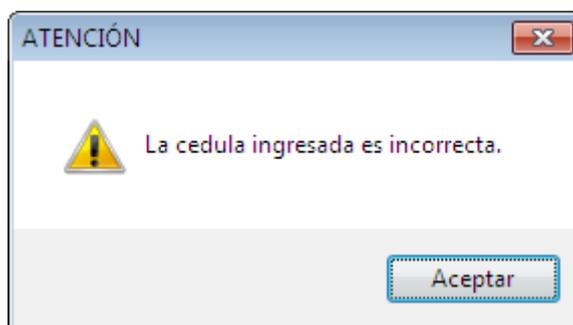


Figura 20: Mensaje de error de cedula incorrecta

La cedula del alumno solo se podrá ingresar una vez por periodo lectivo, si se lo intenta registrar nuevamente aparecerá el siguiente mensaje de error:

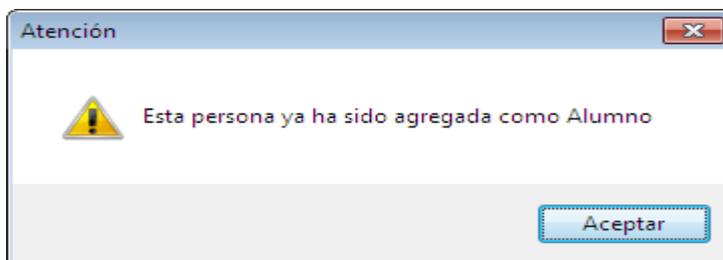


Figura 21: Mensaje de error alumno ya ingresado.

Si se ingresa un número de cedula que pertenece a un docente o un representante ya almacenado en la base de datos, se mostrara el siguiente mensaje de error:

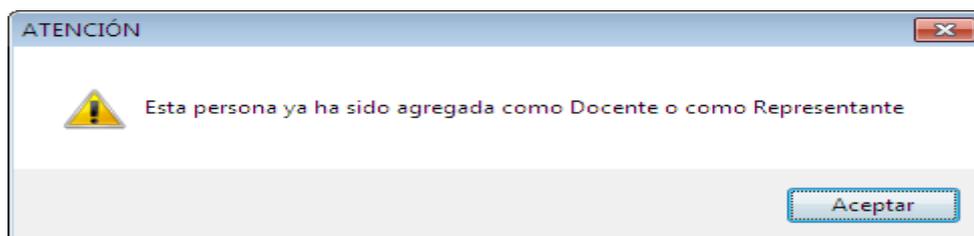


Figura 22: Mensaje de error. Cedula de docente o representante

Si la cedula ingresada no presenta ninguno de los errores anteriormente nombrados, se activaran todos los controles para registrar los datos del estudiante.

Figura 23: Ingreso de datos personales

Una vez ingresados correctamente los datos personales del estudiante se habilita la pestaña datos de vivienda.

Datos Personales Datos Vivienda Datos Representante

DATOS DE VIVIENDA DE MARTHA CRUZ

Dirección *
BARRIO NORTE

Nacionalidad * ECUATORIANA

Provincia * SANTA ELENA

Cantón * SALINAS

Parroquia * GRAL. ALBERTO ENRIQU

Nuevo Guardar Editar Eliminar Cancelar

Estado

Figura 24: Ingreso de datos de vivienda

Después de llenar la información de la vivienda se habilitara la pestaña datos representantes.

Datos Personales Datos Vivienda Datos Representante

REPRESENTANTE DE MARTHA CRUZ

Representante * CEDENO WINTER Agregar

Nuevo Guardar Editar Eliminar Cancelar

Estado

Figura 25: Ingreso datos del representante

En caso de no tener un representante ingresado, tenemos el botón agregar que nos llevará a el formulario representante, en el mismo podremos ingresar un nuevo representante.

The screenshot shows a web form titled 'AGREGAR REPRESENTANTE' with two tabs: 'Datos Personales' (selected) and 'Datos Vivienda'. Under 'DATOS PERSONALES', there are several input fields: 'Cedula *' with a placeholder 'Ingrese cedula.', 'Nombres *', 'Apellidos *', 'Genero *' (a dropdown menu), 'Teléfono', and 'Fecha de Nacimiento *' (a date picker set to 'lunes, 27 de agosto de 2012'). To the right of these fields is a placeholder for a photo, showing a camera icon with a red 'X' and the text 'Sin Foto'. At the bottom right of the form are buttons for 'Agregar' and 'Quitar'. At the bottom of the entire window are buttons for 'Guardar' and 'Cancelar'.

Figura 26: Agregar un representante desde formulario alumno

8.2. GUARDAR DATOS DE UN NUEVO ALUMNO

Si un campo que está estipulado como obligatorio (*) se encuentra vacío y presionamos el botón guardar, nos aparecerá el siguiente mensaje de error:

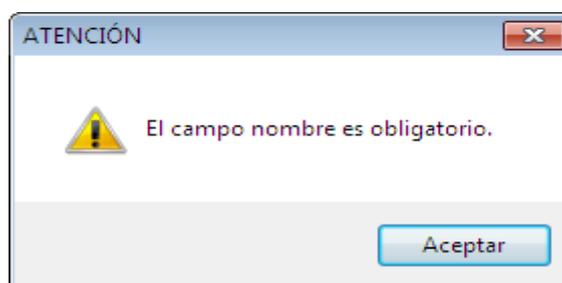


Figura 27: Mensaje de error. Campos obligatorios

Al contar con todos los datos ingresados correctamente procedemos a almacenar la información, y se visualizará el siguiente cuadro:

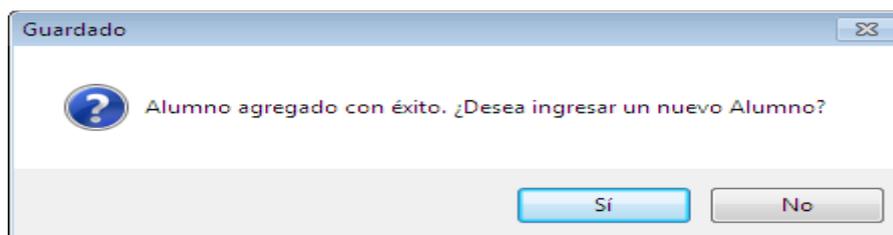


Figura 28: Guardado con éxito

Si presionamos no se cancela la acción de guardar

Si Presionamos si los datos del alumno se podrán visualizar en el siguiente datagrid:

CEDULA	NOMBRES	GENERO	NACIONALIDAD	DIRECCIÓN	FECHA NACIMIENTO
1312536483	CARRERRA SOLIS DOMENICA CECIBEL	FEMENINO	ECUATORIANA	LOS CEIBOS	12/02/1888
1313136432	CRUZ MARTHA	MASCULINO	ECUATORIANA	BARRIO NORTE	22/08/1989
1310621964	GARCIA MENENDEZ VIVIANA CLARIBEL	FEMENINO	ECUATORIANA	LAS VILLAS	04/05/1992
1312850496	MENDOZA MARIANO	MASCULINO	ECUATORIANA	CANUTO	22/08/1985
1313136424	MUÑOZ JUAN	MASCULINO	ECUATORIANA	CALLE CHILE Y JOSE MARIA HUERTA	22/08/1989

Figura 29: Datos almacenados en el datagrid

8.3. EDITAR DATOS DE UN ESTUDIANTE

Para editar un alumno ya ingresado, debemos seleccionarlo y presionar el botón editar. Seguido realizamos los cambios que se crean pertinentes y presionamos guardar.

CEDULA	NOMBRES	GENERO	NACIONALIDAD	DIRECCIÓN	FECHA NACIMIENTO
1312536483	CARRERRA SOLIS DOMENICA CECIBEL	FEMENINO	ECUATORIANA	LOS CEIBOS	12/02/1888
1313136432	CRUZ MARTHA	MASCULINO	ECUATORIANA	BARRIO NORTE	22/08/1989
1310621964	GARCIA MENENDEZ VIVIANA CLARIBEL	FEMENINO	ECUATORIANA	LAS VILLAS	04/05/1992
1313136473	LOPEZ PEDRO	MASCULINO	ECUATORIANA	LA CARINA	22/08/1978
1312850496	MENDOZA MARIANO	MASCULINO	ECUATORIANA	CANUTO	22/08/1985
1313136424	MUÑOZ JUAN	MASCULINO	ECUATORIANA	CALLE CHILE Y JOSE MARIA HUERTA	22/08/1989

Datos Personales

Cedula * ACTIVO

Nombres *

Apellidos *

Genero *

Colegio Procedencia

Fecha de Nacimiento *



Agregar Quitar

Figura 30: Editar un alumno almacenado

8.4. ELIMINAR DATOS DE UN ESTUDIANTE

Para eliminar datos de un estudiante ya ingresado, debemos seleccionarlo y presionar el botón eliminar y dar clic en si.

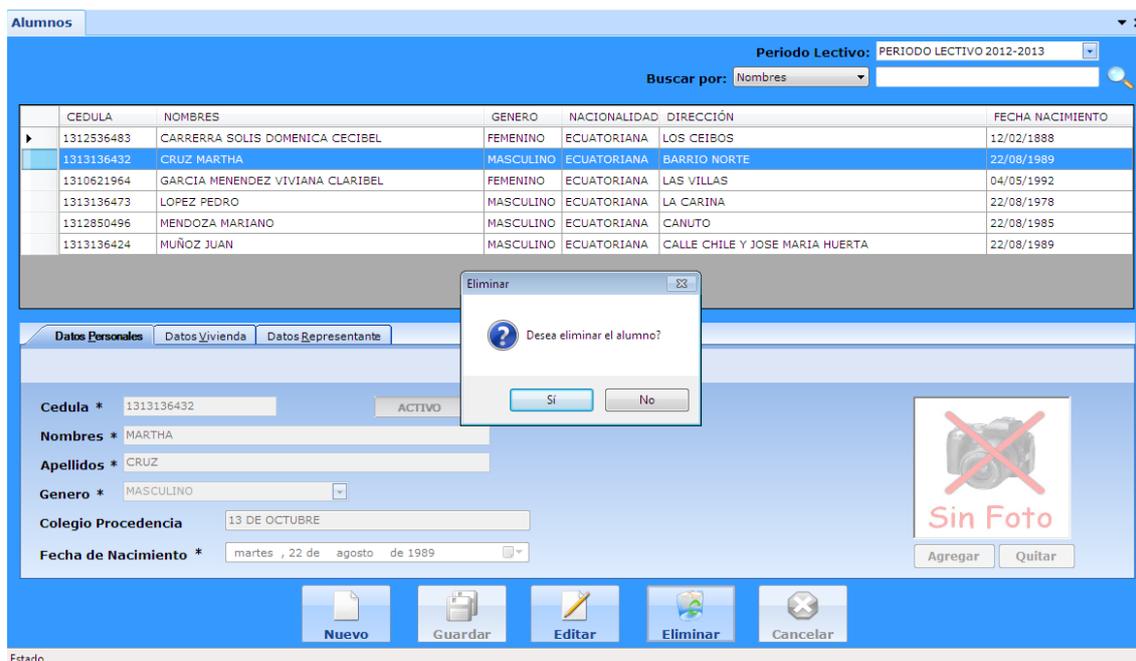


Figura 31: Eliminar un alumno ingresado

9. FORMULARIO DE REPRESENTANTES.

Este formulario contiene toda la información respectiva acerca de los representantes.

