

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA, MANAGUA
Facultad Regional Multidisciplinaria de Chontales
“Cornelio Silva Arguello”
UNAN – Managua / FAREM – Chontales



**Seminario de Graduación para optar al Título de
Ingeniero en Sistemas de Información**

Tema General

Desarrollo de sistemas web y/o aplicaciones móviles, utilizando arquitectura distribuida

Subtema

Desarrollo de un sistema web para la gestión Académica, utilizando arquitectura distribuida para Academia Bilingüe Amerrisque, de la ciudad de Juigalpa, durante el segundo semestre del año 2016.

Presentado por:

- Br. Joaquín Melquisedec Espinoza Miranda
- Br. Robinson Akiles Obando Aragón

Tutores:

- M.Sc. Miriam Patricia Téllez Marín
- M.Sc. Saira María Urbina Cienfuegos

Juigalpa, 2017



Universidad Nacional Autónoma De Nicaragua, Managua
Facultad Regional Multidisciplinaria de Chontales
“Cornelio Silva Arguello”

2017 “Año de la Universidad Emprendedora”

Departamento de Ciencia, Tecnología y Salud

Valoración Docente

Los sistemas web son vitales en los centros educativos privados y públicos del país, las autoridades correspondientes son responsables en que se implementen dando paso al avance tecnológico que facilita la comunicación a través internet y conexión a dispositivos móviles de uso común en la comunidad.

Los negocios de todas las áreas están siendo promovidos por Internet a través de sistemas web, acercándose con facilidad a usuarios finales, el sector educativo requiere incorporar éste tipo de tecnología que les permita mejorar la calidad de la gestión académica. El tema “Desarrollo de un sistema web para la **gestión académica**, utilizando arquitectura distribuida en Academia Bilingüe Amerrisque, de la ciudad de Juigalpa, durante el segundo semestre del año 2016”, es pertinente en el centro de educación primaria, que siendo aún joven en la ciudad emerge como pionero en la tecnología web, para facilitar la comunicación con los padres de familia evitando realizar trámites tediosos de manera personal y agilizar los procesos de registro del colegio.

Se considera que los estudiantes **Joaquín Melquisedec Espinoza Miranda y Robinson Akiles Obando Aragón** reúnen los requisitos y méritos para la evaluación del jurado examinador en la defensa de trabajo de grado para optar al título de **Ingeniero en Sistemas de Información**.

M.Sc. Miriam Patricia Téllez Marín

M.Sc. Saira María Urbina Cienfuegos

Dedicatoria

Dedicamos este documento de tesis:

A Dios.

Por habernos permitido llegar hasta este punto y habernos dado salud y sabiduría para lograr nuestros objetivos, además de su infinita bondad y amor.

A nuestros padres

A nuestros padres por ser el pilar fundamental en todo lo que somos, en toda nuestra educación, tanto académica, como de la vida, por su incondicional apoyo perfectamente mantenido a través del tiempo.

A nuestros amigos.

Que nos apoyamos mutuamente en nuestra formación profesional y que hasta ahora, seguimos siendo amigos: Jorge Cruz, Weslin Vargas, Miguel Díaz, Karen Álvarez, Yari Fonseca, Freddy Sirias, Evert Ortíz, Jorlenis Ríos.

A nuestros maestros

Aquellos que marcaron cada etapa de nuestro camino universitario, y que nos ayudaron en asesorías y dudas presentadas en la elaboración de la tesis: Lic. Alfredo Rodríguez, MSC. Miriam Téllez, MSC. Saira Urbina, Ing. Dionicio Manzanares, MSC Jonatan Moreno, MSC. Jazcar Bravo.

Agradecimientos

El presente trabajo va dirigido con una expresión de gratitud para nuestros distinguidos maestros, que con nobleza y entusiasmo, nos transmitieron todo su conocimiento para formarnos como profesionales, dando lo mejor y soportando con paciencia nuestras actitudes negativas en algunos momentos.

Agradecemos a Dios por prestarnos la salud y la sabiduría para haber culminado la etapa de esta carrera universitaria.

Agradecimiento profundo a nuestros padres quienes han luchado con mucho amor de nuestro lado, brindándonos todo lo necesario en aspectos económicos y espirituales.

Finalmente, agradecemos de forma especial a los encargados de la organización de la universidad por motivarnos y ayudarnos a salir adelante brindándonos las oportunidades y facilidades de estudiar mediante los programas internos de ayuda a los estudiantes.

Resumen Ejecutivo

El proyecto desarrollado tiene la finalidad de ayudar en la automatización de algunos de los procesos de matrícula, notas y caja que se desarrollan de forma manual en Academia Bilingüe Amerrisque.

Algunos de los problemas que se dan en estos procesos son:

- Filas para realizar pagos en caja
- Dificultad en tiempo y distancia para llegar al centro educativo.
- Almacenamiento de la información.
- Llenado de formularios de datos.
- Seguridad de la información.

Como modelo para el desarrollo de un Software, este proyecto está siguiendo las técnicas que se ha encontrado en el modelo de desarrollo del sistema que sugiere el Señor Ian Sommerville, encaminando este proyecto cumpliendo con los cuatro pasos que propone este paradigma: Análisis, diseño, desarrollo y validación. [1].

Para obtener e identificar el tipo de información que se necesitó para realizar este proyecto de software se han utilizado fuentes y técnicas que se emplearon para la recolección de la información. Por ejemplo:

- La observación.
- Entrevistas.
- Estándares de Calidad.

Para el ciclo de desarrollo de este software se han seguido estándares y normas de calidad para obtener los mejores resultados y beneficios que puedan ser vistos por los clientes o usuarios del sistema.

Para el análisis de los requerimientos funcionales y no funcionales, que son todas las necesidades, roles y característicos que requiere el cliente y los usuarios para el correcto funcionamiento del sistema, se utilizó un documento de especificación de requerimientos (SRS) el cual está basado en el Estándar IEEE 830 – 1998.

A través del estándar IEEE830, la recolección de todos los requerimientos funcionales y no funcionales que son las necesidades del cliente y lo que el cliente espera que el software realice.

Mediante las herramientas de trabajo como Visual Studio.Net, SQL Server, entre otras, se logró el desarrollo del software y sus funciones en base a los requisitos solicitados por el cliente.

Se creó una base de datos con 17 tablas, cada una con sus respectivos campos y sus relaciones de la cual depende una de las capas de este proyecto, también se han generado bocetos a seguir para la codificación del diseño y se ha desarrollado la codificación de los servicios, clases, métodos y técnicas apropiadas para la programación del sistema.

Obteniendo como resultados la creación funcional de las tres modularidades con las que cuenta este sistema, las cuales son: Matricula en línea, notas en línea y caja.

Contenido

| | | |
|-------|---|----|
| I. | Introducción | 1 |
| II. | Antecedentes | 2 |
| III. | Planteamiento del problema..... | 4 |
| IV. | Objetivos | 6 |
| 1.1 | Objetivo General..... | 6 |
| 1.2 | Objetivos Específicos | 6 |
| V. | Marco teórico | 7 |
| 5.1 | Tipos de desarrollo de aplicaciones web | 7 |
| 5.1.1 | Aplicación web estática | 7 |
| 5.1.2 | Aplicación web dinámica | 7 |
| 5.1.3 | Tienda virtual o comercio electrónico | 8 |
| 5.1.4 | Portal web App | 8 |
| 5.1.5 | Aplicación web animada..... | 8 |
| 5.1.6 | Aplicación web con “Gestor de Contenidos” | 9 |
| 5.2 | Arquitectura de sistemas distribuidos..... | 9 |
| 5.2.1 | Definición: | 9 |
| 5.2.2 | Propiedades de los sistemas distribuidos | 10 |
| 5.3 | Sistemas Distribuidos | 11 |
| 5.3.1 | Características clave de los sistemas distribuidos | 12 |
| 5.3.2 | Compartición de Recursos..... | 12 |
| 5.3.3 | Apertura (openness) | 13 |
| 5.3.4 | Básicamente los sistemas distribuidos cumplen una serie de características: | 13 |
| 5.4 | El Modelo Cliente Servidor | 17 |
| 5.4.1 | Middleware..... | 19 |
| 5.4.2 | Middleware para sistemas cliente-servidor: RPC..... | 19 |
| 5.4.3 | Servidores Pesados vs Clientes Pesados..... | 21 |
| 5.4.4 | Sistemas N-Tiered | 22 |

| | |
|--|----|
| 5.4.5 Ventajas de los sistemas cliente-servidor | 23 |
| 5.5 Sitios web..... | 23 |
| 5.5.1 Ventajas de un sitio Web | 24 |
| 5.5.2 Domain Name Server (DNS)..... | 24 |
| 5.5.3 Servicio de Hosting..... | 24 |
| 5.5.4 Dominio | 24 |
| 5.5.5 Redirección de Dominio | 24 |
| 5.5.6 Diferencia un Hosting de un Dominio..... | 25 |
| 5.5.7 File Transfer Protocol (FTP) | 25 |
| 5.5.8 Lista de correo y sus ventajas | 25 |
| 5.5.9 Banner..... | 26 |
| 5.6 Etapa de análisis | 26 |
| 5.6.1 Requerimientos del software | 26 |
| 5.6.2 Características de los requerimientos | 27 |
| 5.6.3 Los requerimientos deben ser | 27 |
| 5.6.4 Fundamentos del Análisis de Requerimientos..... | 27 |
| 5.6.5 Tareas del Análisis..... | 29 |
| 5.6.6 Especificación de Requisitos de Software (SRS) | 29 |
| 5.7 Técnicas para hallar datos..... | 30 |
| 5.7.1 Entrevista | 30 |
| 5.7.2 Observación | 31 |
| 5.7.3 Experiencia | 31 |
| 5.8 Diseño del sistema recomendado..... | 32 |
| 5.8.1 Balsamiq Mockups | 32 |
| 5.8.2 StarUML..... | 33 |
| 5.8.3 HTML..... | 34 |
| 5.8.4 CSS | 35 |
| 5.8.5 Bootstrap..... | 37 |
| 5.8.6 Materialize CSS | 37 |
| 5.9 Desarrollo y documentación del sistema | 37 |
| 5.9.1 La Documentación..... | 38 |
| 5.9.2 Visual Studio | 39 |

| | | |
|-------|---|----|
| 5.9.3 | Arquitectura Cliente servidor | 39 |
| 5.9.4 | Programación en capas | 39 |
| 5.9.5 | Prueba y Mantenimiento del Sistema | 40 |
| VI. | Marco Metodológico..... | 43 |
| 6.1 | Sistema de Variables..... | 43 |
| 6.2 | Hipótesis | 43 |
| 6.3 | Operacionalización de las variables..... | 44 |
| 6.4 | Tipo de investigación..... | 47 |
| 6.5 | Métodos y técnicas de recolección. | 47 |
| 6.6 | Técnicas de procesamiento y análisis de datos..... | 48 |
| 6.7 | Modelo de desarrollo del Software..... | 48 |
| VII. | Marco administrativo | 50 |
| 7.1 | Cronograma | 51 |
| 7.2 | Presupuesto del trabajo..... | 54 |
| VIII. | Marco de resultado | 61 |
| 8.1 | Cronograma del desarrollo del software..... | 61 |
| 8.2 | Descripción de los resultados por objetivos | 63 |
| 8.2.1 | Análisis de requerimientos del software..... | 63 |
| 8.2.2 | Entrevista realizada a dirección del colegio A.B.A. | 63 |
| 8.2.3 | Documento de especificación de requerimientos del software..... | 63 |
| 8.3 | Diseño y modelado del software..... | 64 |
| 8.3.1 | Diagramas de Casos de uso | 64 |
| 8.3.2 | Bocetos del Sistema..... | 64 |
| 8.3.3 | Diseño de la base de datos | 65 |
| 8.3.4 | Diccionario de datos | 65 |
| 8.4 | Desarrollo del software..... | 65 |
| IX. | Conclusiones. | 67 |
| X. | Recomendaciones..... | 68 |

| | | |
|-------|---------------------------|----|
| XI. | Glosario de términos..... | 69 |
| XII. | Referencias..... | 75 |
| XIII. | ANEXOS..... | 78 |

Índice de Tablas

| | |
|--|----|
| Tabla 1 - Operacionalización de las variables..... | 45 |
| Tabla 2 – Identificación de requerimiento..... | 46 |
| Tabla 3 - Diseño de las interfaces..... | 47 |
| Tabla 4 - Total de líneas de códigos..... | 54 |
| Tabla 5 - Costo de desarrollo..... | 59 |
| Tabla 6 - Costo de licencia..... | 59 |
| Tabla 7 - Costo del documento..... | 60 |
| Tabla 8 - Costo total del proyecto | 60 |

Índice de Ecuaciones

| | |
|-------------------------------------|----|
| Ecuación 1 - Esfuerzo..... | 54 |
| Ecuación 2 - Cronograma..... | 54 |
| Ecuación 3 - Tiempo cronograma..... | 58 |

Índice de Ilustraciones

| | |
|--|----|
| Ilustración 1 - Cronograma de actividades..... | 52 |
| Ilustración 2 - Muestras de variaciones..... | 54 |

I. Introducción

Este documento se refiere al desarrollo de una aplicación web para la gestión académica de Academia Bilingüe Amerrisque de Chontales, Nicaragua, la cual se encarga de llevar el registro de los estudiantes que han sido matriculados en el centro educativo, junto con sus tutores. Para ello esta aplicación permitirá la matrícula online para que los estudiantes ya registrados en el sistema no se queden sin cupos de matrículas para estudiar en el mismo colegio. Además el sistema mostrará en línea las notas de los estudiantes por cada corte evaluativo. También este sistema tendrá un módulo que permitirá llevar la contabilidad de la caja.

Este proyecto de software se desarrolla con la finalidad de innovar y a la vez facilitar ciertas tareas automatizando algunos de los procesos que llevan mucho tiempo hacerlas de forma manual en las instituciones educativas.

En años anteriores, se han desarrollado aplicaciones del mismo tipo, con la misma finalidad en Juigalpa Chontales, pero en esta ciudad no ha sido implementada en ningún colegio o institución educativa, debido a que se tiene un temor al cambio, que genera la incertidumbre de: si una aplicación de este tipo daría resultado en una ciudad como esta; se ha decidido tomar esta opción por ser muy importante y necesaria para la sociedad, beneficiando directamente a los padres de familia que ahorrarían mucho tiempo en realizar una matrícula de sus hijos en línea, desde la comodidad de su hogar. De igual forma este sistema beneficia directamente al centro educativo que lo implemente ya que el control interno y manejo de la información se agilizaría, facilitaría y sería más seguro.