9.1. INGRESO DE UN NUEVO REPRESENTANTE

El primer paso es abrir el formulario representante (ver figura 6) o (ver figura 14)

The screenshot displays a web application window titled 'Representante'. At the top, there is a search bar labeled 'Buscar por:' with a dropdown menu and a search icon. Below the search bar is a table with the following data:

	CEDULA	NOMBRES	GENERO	TELÉFONO	DIRECCIÓN	NACIONALIDAD
▶	1309561692	CEDEÑO WINTER	MASCULINO	085674276	LAS VILLAS	ECUATORIANA
	1301354906	VITERI MARIANA	FEMENINO	086554253	LOMA SECA	ECUATORIANA
	1310476567	BRAVO JOSE	MASCULINO	059683463	SAN BARTOLO	ECUATORIANA
	1311419210	MENDOZA RAUL	MASCULINO	098756363	LAS MERCEDES	ECUATORIANA
	1307123933	MACIAS ANGELICA	FEMENINO	096637278	LAS MERCEDES	ECUATORIANA

Below the table, there are two tabs: 'Datos Personales' (selected) and 'Datos Vivienda'. The 'Datos Personales' section is titled 'DATOS PERSONALES' and contains the following fields:

- Cedula *: 1309561692
- Nombre *: WINTER
- Apellido *: CEDEÑO
- Genero *: MASCULINO

There is also a placeholder for a profile picture, which is currently blank and marked with a red 'X' and the text 'Sin Foto'.

Figura 32: Formulario representante

9.2. GUARDAR, EDITAR Y ELIMINAR UN REPRESENTANTE

Para guardar, editar o eliminar un representante, se efectúan los mismos pasos realizados con el formulario alumnos. Ya que los procesos se los realiza de igual manera. (Ver pagina 12-16)

10. FORMULARIO DE DOCENTES.

Este formulario contiene toda la información respectiva acerca de los docentes.

10.1. INGRESO DE UN NUEVO DOCENTE

El primer paso es abrir el formulario docentes (ver figura 10) o (ver figura 16)

1. Para crear una nueva matrícula damos clic en el botón nuevo.
2. Luego seleccionamos el periodo lectivo, que por defecto queda activado el que esta en vigencia.
3. Seguido a esto seleccionamos la fecha en que se realiza la matriculación
4. Después seleccionamos el alumno que se desea matricular.
5. Para buscar al alumno de una forma mas rápida, damos clic en la lupa y asignamos el estudiante deseado.
6. El control que contiene si un estudiante esta o no aprobado, se activará cuando un alumno ya ha tenido un historial en la institución. Si esta aprobado (recoge el resultado del promedio final) automáticamente aparece el año de educación básica superior, caso contrario solo aparece el mismo año de educación.
7. Procedemos a seleccionar el nivel (pre básico, básico, secundario)
8. Escogemos el año de educación básica.
9. Una vez efectuado los pasos anteriormente señalados, damos clic en el botón guardar para almacenar la información en la base de datos.
10. Si se desea editar una matrícula, damos clic en el botón editar y guardamos el cambio.
11. Si se desea eliminar una matrícula damos clic en el botón eliminar.
12. Si desea suspender una actividad iniciada en este formulario, de clic en cancelar.

MENSAJES DE ERROR:

Si ubicamos una fecha superior a la fecha que se realiza la matrícula, se presentará el siguiente mensaje:

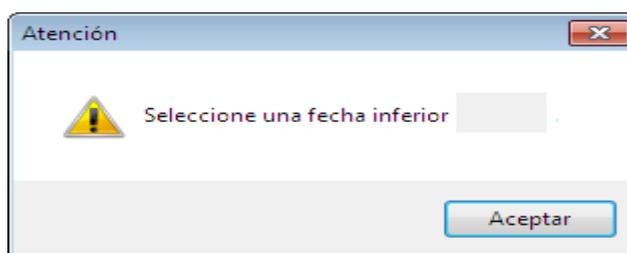


Figura 35: Error en fecha de matriculación

Si se desea eliminar la matrícula de un alumno que contiene calificaciones, saldrá un mensaje de error que indica que ese alumno se encuentra en uso y por lo tanto no puede ser eliminado.

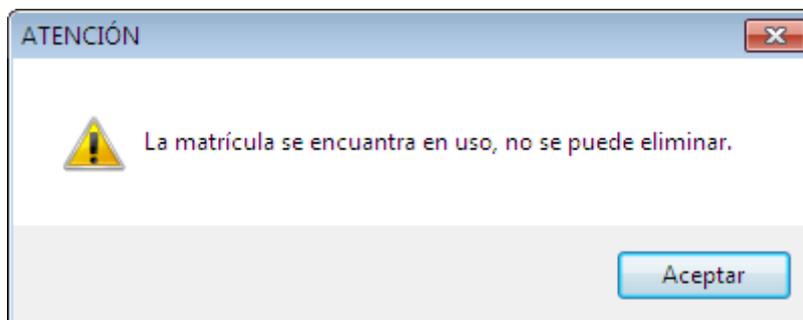


Figura 36: Error: matrícula en uso

12. FORMULARIO DE INGRESO DE CALIFICACIONES

Para tener el registro de las matrículas, lo primero que debe hacerse, es abrir el formulario de registro de calificaciones (ver figura 7) o (ver figura 17)

APellidos y Nombres	Apro	Conc	Prec	Argu	CPA	SUMA	DISC	INGRESADO
CARRERRA SOLIS DOMENICA CECIBEL	2,00	2,00	2,00	2,00	2,00	10,00	9,00	<input checked="" type="checkbox"/>
CRUZ MARTHA	2,00	2,00	2,00	2,00	2,00	10,00	2,00	<input checked="" type="checkbox"/>

Figura 37: Formulario de ingreso de calificaciones

1. Una vez que aparezca la ventana, procedemos a seleccionar el periodo lectivo, que por defecto queda activado el ultimo que esta en uso.
2. Seguido seleccionamos el nivel (pre básica, básico, secundario).
3. Realizado esto escogemos el año de educación básica al que pertenece el alumno.
4. Después escogemos el docente y
5. Seleccionamos la materia.
6. Completa toda esta información, nos dirigimos a la pestaña quimestre, donde procedemos a llenar las calificaciones por quimestre en sus respectivos bimestres.
7. Procedemos a guardar las calificaciones dando clic en el botón guardar. Es importante saber que una vez guardada las calificaciones, estas no podrán ser editadas.
8. Cuando las notas se encuentran llenas en su totalidad, en la pestaña total, aparece el promedio automáticamente, la disciplina y un mensaje que indica el estado del alumno: Aprobado o Reprobado.