Los estudiantes y docentes de la institución educativa donde se implemente este sistema también serán beneficiados de forma indirecta debido a la organización y las alternativas que brinda el software para la toma de decisiones por parte de la dirección y administración del centro.

Para el ciclo de desarrollo de este software se han seguido estándares y normas de calidad para obtener los mejores resultados y beneficios que puedan ser vistos por los clientes o usuarios del sistema y para que proyectos de este tipo sean mucho más confiables y las instituciones educativas permitan e inviertan en la implementación de estas aplicaciones.

II. Antecedentes

Existen muchos sistemas web de gestión académica en todo el mundo en cada colegio o universidad de cada país, algunos ejemplos de ellos son los siguientes:

- ✓ SIMAT: Ministerio de Educación Nacional de Colombia. Sistema integrado de matrícula SIMAT es una herramienta que permite organizar y controlar los procesos de matrícula en todas sus etapas. [2]
- ✓ Sistema de Consultas y Matricula Web – Universidad tecnológica de Panamá. [3]
- ✓ Sistema Web para el Registro y Control Estudiantil de la Misión Sucre, Aldea Regional de Apoyo al Maestro (CRAM), Municipio Sucre del Estado Mérida. Marzo 2014 [4]

En Nicaragua existen muchos sistemas de matrículas, algunos de ellos son:

- ✓ Sistema de matrícula UNAN Managua. Este Sistema permite realizar matriculas online. [5]
- ✓ Universidad Centroamericana UCA de Nicaragua. Este sistema permite a sus estudiantes registrar su matrícula y llevar el control de sus calificaciones, etc. [6]
- ✓ Renovado y moderno sistema de Gestión Académica, UNI: *Consultar calificaciones, revisar horarios, actualizar información, solicitar becas, inscribir asignaturas, conocer resultados evaluativos parciales y totales, además de múltiples gestiones son las ventajas del Sistema de Registro Académico en línea de la UNI que en este inicio del 2016 tiene una nueva versión que permitirá a los estudiantes tener acceso desde cualquier dispositivo y desde todo el país.* [7]

En Juigalpa existen algunas aplicaciones de este tipo pero no han sido implementadas, pero en algunas universidades existe un módulo de un sistema integrado destinado para la región que lo requiera, tal es el caso del sistema de gestión de la UNAN Managua.

- ✓ Sistema Integrado de UNAN Managua, FAREM – Chontales: Permite Administración, Secretaría Académica, Oferta Académica, Posgrado, Proyectos, Boletín, Cultura, Deporte, UNEN, Noticias, Matricula en Línea, etc. [8]

También existen aplicaciones que realizan los mismos procesos y que se pueden comprar en internet sin estar adecuadas a una empresa específica.

- ✓ DocCF Software de Gestión Escolar. – Grupo CF Developer ✓ MDG Matrículas, año 2008.
- ✓ aGora ERP.
- ✓ Argos Academia. – Argos Galaica.
- ✓ Gestión Escolar – River Soft.
- ✓ MegaConta – InfoAgil.
- ✓ FactoryGest – Microsoft Sistemas, S.L. ✓ Entre otros. [9]

En Academia Bilingüe Amerrisque (ABA), aún no se han implementado sistemas de información de ningún tipo.

III. Planteamiento del problema

La implementación de este sistema pretende ayudar a la agilización de los procesos para consultar y resguardar la información, tales procesos se realizaban de forma manual por lo tanto las consecuencias eran la inconformidad de los clientes y lentitud en las consultas de la información.

Ejemplos de algunos de estos procesos son los siguientes:

- ❖ Llenado de hoja de matricula
- ❖ Pagos en caja
- ❖ Consulta de calificaciones
- ❖ Consulta de datos de estudiantes y tutores
- ❖ Pasar calificaciones
- ❖ Listado de estudiantes por grado
- ❖ Listado de docentes
- ❖ Generar informes de estado de clientes
- ❖ Generar informes de ingresos y egresos.

Academia Bilingüe Amerrisque (ABA) de Juigalpa Chontales, fue fundada el 08 de junio del año 2012, comenzando a desarrollar las modalidades de clases: Materno y preescolar luego se incorporó la modalidad de Primaria.

El número de estudiantes inscritos desde el 2012 al 2016 aumentó con un porcentaje considerablemente favorable, aproximadamente de 39 estudiantes a 115 actualmente.

En Academia Bilingüe Amerrisque (ABA), los procesos de matrícula, entregas de notas y pagos de mensualidades que involucran al personal administrativo de la Academia y a los padres de familia junto con sus hijos, se efectúan de forma manual y obligatoriamente presencial, es decir, el tutor o responsable del estudiante, tiene la obligación de asistir al centro de estudio para obtener la garantía de no perder un cupo de su matrícula, hacer colas para pagar y luego para llenar la matrícula, teniendo como consecuencia un alto nivel de inconformidad por parte de los clientes y un manejo confuso de la información, puesto que existe redundancia en los datos y toda la información deberá ser almacenada en enormes cuadernos dificultando las consultas de información, haciendo más difícil el trabajo de modificar los datos de los registros y corriendo el riesgo de perder la información por falta de seguridad en el almacenamiento.

Otro de los aspectos que se debe tener en cuenta es, que este colegio es privado y está orientado para personas de clase media y clase alta que laboran, ya sea en un centro de trabajo o negocios personales, pero que tienen muchas obligaciones laborales teniendo así un horario que no les deja mucho tiempo libre. Además que el colegio está ubicado a la salida de la ciudad, por lo que toma mucho tiempo llegar desde cualquier parte del pueblo hasta la institución educativa, para entonces poder hacer el resto del proceso de pago de mensualidad, proceso de matrícula o consultas de notas.

Por este motivo se pensó en automatizar estos procesos elaborando un sistema de información que dé respuesta a esta problemática de forma que estos procesos se agilicen y a la vez la información sea más fácil de consultar y se encuentre almacenada con un mejor nivel de seguridad. Se usa la tecnología Web debido a que la matrícula será realizada en línea al igual que la consulta de calificaciones.

Control de pagos: Actualmente el colegio ha requerido agregar al sistema un nuevo módulo, y es el de “caja” ya que los procesos de pagos que se efectúan manualmente conlleva la misma dificultad de pérdida de recursos en tiempo, papelería, mobiliarios, equipos de oficina y personal. El sistema que se realizará gestiona: matrícula en línea, notas en línea y la contabilidad del colegio.

IV. Objetivos

1.1 Objetivo General

1. Desarrollar una aplicación web para la gestión académica, utilizando arquitectura distribuida en Academia Bilingüe Amerrisque de la ciudad de Juigalpa, durante el segundo semestre del año 2016.

1.2 Objetivos Específicos

1. Identificar los requerimientos funcionales y no funcionales que necesita la aplicación web para la gestión académica, utilizando estándar de especificación de requisitos de software IEEE 830.
2. Diseñar las interfaces y procesos teniendo en cuenta los requerimientos identificados, aplicando herramientas de diseño.
3. Implementar la codificación del diseño obtenido, a través del lenguaje de programación C# del modelo de desarrollo Web unificado ASP.NET, orientado a arquitectura distribuida.
4. Evaluar el desempeño del sistema web, de acuerdo a los requerimientos establecidos, utilizando el manual de pruebas de requerimientos funcionales del software.

V. Marco teórico

A continuación se presentan las definiciones del estudio realizado en la implementación de este proyecto, esta sección del documento incluye conceptos y definiciones de los diferentes tipos de tecnología y arquitectura utilizada, de igual forma se presentan los conceptos de la fase del Análisis y desarrollo del software con la que se estará trabajando.

5.1 Tipos de desarrollo de aplicaciones web

El desarrollo de aplicaciones web no es precisamente exclusivo para smartphones o tablets, sino que está pensado para ser ejecutado en cualquier navegador, ya sea en equipos de sobremesa o bien en portátiles o dispositivos móviles. Pero en esta ocasión queremos hacer una clasificación de los tipos de aplicaciones web.

Esta clasificación se hará dependiendo de la función de cómo se presentan las web apps con el contenido que deben mostrar. Distinguimos 6 tipos de aplicaciones web, y son las siguientes:

5.1.1 Aplicación web estática

Si optamos por crear una app web estática, lo primero que debemos saber es que este tipo de web app muestra poca información, y no suelen cambiar mucho.

Por regla general suelen estar desarrolladas en HTML y CSS. Aunque también pueden mostrar en alguna parte de la aplicación web objetos en movimiento como por ejemplo banners, GIF animados, vídeos, etc. También pueden desarrollarse aplicaciones web con jQuery y Ajax.

Además, modificar los contenidos de las apps estáticas no es sencillo. Para hacerlo, habría que descargar el HTML modificarlo y volver a subirlo al servidor. Y esos cambios solo puede hacerlos el webmaster o la empresa de desarrollo que programó y diseñó la web app en su momento.

Algunos ejemplos de desarrollo de aplicaciones web estáticas podrían ser por ejemplo portfolios de profesionales o bien un curriculum digital. Del mismo modo una página de presentación de empresa podría tener también este tipo de aplicación web para mostrar sus datos de contacto, etc. [10]

5.1.2 Aplicación web dinámica

Las aplicaciones web dinámicas son mucho más complejas a nivel técnico. Utilizan bases de datos para cargar a información, y estos contenidos se van actualizando cada vez que el usuario accede a la web

app. Generalmente cuentan con un panel de administración (llamado CMS) desde dónde los administrados pueden corregir o modificar los contenidos, ya sean textos o imágenes.

Existen muchos lenguajes de programación para desarrollar aplicaciones web dinámicas. Los lenguajes PHP y ASP son los más comunes porque permiten una buena estructuración del contenido.

El proceso de actualización es muy sencillo y ni siquiera necesita entrar en el servidor para modificarlo, además de que permite implementar muchas funcionalidades como foros o bases de datos. El diseño, y no solo el contenido pueden modificarse al gusto del administrador. [11]

5.1.3 Tienda virtual o comercio electrónico

Si por el contrario la aplicación web es una tienda o comercio digital, podemos decir que el desarrollo tenderá a parecerse al de un m-commerce o un e-commerce. El desarrollo es más complicado porque debe permitir pagos electrónicos a través de tarjeta de crédito, PayPal, u otro método de pago. En estos el desarrollador también deberá crear un panel de gestión para el administrador. A partir de él subirá los productos, actualizarlos o eliminarlos, así como gestionar pedidos y los pagos.

Algunos ejemplos de tiendas que tienen una aplicación web comercial podría ser la de cualquier gran empresa como El Corte Inglés en la que la web app se ajusta al dispositivo móvil como una aplicación móvil, permitiendo interactuar con ella como si fuera una app nativa.

5.1.4 Portal web App

Con portal nos referimos a un tipo de aplicación en el que la página principal permite el acceso a diversos apartados, categorías o secciones. Puede haber de todo: foros, chats, correo electrónico, un buscador, zona de acceso con registro, contenido más reciente, etc.

5.1.5 Aplicación web animada

Al hablar de animación, lo asociamos a la tecnología FLASH. Este tipo de programación permite presentar contenidos con efectos animados. Permite también diseños más creativos y modernos, es una de las tecnologías más utilizadas por diseñadores y creativos. El inconveniente de desarrollar aplicaciones web animadas es que para temas de posicionamiento web y optimización SEO, este tipo de tecnología no es la más adecuada ya que los buscadores no pueden leer correctamente las informaciones.

[11]

5.1.6 Aplicación web con “Gestor de Contenidos”

En el caso de aplicaciones web en las que el contenido se debe ir actualizando continuamente, se necesitará instalar un gestor de contenidos (CMS) a través del que el administrador puede ir realizando los cambios y actualizaciones él mismo.

Estos gestores son intuitivos y muy sencillos de gestionar. Algunos ejemplos de gestores de contenidos son:

- **WordPress:** Sin duda es el más extendido de los gestores de contenidos. Existe mucha información en la red, tutoriales y guías para personalizarlo, entenderlo y además es gratuito.
- **Joomla:** Es el segundo en el top CMS, tras WordPress. Aunque no goza de tantos usuarios si que tiene una comunidad potente.
- **Drupal:** Es un CSM de software libre. Es muy adaptable, y recomendado especialmente para generar comunidades.

Este tipo de aplicación web es muy común entre páginas de contenidos: blogs personales, blogs corporativos, blogs profesionales, páginas de noticias, de artículos, de medios de comunicación, etc...

Otras consideraciones en el desarrollo de aplicaciones web

Cada tipo de aplicación web tendrás sus fortalezas y debilidades, pero recordamos que a fin de cuentas sigue siendo una web, no una app nativa. Aunque en apariencia sean parecidas (eso dependerá del diseño de aplicaciones web, no del desarrollo). De modo que tendrás que respetar la normativa de cookies y fortalece la seguridad a posibles ataques de hackers, de forma similar a una web.

Del mismo modo recuerda trabajar correctamente el SEO (no el posicionamiento ASO), ya que la aplicación web aparecerá en los resultados de búsqueda de motores como Google o Yahoo; especialmente desde la última actualización del algoritmo de Google. Las aplicaciones web buscan tener apariencia de app móvil, pero con tecnología web que permite abaratar considerablemente los gastos, por lo que tendrás que trabajarla como tal. [12]

5.2 Arquitectura de sistemas distribuidos

5.2.1 Definición:

Los recursos de diferentes máquinas en red se integran de forma que desaparece la dualidad local/remoto. La diferencia fundamental con los sistemas en red es que la ubicación del recurso es transparente a las

aplicaciones y usuarios, por lo que, desde este punto de vista, no hay diferencia con un sistema de tiempo compartido. El usuario accede a los recursos del sistema distribuido a través de una interfaz gráfica de usuario desde un terminal, despreocupándose de su localización. Las aplicaciones ejecutan una interfaz de llamadas al sistema como si de un sistema centralizado se tratase, por ejemplo POSIX. Un servicio de invocación remota (por ejemplo a procedimientos, RPC, o a objetos, RMI) resuelve los accesos a los recursos no locales utilizando para ello la interfaz de red. Los sistemas distribuidos proporcionan de forma transparente la compartición de recursos, facilitando el acceso y la gestión, e incrementando la eficiencia y la disponibilidad.

El modelo de sistema distribuido es el más general, por lo que, aunque no se ha alcanzado a nivel comercial la misma integración para todo tipo de recursos, la tendencia es clara a favor de este tipo de sistemas. La otra motivación es la relación de costes a la que ha llevado la evolución tecnológica en los últimos años. Hoy en día existe un hardware estándar de bajo coste, los ordenadores personales, que son los componentes básicos del sistema. Por otra parte, la red de comunicación, a no ser que se requieran grandes prestaciones, tampoco constituye un gran problema económico, pudiéndose utilizar infraestructura cableada ya existente (Ethernet, la red telefónica, o incluso la red eléctrica) o inalámbrica.