MENSAJES DE ERRORES:

Si pretende guardar una calificación con un parámetro vacío, le aparecerá el siguiente mensaje de error:

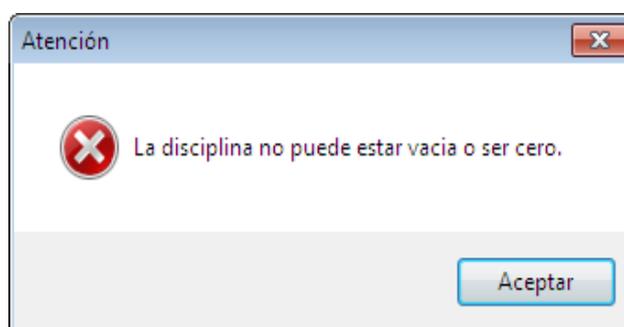
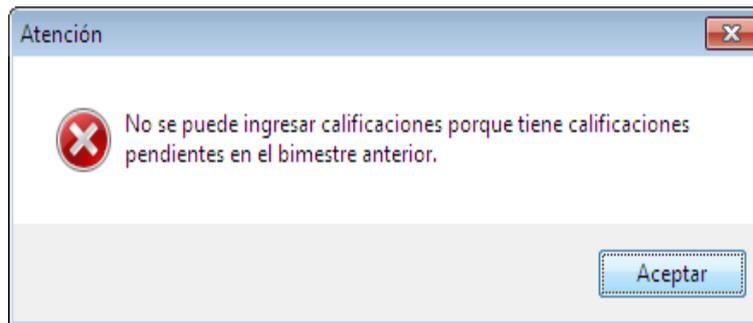


Figura 38: Mensaje error: campo de calificación esta vacío

Si tiene calificaciones pendientes en un quimestre o bimestre, no podrá ingresar las calificaciones en el quimestre o bimestre superior.



13. FORMULARIO DISTRIBUCIÓN

Permite distribuirle a cada profesor las asignaturas correspondientes en su debido año de educación básica en el periodo lectivo activo.

Lo primero que debemos hacer, es abrir el formulario Distribución. (Ver figura18)

Figura 39: Figura de distributivo

1. Cuando nos aparece la ventana de distribución, debemos seleccionar el periodo lectivo, que por defecto queda activado el ultimo que esta en uso.
2. Seguido seleccionamos el nivel (pre básica, básico, secundario).
3. Realizado esto escogemos el año de educación básica.
4. Después escogemos el docente

Procedemos a asignar las materias. Para ello contamos con los siguientes botones:

BOTONES:

5. Agregar: Permite asignar una materia a la vez.
6. Agregar todo: Permite asignar todas las materias.
7. Quitar: Quita una materia a la vez.
8. Quitar todo: Quita todas las materias.

Finalmente, cuando ya estén asignadas las materias a cada curso con su respectivo docente, están se almacenan automáticamente en la base de datos (no hay la necesidad de contar con un botón guardar, para almacenar la información).

14. FORMULARIO PERIODO LECTIVO.

Permite establecer el periodo lectivo que cursa la institución.

Lo primero que debemos hacer, es abrir el formulario Periodo lectivo. (Ver figura10) o (Ver figura 13)

Figura 40: Formulario periodo lectivo.

1. Una vez visualizada la ventana de periodo lectivo, hacemos clic en el botón nuevo
2. Definimos la fecha de inicio del periodo lectivo
3. Seguido definimos la fecha fin del periodo lectivo.
4. Establecidas las dos fechas, automáticamente se ingresa el nombre del periodo lectivo, tomando en cuenta las fechas con los rangos anteriormente ubicados.
5. Seleccionamos el botón activo, para que solo se encuentre activo el periodo vigente
6. Procedemos a guardar.
7. Si se desea editar un periodo lectivo
8. Eliminar un periodo lectivo
9. Cancelar una acción.

MENSAJES DE ERROR

Si se pretende eliminar un periodo lectivo que esta en uso, aparecerá el siguiente mensaje de error:

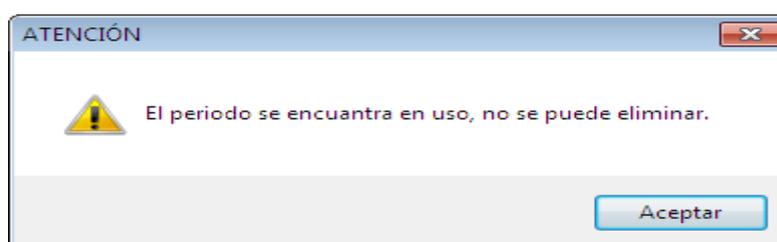


Figura 41: Mensaje error: No se puede eliminar periodo lectivo

Si se pretende ingresar un nuevo periodo lectivo con fechas ya ingresadas, aparecerá:

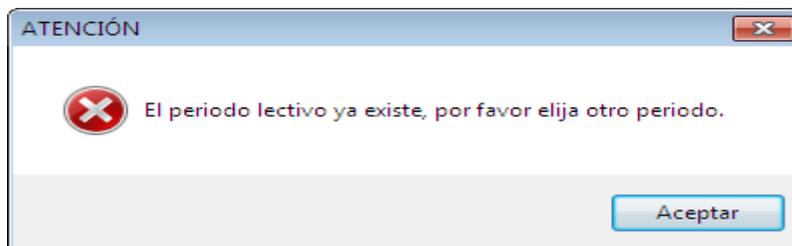


Figura 42: Error: Periodo lectivo ya existe

15. FORMULARIO DE ASIGNATURAS

Permite ingresar y almacenar las asignaturas que se imparten en cada año de educación básica. .