5.2.2 Propiedades de los sistemas distribuidos

Un sistema distribuido que pretenda ofrecer una visión de sistema único deberá cumplir las propiedades que se presentan a continuación:

- a. **Transparencia:** El objetivo esencial de un sistema distribuido es proporcionar al usuario y a las aplicaciones una visión de los recursos del sistema como gestionados por una sola máquina virtual. La distribución física de los recursos es transparente.
- b. **Escalabilidad:** Una de las características de los sistemas distribuidos es su modularidad, lo que le permite una gran flexibilidad y posibilita su escalabilidad, definida como la capacidad del sistema para crecer sin aumentar su complejidad ni disminuir su rendimiento. Uno de los objetivos del diseño de un sistema distribuido es extender la escalabilidad a la integración de servicios.

La escalabilidad presenta dos aspectos. El sistema distribuido debe (1) proporcionar espacios de nombres suficientemente amplios, de forma que no supongan una

limitación inherente, y (2) mantener un buen nivel de rendimiento en el acceso a los recursos cuando el sistema crece.

- c. **Fiabilidad y tolerancia a fallos:** La fiabilidad de un sistema puede definirse como su capacidad para realizar correctamente y en todo momento las funciones para las que se ha diseñado.

- d. **Consistencia:** La distribución de recursos introduce importantes beneficios. Por una parte, contribuye al incremento del rendimiento a través del paralelismo y promoviendo el acceso a copias locales del recurso (disminuyendo los costes de comunicación). Por otra, como se acaba de ver, la replicación aumenta la disponibilidad, siendo la base para proporcionar tolerancia a fallos. No obstante, distribuir recursos acarrea algunos problemas. Por una parte, la red de interconexión es una nueva fuente de fallos. Además, la seguridad del sistema es más vulnerable ante accesos no permitidos. Pero el problema de mayor complejidad es el de la gestión del estado global para evitar situaciones de inconsistencia entre los componentes del sistema. Este es un aspecto fundamental en el diseño del sistema distribuido, por lo que lo comentaremos a continuación. . [13]

5.3 Sistemas Distribuidos

Un sistema distribuido se define como una colección de computadores autónomos conectados por una red, y con el software distribuido adecuado para que el sistema sea visto por los usuarios como una única entidad capaz de proporcionar facilidades de computación.

El desarrollo de los sistemas distribuidos vino de la mano de las redes locales de alta velocidad a principios de 1970. Más recientemente, la disponibilidad de computadoras personales de altas prestaciones, estaciones de trabajo y ordenadores servidores ha resultado en un mayor desplazamiento hacia los sistemas distribuidos en detrimento de los ordenadores centralizados multiusuario. Esta tendencia se ha acelerado por el desarrollo de software para sistemas distribuidos, diseñado para soportar el desarrollo de aplicaciones distribuidas. Este software permite a los ordenadores coordinar sus actividades y compartir los recursos del sistema – hardware, software y datos.

Los sistemas distribuidos se implementan en diversas plataformas hardware, desde unas pocas estaciones de trabajo conectadas por una red de área local, hasta Internet, una colección de redes de área local y de área extensa interconectados, que enlazan millones de ordenadores.

Las aplicaciones de los sistemas distribuidos varían desde la provisión de capacidad de cómputo a grupos de usuarios, hasta sistemas bancarios, comunicaciones multimedia y abarcan prácticamente todas las aplicaciones comerciales y técnicas de los ordenadores. Los requisitos de dichas aplicaciones incluyen un alto nivel de fiabilidad, seguridad contra interferencias externas y privacidad de la información que el sistema mantiene. Se deben proveer accesos concurrentes a bases de datos por parte de muchos usuarios, garantizar tiempos de respuesta, proveer puntos de acceso al servicio que están distribuidos geográficamente, potencial para el crecimiento del sistema para acomodar la expansión del negocio y un marco para la integración de sistemas usados por diferentes compañías y organizaciones de usuarios.

5.3.1 Características clave de los sistemas distribuidos

George Coulouris establece que son seis las características principales responsables de la utilidad de los sistemas distribuidos. Se trata de comparación de recursos, apertura (openness), concurrencia, escalabilidad, tolerancia a fallos y transparencia. En las siguientes líneas trataremos de abordar cada una de ellas.

5.3.2 Compartición de Recursos

El término 'recurso' es bastante abstracto, pero es el que mejor caracteriza el abanico de entidades que pueden compartirse en un sistema distribuido. El abanico se extiende desde componentes hardware como discos e impresoras hasta elementos software como ficheros, ventanas, bases de datos y otros objetos de datos.

La idea de compartición de recursos no es nueva ni aparece en el marco de los sistemas distribuidos. Los sistemas multiusuario clásicos desde siempre han provisto compartición de recursos entre sus usuarios. Sin embargo, los recursos de una computadora multiusuario se comparten de manera natural entre todos sus usuarios. Por el contrario, los usuarios de estaciones de trabajo monousuario o computadoras personales dentro de un sistema distribuido no obtienen automáticamente los beneficios de la compartición de recursos.

Los recursos en un sistema distribuido están físicamente encapsulados en una de las computadoras y sólo pueden ser accedidos por otras computadoras mediante las comunicaciones (la red). Para que la compartición de recursos sea efectiva, ésta debe ser manejada por un programa que ofrezca un interfaz de comunicación permitiendo que el recurso sea accedido, manipulado y actualizado de una manera fiable y consistente. Surge el término genérico de gestor de recursos.

Un gestor de recursos es un módulo software que maneja un conjunto de recursos de un tipo en particular. Cada tipo de recurso requiere algunas políticas y métodos específicos junto con requisitos comunes para todos ellos. Éstos incluyen la provisión de un esquema de nombres para cada clase de recurso, permitir que los recursos individuales sean accedidos desde cualquier localización; la traslación de nombre de recurso a direcciones de comunicación y la coordinación de los accesos concurrentes que cambian el estado de los recursos compartidos para mantener la consistencia.

Un sistema distribuido puede verse de manera abstracta como un conjunto de gestores de recursos y un conjunto de programas que usan los recursos. Los usuarios de los recursos se comunican con los gestores de los recursos para acceder a los recursos compartidos del sistema. Esta perspectiva nos lleva a dos modelos de sistemas distribuidos: el modelo cliente-servidor y el modelo basado en objetos.

5.3.3 Apertura (openness)

Un sistema informático es abierto si el sistema puede ser extendido de diversas maneras. Un sistema puede ser abierto o cerrado con respecto a extensiones hardware (añadir periféricos, memoria o interfaces de comunicación, etc...) o con respecto a las extensiones software (añadir características al sistema operativo, protocolos de comunicación y servicios de compartición de recursos, etc...). La apertura de los sistemas distribuidos se determina primariamente por el grado hacia el que nuevos servicios de compartición de recursos se pueden añadir sin perjudicar ni duplicar a los ya existentes.