Lo primero que debemos hacer, es abrir el formulario de asignaturas. (Ver figura10) o (Ver figura 13)

ASIGNATURA	OPTATIVA	NIVEL	CARGA HORARIA
LOGICA MATEMATICA	NO	PREBÁSICA	4
MATEMÁTICA	NO	PREBÁSICA	5
LENGUA	NO	PREBÁSICA	3
CUENTO	NO	PREBÁSICA	8
COMPUTACIÓN	SI	PREBÁSICA	6
EDUCACIÓN SEXUAL	SI	PREBÁSICA	4
CIENCIAS SOCIALES	NO	PREBÁSICA	3
CULTURA FISICA	SI	PREBÁSICA	2
CULTURA FISICA 1	SI	BÁSICO	3
EDUCACIÓN SEXUAL 1	NO	BÁSICO	4
LOGICA MATEMATICA 1	NO	BÁSICO	7

ASIGNATURA

Nombre: LOGICA MATEMATICA **2**

Nivel: PREBÁSICA **3**

Carga horaria semanal: 4 **4**

Optativa: SI NO **5**

1 Nuevo **6** Guardar **7** Editar **8** Eliminar **9** Cancelar

Figura 43: Formulario de registro de asignaturas

1. Al aparecer la pantalla de asignatura, el primer paso que debemos hacer es dar clic en nuevo, para ingresar una asignatura.
2. Luego ubicamos el nombre de la asignatura que deseemos ingresar
3. Seleccionamos el nivel donde se impartirá esta asignatura.
4. Asignamos la carga horaria semanal
5. Si la materia es optativa, ubicamos si en el botón optativa. (optativa significa, materias impartidas por un mismo docente, en todos los niveles, como por ejemplo: cultura física.
6. Una vez ingresados los datos correctamente, damos clic en guardar, para almacenar la información.
7. Si deseamos editar una materia, seleccionamos una dando clic sobre ella y luego presionamos el botón editar y guardamos.
8. Si deseamos eliminar una asignatura, seleccionamos una dando clic sobre ella y luego presionamos el botón eliminar.
9. Para cancelar una acción, damos clic en el botón cancelar.

MENSAJES DE ERRORES

Cuando las asignaturas se ingresan dos veces en el mismo año de educación básica aparece el siguiente cuadro de error:

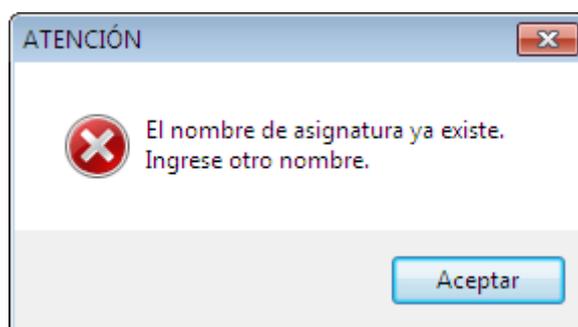


Figura 44: Mensaje de error: asignatura ya existe

Si se pretende guardar el registro de una asignatura y falta llenar algún campo, aparecerá el siguiente mensaje de error:

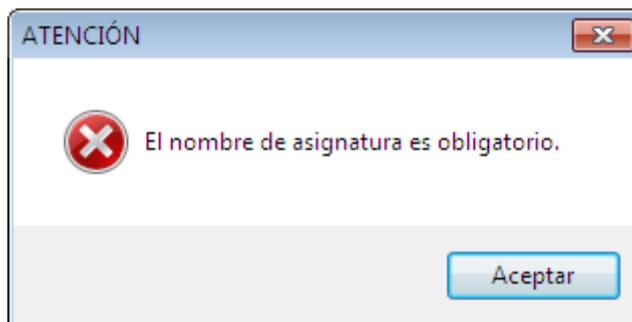


Figura 45: Mensaje de error: campos obligatorios

16. FORMULARIO AÑO DE EDUCACIÓN BÁSICA

Permite ingresar los años de educación básica con sus respectivos paralelos.

Lo primero que debemos hacer, es abrir el formulario de año de educación básica. (Ver figura10) o (Ver figura 13)

AEB	PARALELO	NIVEL
NIVEL	A	PREBÁSICA
NIVEL	B	PREBÁSICA
PRIMERO	A	BÁSICO
PRIMERO	B	BÁSICO
SEGUNDO	A	BÁSICO
SEGUNDO	B	BÁSICO
TERCERO	A	BÁSICO
CUARTO	A	BÁSICO
QUINTO	A	BÁSICO
QUINTO	B	BÁSICO
SEXTO	A	BÁSICO
SEPTIMO	A	BÁSICO

AÑO DE EDUCACIÓN BÁSICA

Nombre: NIVEL **2**

Paralelo: A **3**

Nivel: PREBÁSICA **4**

Número: 1 **5**

1 Nuevo Guardar Editar Eliminar Cancelar

Estado **6 7 8 9**

Figura 46: Formulario de año de educación básica

1. Al aparecer el formulario de año de educación básica, el primer paso que debemos hacer es dar clic en nuevo, para ingresar uno.

2. Luego ubicamos el nombre del año de educación básica o curso.
3. Ingresamos el paralelo al que pertenece
4. Seleccionamos el nivel al que pertenecerá este año de educación básica.
5. El número que le asignemos, sirve para ordenar los años de educación básica en forma ascendente, ejemplo a primero y a pre básica le asignamos 1, a segundo el 2 y así sucesivamente para que se ordenen en el datagrid
6. Si Una vez ingresados los datos correctamente, damos clic en guardar, para almacenar la información.
7. Si deseamos editar, seleccionamos un año de educación básica dando clic sobre el registro y luego presionamos el botón editar y guardamos.
8. Si deseamos eliminar un año de educación básica, seleccionamos uno dando clic sobre el y luego presionamos el botón eliminar.
9. Para cancelar una acción, damos clic en el botón cancelar

MENSAJE DE ERROR:

Si se ingresa un año de educación básica con el mismo paralelo, aparece el siguiente mensaje