5.3.4 Básicamente los sistemas distribuidos cumplen una serie de características:

- ❖ Los interfaces software clave del sistema están claramente especificados y se ponen a disposición de los desarrolladores. En una palabra, los interfaces se hacen públicos.
- ❖ Los sistemas distribuidos abiertos se basan en la provisión de un mecanismo uniforme de comunicación entre procesos e interfaces publicados para acceder a recursos compartidos.
- ❖ Los sistemas distribuidos abiertos pueden construirse a partir de hardware y software heterogéneo, posiblemente proveniente de vendedores diferentes. Pero la conformidad de cada

componente con el estándar publicado debe ser cuidadosamente comprobada y certificada si se quiere evitar tener problemas de integración.

a. Concurrencia

Cuando existen varios procesos en una única máquina decimos que se están ejecutando concurrentemente. Si el ordenador está equipado con un único procesador central, la concurrencia tiene lugar entrelazando la ejecución de los distintos procesos. Si la computadora tiene N procesadores, entonces se pueden estar ejecutando estrictamente a la vez hasta N procesos.

En los sistemas distribuidos hay muchas máquinas, cada una con uno o más procesadores centrales. Es decir, si hay M ordenadores en un sistema distribuido con un procesador central cada una entonces hasta M procesos estar ejecutándose en paralelo.

En un sistema distribuido que está basado en el modelo de compartición de recursos, la posibilidad de ejecución paralela ocurre por dos razones:

- A. Muchos usuarios interactúan simultáneamente con programas de aplicación.
- B. Muchos procesos servidores se ejecutan concurrentemente, cada uno respondiendo a diferentes peticiones de los procesos clientes.

El caso (1) es menos conflictivo, ya que normalmente las aplicaciones de interacción se ejecutan aisladamente en la estación de trabajo del usuario y no entran en conflicto con las aplicaciones ejecutadas en las estaciones de trabajo de otros usuarios.

El caso (2) surge debido a la existencia de uno o más procesos servidores para cada tipo de recurso. Estos procesos se ejecutan en distintas máquinas, de manera que se están ejecutando en paralelo diversos servidores, junto con diversos programas de aplicación. Las peticiones para acceder a los recursos de un servidor dado pueden ser encoladas en el servidor y ser procesadas secuencialmente o bien pueden ser procesadas varias concurrentemente por múltiples instancias del proceso gestor de recursos. Cuando esto ocurre los procesos servidores deben sincronizar sus acciones para asegurarse de que no existen conflictos. La sincronización debe ser cuidadosamente planeada para asegurar que no se pierden los beneficios de la concurrencia.

b. Escalabilidad

Los sistemas distribuidos operan de manera efectiva y eficiente a muchas escalas diferentes. La escala más pequeña consiste en dos estaciones de trabajo y un servidor de ficheros, mientras que un sistema distribuido construido alrededor de una red de área local simple podría contener varios cientos de estaciones de trabajo, varios servidores de ficheros, servidores de impresión y otros servidores de propósito específico. A menudo se conectan varias redes de área local para formar internetworks, y éstas podrían contener muchos miles de ordenadores que forman un único sistema distribuido, permitiendo que los recursos sean compartidos entre todos ellos.

Tanto el software de sistema como el de aplicación no deberían cambiar cuando la escala del sistema se incrementa. La necesidad de escalabilidad no es solo un problema de prestaciones de red o de hardware, sino que está íntimamente ligada con todos los aspectos del diseño de los sistemas distribuidos. El diseño del sistema debe reconocer explícitamente la necesidad de escalabilidad o de lo contrario aparecerán serias limitaciones.

La demanda de escalabilidad en los sistemas distribuidos ha conducido a una filosofía de diseño en que cualquier recurso simple -hardware o software- puede extenderse para proporcionar servicio a tantos usuarios como se quiera. Esto es, si la demanda de un recurso crece, debería ser posible extender el sistema para darla servicio. Por ejemplo, la frecuencia con la que se accede a los ficheros crece cuando se incrementa el número de usuarios y estaciones de trabajo en un sistema distribuido. Entonces, debe ser posible añadir ordenadores servidores para evitar el cuello de botella que se produciría si un solo servidor de ficheros tuviera que manejar todas las peticiones de acceso a los ficheros. En este caso el sistema deberá estar diseñado de manera que permita trabajar con ficheros replicados en distintos servidores, con las consideraciones de consistencias que ello conlleva.

Cuando el tamaño y complejidad de las redes de ordenadores crece, es un objetivo primordial diseñar software de sistema distribuido que seguirá siendo eficiente y útil con esas nuevas configuraciones de la red. Resumiendo, el trabajo necesario para procesar una petición simple para acceder a un recurso compartido debería ser prácticamente independiente del tamaño de la red. Las técnicas necesarias para conseguir estos objetivos incluyen el uso de datos replicados, la técnica asociada de caching, y el uso de múltiples servidores para manejar ciertas tareas, aprovechando la concurrencia para permitir una mayor productividad. Una explicación completa de estas técnicas puede encontrarse en *DISTRIBUTED SYSTEMS Concepts and Design*, George Coulouris 1994. [14]

c. Tolerancia a Fallos

Los sistemas informáticos a veces fallan. Cuando se producen fallos en el software o en el hardware, los programas podrían producir resultados incorrectos o podrían pararse antes de terminar la computación que estaban realizando. El diseño de sistemas tolerantes a fallos se basa en dos cuestiones, complementarias entre sí: Redundancia hardware (uso de componentes redundantes) y recuperación del software (diseño de programas que sean capaces de recuperarse de los fallos).

En los sistemas distribuidos la redundancia puede plantearse en un grano más fino que el hardware, pueden replicarse los servidores individuales que son esenciales para la operación continuada de aplicaciones críticas.

La recuperación del software tiene relación con el diseño de software que sea capaz de recuperar (rollback) el estado de los datos permanentes antes de que se produjera el fallo.

Los sistemas distribuidos también proveen un alto grado de disponibilidad en la vertiente de fallos hardware. La disponibilidad de un sistema es una medida de la proporción de tiempo que está disponible para su uso. Un fallo simple en una maquina multiusuario resulta en la no disponibilidad del sistema para todos los usuarios. Cuando uno de los componentes de un sistema distribuidos falla, solo se ve afectado el trabajo que estaba realizando el componente averiado. Un usuario podría desplazarse a otra estación de trabajo; un proceso servidor podría ejecutarse en otra máquina.

d. Transparencia

La transparencia se define como la ocultación al usuario y al programador de aplicaciones de la separación de los componentes de un sistema distribuido, de manera que el sistema se percibe como un todo, en vez de una colección de componentes independientes. La transparencia ejerce una gran influencia en el diseño del software de sistema.

El manual de referencia RM-ODP [ISO 1996a] identifica ocho formas de transparencia. Estas proveen un resumen útil de la motivación y metas de los sistemas distribuidos. Las transparencias definidas son:

- **Transparencia de Acceso:** Permite el acceso a los objetos de información remotos de la misma forma que a los objetos de información locales.

- **Transparencia de Localización:** Permite el acceso a los objetos de información sin conocimiento de su localización
- **Transparencia de Concurrencia:** Permite que varios procesos operen concurrentemente utilizando objetos de información compartidos y de forma que no exista interferencia entre ellos.
- **Transparencia de Replicación:** Permite utilizar múltiples instancias de los objetos de información para incrementar la fiabilidad y las prestaciones sin que los usuarios o los programas de aplicación tengan por que conoces la existencia de las réplicas.
- **Transparencia de Fallos:** Permite a los usuarios y programas de aplicación completar sus tareas a pesar de la ocurrencia de fallos en el hardware o en el software.
- **Transparencia de Migración:** Permite el movimiento de objetos de información dentro de un sistema sin afectar a los usuarios o a los programas de aplicación.
- **Transparencia de Prestaciones.** Permite que el sistema sea reconfigurado para mejorar las prestaciones mientras la carga varia.
- **Transparencia de Escalado:** Permite la expansión del sistema y de las aplicaciones sin cambiar la estructura del sistema o los algoritmos de la aplicación.