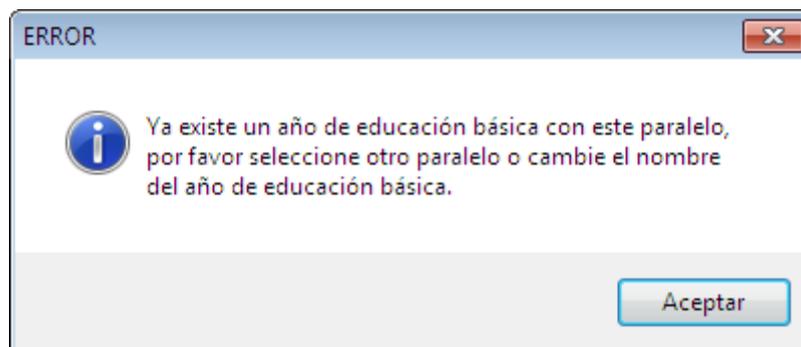
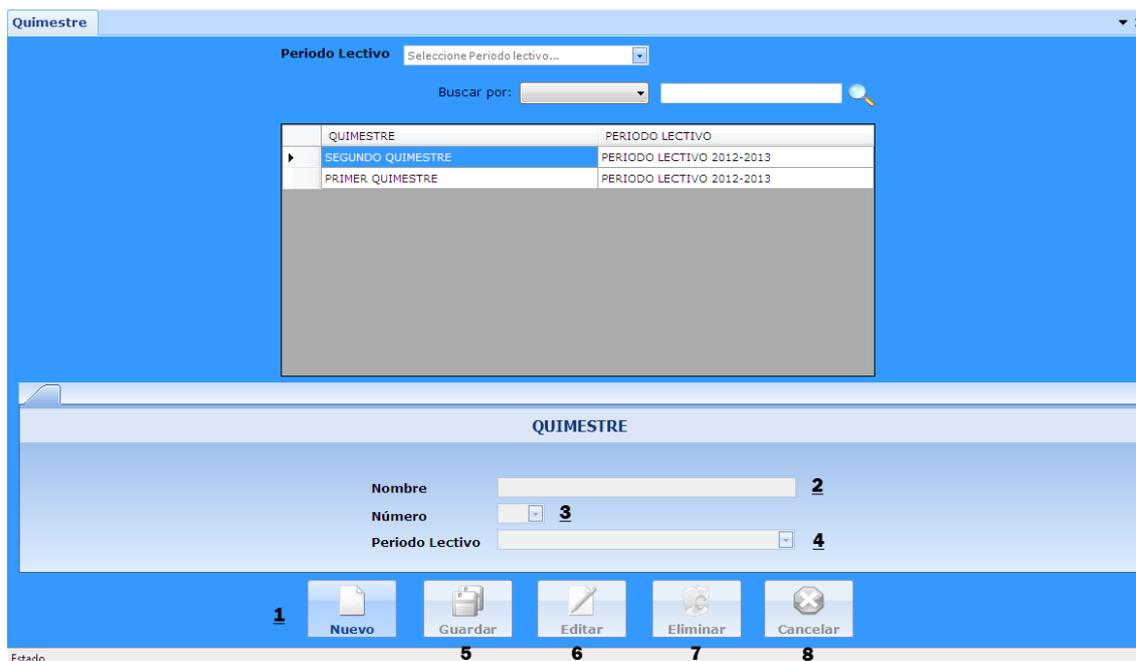


Figura 47: Mensaje de error: Año de educación básica ya existe

17. FORMULARIO QUIMESTRE

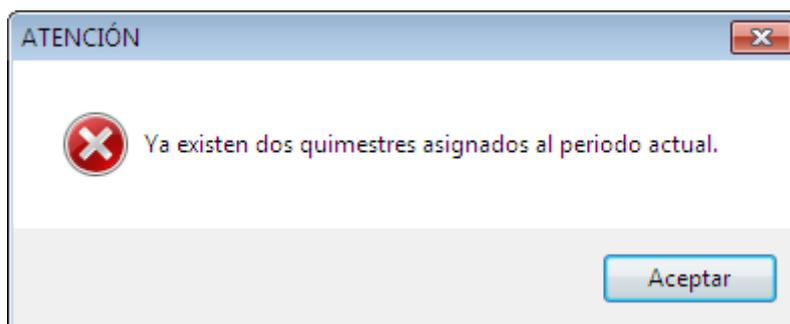
Permite ingresar y almacenar los respectivos quimestres que se utilizan en la institución: Primer y segundo quimestre. Lo primero que debemos hacer es abrir el formulario (Ver figura 6).



1. Al aparecer el formulario de quimestre, el primer paso que debemos hacer es dar clic en nuevo, para ingresar uno.
2. Luego ubicamos el nombre del quimestre.
3. Seleccionamos el número uno, que corresponde al primer quimestre
4. Le asignamos el periodo lectivo correspondiente.
5. Una vez ingresados los datos correctamente, damos clic en guardar, para almacenar la información.
6. Si deseamos editar, seleccionamos un quimestre dando clic sobre el registro y luego presionamos el botón editar y guardamos.
7. Si deseamos eliminar un quimestre, seleccionamos uno dando clic sobre el y luego presionamos el botón eliminar.
8. Para cancelar una acción, damos clic en el botón cancelar

MENSAJE DE ERROR:

No se puede ingresar otro quimestre, debido a que la institución solo se manejan dos, si se pretende ingresar otro quimestre aparecerá el siguiente mensaje:



18. FORMULARIO BIMESTRE

Permite ingresar y almacenar los respectivos bimestres que se utilizan en la institución: Primer, segundo, tercero y cuarto bimestre. (Dos para cada quimestre). Lo primero que debemos hacer es abrir el formulario (Ver figura 6)

BIMESTRE	QUIMESTRE	PERIODO LECTIVO
PRIMER BIMESTRE	PRIMER QUIMESTRE	PERIODO LECTIVO 2012-2013
SEGUNDO BIMESTRE	PRIMER QUIMESTRE	PERIODO LECTIVO 2012-2013
TERCERO BIMESTRE	SEGUNDO QUIMESTRE	PERIODO LECTIVO 2012-2013
CUARTO BIMESTRE	SEGUNDO QUIMESTRE	PERIODO LECTIVO 2012-2013

BIMESTRE

Nombre: PRIMER BIMESTRE **2**

Número: 1 **3**

Período Lectivo: PERIODO LECTIVO 2012-2013 **4**

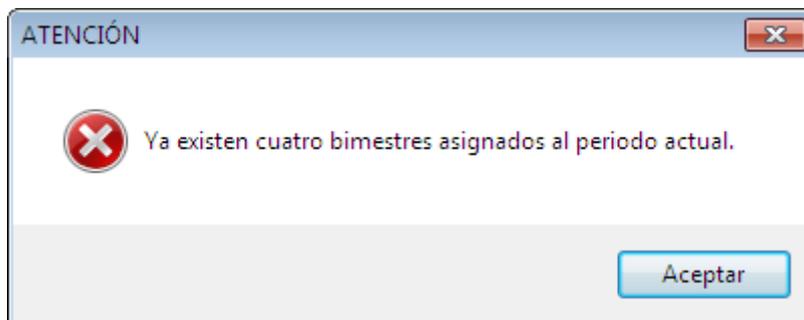
Quimestre: PRIMER QUIMESTRE **5**

1 Nuevo **6** Guardar **7** Editar **8** Eliminar **9** Cancelar

1. Al aparecer el formulario de bimestre, el primer paso que debemos hacer es dar clic en nuevo, para ingresar uno.
2. Luego ubicamos el nombre del bimestre.
3. Seleccionamos el número uno, que corresponde al primer bimestre, el dos para el segundo bimestre, el tres para el tercero y el cuatro para el cuarto.
4. Le asignamos el periodo lectivo correspondiente.
5. Seleccionamos al quimestre que pertenece
6. Una vez ingresados los datos correctamente, damos clic en guardar, para almacenar la información.
7. Si deseamos editar, seleccionamos un bimestre dando clic sobre el registro y luego presionamos el botón editar y guardamos.
8. Si deseamos eliminar un bimestre, seleccionamos uno dando clic sobre el y luego presionamos el botón eliminar.
9. Para cancelar una acción, damos clic en el botón cancelar

MENSAJE DE ERROR:

No se puede ingresar más de cuatro bimestres, debido a que la institución solo se manejan cuatro, dos por cada quimestre. Si se pretende ingresar otro bimestre aparecerá el siguiente mensaje:



19. OPCIÓN DE BÚSQUEDAS.

Los formularios tienen incluido una opción de búsquedas, que mediante filtros agilizan la búsqueda de la información. Para realizar la búsqueda solo seleccionamos la opción por la que deseamos buscar, puede ser por medio de nombres, cedula, periodo lectivo, etc, esto depende del formulario en que nos encontremos.

Alumnos

Periodo Lectivo: PERIODO LECTIVO 2012-2013

Buscar por: Nombres

	CEDULA	NOMBRES	GENERO	NACIONALIDAD	DIRECCIÓN	FECHA NACIMIENTO
▶	1312536483	CARRERRA SOLIS DOMENICA CECIBEL	FEMENINO	ECUATORIANA	LOS CEIBOS	12/02/1888
	1310621964	GARCIA MENENDEZ VIVIANA CLARIBEL	FEMENINO	ECUATORIANA	LAS VILLAS	04/05/1992
	1312850496	MENDOZA MARIANO	MASCULINO	ECUATORIANA	CANUTO	22/08/1985

Matricula

Nivel: BÁSICO

AEB: PRIMERO A

Buscar por: Cedula

	CEDULA	ALUMNO	FECHA MATRÍCULA	AEB	PERIODO LECTIVO
▶	1313136432	MARTHA CRUZ	22/08/2012	PRIMERO A	PERIODO LECTIVO 2012-2013
	1312536483	DOMENICA CECIBEL CARRERRA SOLIS	23/08/2012	PRIMERO A	PERIODO LECTIVO 2012-2013

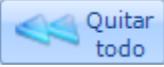
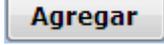
Año educación basica

Buscar por: Nivel

	AEB	PARALELO	NIVEL
▶	OCTAVO	A	SECUNDARIO
	NOVENO	A	SECUNDARIO
	DECIMO	A	SECUNDARIO

20. CUADRO EXPLICATIVO DE LOS BOTONES

 Nuevo	Permite crear un nuevo registro.
 Guardar	Permite guardar un registro
 Editar	Permite editar un registro
 Eliminar	Permite eliminar un registro
 Cancelar	Permite cancelar una acción

	Permite agregar una asignatura
	Permite agregar todas las asignaturas
	Permite quitar una asignatura
	Permite quitar todas las asignaturas
	Permite agregar datos.
	Indica que alguna entidad se encuentra en estado activo.
	Indica que alguna entidad se encuentra en estado inactivo.

21. REPORTES

- **REPORTES DE MATRICULACIÓN**

Si deseamos tener el reporte de los estudiantes matriculados durante todo el periodo lectivo hacemos lo siguiente:

- 1) Seleccionamos el periodo lectivo
- 2) Clic en generar.

Reporte Matriculados

Periodo lectivo: PERIODO LECTIVO 2012-2013 Nivel: Seleccione nivel. AEB: Seleccione año de educación básica.

1

GENERAR **2**

Informe principal

ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA
"ARISTOS - CALCETA"
Creada mediante acuerdo ministerial #006 DP - DECM
REPORTE DE ALUMNOS MATRICULADOS
PERIODO LECTIVO 2012-2013 27/08/2012

NIVEL: BÁSICO

AÑO EDUCACIÓN BÁSICA: PRIMERO

CEDULA	APELLIDOS	NOMBRES	PARALELO
1312536483	CARRERRA SOLIS	DOMENICA CECIBEL	A
1313136432	CRUZ	MARTHA	A

AÑO EDUCACIÓN BÁSICA: QUINTO

CEDULA	APELLIDOS	NOMBRES	PARALELO
4346544664	GUADALUPE MENEZES	GUADALUPE CLAUDIA	A

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Estado

Figura 48: Reporte de alumnos matriculados por periodo lectivo

Si deseamos tener el reporte de los estudiantes matriculados durante todo el periodo lectivo por nivel hacemos lo siguiente:

- 1) Seleccionamos el periodo lectivo
- 2) Escogemos el nivel
- 3) Clic en generar

Reporte Matriculados

Periodo lectivo: PERIODO LECTIVO 2012-2013 Nivel: PREBÁSICA AEB: Seleccione año de educación básica.