Las dos más importantes son las transparencias de acceso y de localización; su presencia o ausencia afecta fuertemente a la utilización de los recursos distribuidos. A menudo se las denomina a ambas transparencias de red. La transparencia de red provee un grado similar de anonimato en los recursos al que se encuentra en los sistemas centralizados.

5.4 El Modelo Cliente Servidor

El modelo cliente-servidor de un sistema distribuido es el modelo más conocido y más ampliamente adoptado en la actualidad. Hay un conjunto de procesos servidores, cada uno actuando como un gestor de recursos para una colección de recursos de un tipo, y una colección de procesos clientes, cada uno llevando a cabo una tarea que requiere acceso a algunos recursos hardware y software compartidos. Los gestores de recursos a su vez podrían necesitar acceder a recursos compartidos manejados por otros procesos, así que algunos procesos son ambos clientes y servidores. En el modelo, cliente-servidor, todos los recursos compartidos son mantenidos y manejados por los procesos servidores. Los procesos clientes realizan peticiones a los servidores cuando necesitan acceder a algún recurso. Si la petición es válida, entonces el servidor lleva a cabo la acción requerida y envía una respuesta al proceso cliente.

El termino proceso se usa aquí en el sentido clásico de los sistemas operativos. Un proceso es un programa en ejecución. Consiste en un entorno de ejecución con al menos un thread de control.

El modelo cliente-servidor nos da un enfoque efectivo y de propósito general para la compartición de información y de recursos en los sistemas distribuidos. El modelo puede ser implementado en una gran variedad de entornos software y hardware. Las computadoras que ejecuten los programas clientes y servidores pueden ser de muchos tipos y no existe la necesidad de distinguir entre ellas; los procesos cliente y servidor pueden incluso residir en la misma máquina.

En esta visión simple del modelo cliente-servidor, cada proceso servidor podría ser visto como un proveedor centralizado de los recursos que maneja. La provisión de recursos centralizada no es deseable en los sistemas distribuidos. Es por esta razón por lo que se hace una distinción entre los servicios proporcionados a los clientes y los servidores encargados de proveer dichos servicios. Se considera un servicio como una entidad abstracta que puede ser provista por varios procesos servidores ejecutándose en computadoras separadas y cooperando vía red.

El modelo cliente-servidor se ha extendido y utilizado en los sistemas actuales con servicios manejando muchos diferentes tipos de recursos compartidos – correo electrónico y mensajes de noticias, ficheros, sincronización de relojes, almacenamiento en disco, impresoras, comunicaciones de área extensa, e incluso las interfaces gráficas de usuario. Pero no es posible que todos los recursos que existen en un sistema distribuido sean manejados y compartidos de esta manera; algunos tipos de recursos deben permanecer locales a cada computadora de cara a una mayor eficiencia – RAM, procesador, interfaz de red local -. Estos recursos clave son manejados separadamente por un sistema operativo en cada máquina; solo podrían ser compartidos entre procesos localizados en el mismo ordenador.

Aunque el modelo cliente-servidor no satisface todos los requisitos necesarios para todas las aplicaciones distribuidos, es adecuado para muchas de las aplicaciones actuales y provee una base efectiva para los sistemas operativos distribuidos de propósito general. [15]

5.4.1 Middleware

El término middleware se discute en Component-Based Client/Server Computing, Lewandowki 1998. El software distribuido requerido para facilitar las interacciones cliente-servidor se denomina middleware. El acceso transparente a servicios y recursos no locales distribuidos a través de una red se provee a través del middleware, que sirve como marco para las comunicaciones entre las porciones cliente y servidor de un sistema.

El middleware define: el API que usan los clientes para pedir un servicio a un servidor, la transmisión física de la petición vía red, y la devolución de resultados desde el servidor al cliente. Ejemplos de middleware estándar para dominios específicos incluyen: ODBC, para bases de datos, Lotus para groupware, HTTP y SSL para Internet y CORBA, DCOM y JAVA RMI para objetos distribuidos.

El middleware fundamental o genérico es la base de los sistemas cliente-servidor. Los servicios de autenticación en red, llamadas a procedimiento remoto, sistemas de ficheros distribuidos y servicios de tiempo en red se consideran parte del middleware genérico. Este tipo de middleware empieza a ser parte estándar de los sistemas operativos modernos como Windows NT. En sistemas donde no se disponga deberá recurrirse a middleware del tipo OSD DCE (Distributed Computing Environment) [OSF 1994] El middleware específico para un dominio complementa al middleware genérico de cara a aplicaciones mucho más específicas. [16]

El protocolo de comunicaciones más usado por el middleware, tanto genérico como específico, es TCP/IP. Esto se debe a su amplia difusión en todos los sistemas operativos del mercado y en especial en los ordenadores personales. [17]

5.4.2 Middleware para sistemas cliente-servidor: RPC

Un servicio proporcionado por un servidor no es más que un conjunto de operaciones disponibles para los clientes. El acceso al servicio se realiza mediante un protocolo de peticiones respuesta con llamadas bloqueantes. Ejemplo: Un servicio de ficheros. El servidor mantiene como recurso compartido los ficheros. Sobre el recurso compartido se pueden realizar diversas operaciones: Crear, Abrir, Leer, etc.

Los mecanismos RPC persiguen que los clientes se abstraigan e invoquen procedimientos remotos (operaciones) para obtener servicios. Así, el procedimiento llamado se ejecuta en otro proceso de otra

máquina (servidor). El objetivo de RPC es mantener la semántica de la llamada a procedimiento normal en un entorno de implementación totalmente distinto. La ventaja está en que el desarrollador se preocupa de los interfaces que soporta el servidor. Para especificar dichos interfaces se dispone de un IDL (lenguaje de definición de interfaces).

Los sistemas RPC disponen de mecanismos de RPC integrados en un lenguaje de programación particular que incluye además una notación para definir interfaces entre clientes y servidores (IDL específico). Un IDL permite definir el nombre de las operaciones soportadas por el servidor y sus parámetros (tipo y dirección). También se deben proveer mecanismos para manejo de excepciones, garantizar la ejecución de las operaciones, así como la detección de fallos. Todo ello de la forma más transparente posible.

El software (middleware) que soporta RPC tiene tres tareas fundamentales:

- a) Procesamiento relacionado con los interfaces: Integrar RPC en el entorno de programación, empaquetamiento (marshalling)/desempaquetamiento (unmarshalling) y despachar las peticiones al procedimiento adecuado.
- b) Gestionar las comunicaciones
- c) Enlazado (Binding): Localizar al servidor de un servicio.

La interfaz no es más que un número de procedimiento acordado entre cliente y servidor. Este número viaja dentro del mensaje RPC transmitido por la red. En la parte cliente existe un procedimiento de 'stub' encargado de empaquetar/desempaquetar los argumentos de la llamada, convertir la llamada local en una llamada remota. Esto supone enviar un mensaje, esperar la respuesta y retornar los resultados. En la parte del servidor está el despachador, junto con el conjunto de procedimientos de stub de servidor, que tienen una misión similar a los de la parte cliente. El despachador selecciona el procedimiento de stub adecuado a partir del número de procedimiento requerido.