1 2 3 GENERAR

Informe principal

ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA
"ARISTOS - CALCETA"
Creada mediante acuerdo ministerial #006 DP - DECM
PERIODO LECTIVO 2012-2013 27/08/2012

NIVEL: PREBÁSICA

AÑO EDUCACIÓN BÁSICA: NIVEL

CEDULA	APELLIDOS	NOMBRES	PARALELO
1313136473	LOPEZ	PEDRO	B
1313136424	MUÑOZ	JUAN	A

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Estado

Figura 49: Reporte de alumnos matriculados por nivel

Si deseamos tener el reporte de los estudiantes matriculados por año de educación básica, hacemos lo siguiente:

- 1) Seleccionamos el periodo lectivo
- 2) Escogemos el nivel
- 3) Elegimos el AEB (año de educación básica)
- 4) Clic en generar

Reporte Matriculados

Periodo lectivo: PERIODO LECTIVO 2012-2013 Nivel: BÁSICO AEB: PRIMERO A

1 2 3

GENERAR 4

Informe principal

ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA
"ARISTOS - CALCETA"
Creada mediante acuerdo ministerial #006 DP - DECM
PERIODO LECTIVO 2012-2013 27/08/2012

NIVEL: BÁSICO

AÑO EDUCACIÓN BÁSICA: PRIMERO

CEDULA	APELLIDOS	NOMBRES	PARALELO
1312536483	CARRERRA SOLIS	DOMENICA CECIBEL	A
1313136432	CRUZ	MARTHA	A

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Estado

Figura 50: Reporte de alumnos matriculados por año de educación básica

- **REPORTE DE CALIFICACIONES**

Si deseamos tener el reporte de las calificaciones por periodo lectivo, hacemos lo siguiente:

- 1) Seleccionamos el periodo lectivo
- 2) clic en generar

Reporte Calificaciones

Periodo lectivo: PERIODO LECTIVO 2012-2013 Nivel: Seleccione nivel. AEB: Seleccione año de educación básica.

1 Materia Alumno

GENERAR 2

SAP CRYSTAL REPORTS®

Informe principal

ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA
"ARISTOS - CALCETA"
Creada mediante acuerdo ministerial #006 DP - DECM

REPORTE DE CALIFICACIONES
PERIODO LECTIVO 2012-2013

27/08/2012

ALUMNO	AÑO EDUCACIÓN BÁSICA	NIVEL	TOTAL
CARRERRA SOLIS DOMENICA	PRIMERO A	BASICO	8,23
CRUZ MARTHA	PRIMERO A	BASICO	7,50
LOPEZ PEDRO	NIVEL B	PREBÁSICA	2,00
MUNOZ JUAN	NIVEL A	PREBASICA	1,31

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Estado

Figura 51: Reporte de calificaciones por periodo lectivo

Si deseamos obtener las calificaciones por nivel, hacemos lo siguiente:

- 1) Seleccionamos periodo lectivo
- 2) Escogemos el nivel
- 3) Clic en generar

Reporte Calificaciones

Periodo lectivo: PERIODO LECTIVO 2012-2013 Nivel: **BÁSICO** AEB: Seleccione año de educación básica.

1 Materia Alumno

GENERAR 3

SAP CRYSTAL REPORTS®

Informe principal

ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA
"ARISTOS - CALCETA"
Creada mediante acuerdo ministerial #006 DP - DECM

REPORTE DE CALIFICACIONES
PERIODO LECTIVO 2012-2013

27/08/2012

ALUMNO	AÑO EDUCACIÓN BÁSICA	NIVEL	TOTAL
CARRERRA SOLIS DOMENICA	PRIMERO A	BASICO	8,23
CRUZ MARTHA	PRIMERO A	BASICO	7,50

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Estado

Figura 52: Reporte de calificaciones por nivel

Si deseamos obtener las calificaciones por el año de educación básica, hacemos lo siguiente:

- 1) Seleccionamos el periodo lectivo
- 2) Escogemos el nivel
- 3) Elegimos el AEB
- 4) Clic en generar

Reporte Calificaciones

Periodo lectivo: PERIODO LECTIVO 2012-2013 Nivel: BÁSICO AEB: PRIMERO A

1 2 3

Materia Alumno

GENERAR 4

Informe principal

ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA
"ARISTOS - CALCETA"
Creada mediante acuerdo ministerial #006 DP - DECM

REPORTE DE CALIFICACIONES
PERIODO LECTIVO 2012-2013

27/08/2012

ALUMNO	AÑO EDUCACIÓN BÁSICA	NIVEL	TOTAL
CARRERRA SOLIS DOMENICA	PRIMERO A	BÁSICO	8,23
CRUZ MARTHA	PRIMERO A	BÁSICO	7,50

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Estado

Figura 53: Reporte de calificaciones por materia por año de educación básica

Si deseamos obtener las calificaciones ya sea por estudiante o por materia, hacemos lo siguiente:

- 1) Seleccionamos el periodo lectivo
- 2) Escogemos el nivel
- 3) Elegimos el AEB

4) Seleccionamos si deseamos obtener las calificaciones por alumno o por asignatura

5) Clic en generar

Reporte Calificaciones

Periodo lectivo: PERIODO LECTIVO 2012-2013 Nivel: BÁSICO AEB: PRIMERO A

Materia Alumno

GENERAR

ESCUELA PARTICULAR DE EDUCACIÓN GENERAL BÁSICA
"ARISTOS - CALCETA"
Creada mediante acuerdo ministerial #006 DP - DECM

REPORTE DE CALIFICACIONES
PERIODO LECTIVO 2012-2013
NIVEL
BÁSICO

ALUMNO: CARRERRA SOLIS DOMENI CURSO: PRIMERO A

ASIGNATURA	PROMEDIO	SUPLETORIO	TOTAL	ESTADO
DEPORTES 1	6,93	1,30	8,23	APROBADO
PROMEDIO TOTAL:				8,23

ALUMNO: CRUZ MARTHA CURSO: PRIMERO A

ASIGNATURA	PROMEDIO	SUPLETORIO	TOTAL	ESTADO

Nº de página actual: 1 Nº total de páginas: 1 Factor de zoom: 100%

Estado

Figura 54: Reporte de calificaciones por materia o por alumno.