El compilador de IDL genera los procedimientos de stub de cliente y de servidor, las operaciones de empaquetamiento y desempaquetamiento así como los ficheros de cabecera necesarios.

Las peticiones de los clientes son con respecto a un nombre de servicio. En última instancia deben ser dirigidas a un puerto en el servidor. En un sistema distribuido un binder es un servicio separado que

mantiene una tabla que contiene correspondencias de nombres de servicios con puertos de servidor. Se trata, como veremos más adelante, de un servicio de nombres. Importar un servicio es pedir al binder que busque el nombre del interfaz y retorne el puerto del servidor. Exportar un servicio es registrarlo de cara al binder. El binder deberá estar en un puerto bien conocido o, al menos, se le podrá localizar.

Una implementación bastante habitual de RPC es la de SUN. Incorpora todo un conjunto de primitivas para trabajar con RPC en lenguaje C. Dispone de una representación neutral de los datos para su empaquetamiento, XDR (External Data Representation). XDR también se denomina al IDL que se proporciona.

El compilador de IDL, que se denomina `rpcgen`, genera los procedimientos de `stub`, el procedimiento `main` del servidor y el despachador, el código de conversión de los parámetros a XDR, así como los ficheros de cabecera correspondientes.

El binder recibe el nombre de `port mapper`. Cuando se activa un servidor en una máquina remota ésta se registra con el `port mapper`, obteniendo un puerto de comunicación a través del cual escuchar. Cuando se quiere acceder a un servicio hay que contactar con el `port mapper` de la máquina remota, preguntándole por el nombre de un determinado servicio, devolviéndose el puerto de comunicación en el que está a la escucha el servidor correspondiente. Por tanto para acceder a una determinada operación de un servicio hace falta la siguiente información: `[host:servicio:procedimiento]`.

5.4.3 Servidores Pesados vs Clientes Pesados

Lewandowski realiza una discusión de este concepto dentro del marco de los sistemas cliente-servidor. Los especialistas en sistemas de información califican como ‘pesada’ (*fat*) una parte de un sistema con una cantidad de funcionalidad desproporcionada. Por el contrario, una parte de sistema se considera ligera (*thin*) si tiene menos responsabilidades.

En un sistema cliente-servidor la porción del servidor casi siempre mantiene los datos, mientras que la porción del cliente es responsable de la interfaz de usuario. El desplazamiento de la lógica de la aplicación constituye la distinción entre clientes ‘pesados’ y servidores ‘pesados’. Los sistemas servidores ‘pesados’ delegan más responsabilidad de la lógica de la aplicación en los servidores, mientras que los clientes ‘pesados’ dan al cliente mayor responsabilidad. Ejemplo de servidor pesado es un

servidor Web, mientras que muchos de los clientes en sistemas de bases de datos constituyen clientes ‘pesados’.

Aunque los sistemas basados en servidores ‘pesados’ han sido los más utilizados en el pasado, en la actualidad muchos diseñadores prefieren sistemas con clientes ‘pesados’, debido a que son más fáciles de implementar. Los clientes ‘pesados’ permiten a los usuarios crear aplicaciones y modificar los *frontend* del sistema fácilmente, pero a costa de reducir la encapsulación de los datos; cuanto más responsabilidad se coloque en un cliente, el cliente requerirá un conocimiento más íntimo de la organización de los datos del servidor.

Por otra parte, un servidor ‘pesado’ es más fácilmente explotable, esto es, más fácil de explotar. Además este tipo de servidor asegura una mayor compatibilidad entre clientes y servidores. Por ejemplo, una página Web diseñada bajo este modelo supondría que no hay disponibilidad de applets de Java, plugins o ActiveX’s debido a que el usuario está usando un cliente ‘ligero’, (un navegador básico), y el servidor estaría restringido al estándar HTML 2.0. El uso de este modelo de cliente ‘ligero’ asegura que todos los usuarios visualizan una página “aceptable”, aunque no se pueden proveer las características avanzadas disponibles con un cliente ‘pesado’. [18].

5.4.4 Sistemas N-Tiered

El modelo canónico cliente-servidor asume exactamente dos participantes discretos en el sistema. Se denomina sistema ‘two-tier’; la lógica de la aplicación puede estar en el cliente, el servidor, o compartida entre los dos. También puede darse el caso de tener la lógica de la aplicación separada de los datos y de la interfaz de usuario, convirtiéndose el sistema en ‘three-tiered’. En un sistema ‘threetiered’ ideal toda la lógica de la aplicación reside en una capa separada. Esto ocurre raramente en los sistemas actuales. Siempre hay ciertas partes que permanecen o bien del lado del cliente o del lado del servidor.

Las aplicaciones Web estándar son un ejemplo clásico de sistemas ‘three-tiered’. Por un lado se tienen la interfaz de usuario, provista por la interpretación de HTML, por un navegador. Los componentes embebidos visualizados por el navegador residen en la capa media; pueden ser applets de Java, ActiveX’s o cualquier otra clase de entidad que provea lógica de aplicación para el sistema. Por último se tienen los datos suministradores por el servidor Web.

5.4.5 Ventajas de los sistemas cliente-servidor

La principal ventaja de los sistemas cliente-servidor está en la correspondencia natural de las aplicaciones en el marco cliente-servidor. Un ejemplo de esto es una agenda electrónica. Debido a que los datos son relativamente estáticos y son vistos de manera uniforme por todos los usuarios del sistema parece lógico colocarlos en un servidor que acepte peticiones sobre dichos datos. Es más, en este caso la lógica de aplicación debería estar colocada del lado del servidor, para proporcionar una mayor flexibilidad al sistema de búsquedas (cambios en los algoritmos, etcétera...).

Como resultado de la disponibilidad de middleware compatible para múltiples plataformas y de los avances recientes de la interoperabilidad binaria, los sistemas cliente-servidor pueden conectar clientes ejecutándose en una plataforma con servidores ejecutándose en otra plataforma completamente distinta. Las tecnologías como Java y los ORBs (Object Request Brokers), de los que trata en profundidad este trabajo, esperan proveer una total integración de todas las plataformas en unos pocos años. Si las porciones de un sistema cliente-servidor encapsulan una única función y siguen un interfaz perfectamente definido, aquellas partes del sistema que proveen los servicios pueden ser intercambiadas sin afectar a otras porciones del sistema. Esto permite a los usuarios, desarrolladores y administradores adecuar el sistema con sus necesidades en cada momento.

Otra ventaja es la posibilidad de ejecutar aplicaciones que hacen uso intensivo de los recursos en plataformas hardware de bajo coste. También el sistema es más escalable, pudiéndose añadir tanto nuevo clientes como nuevos servidores.

5.5 Sitios web

Conjunto organizado y coherente de páginas Web que tiene como función ofrecer, informar, publicitar o vender contenidos, productos y servicios al resto del mundo. Para que un sitio Web pueda ser visitado por otras personas es necesario que se encuentre alojado en un servidor. Se trata de una computadora conectada a la World Wide Web con espacio en disco y conectividad suficiente para albergar sitios y servirlos al resto de la comunidad de usuarios de Internet a través de direcciones IP o nombres de dominio. [19]

5.5.1 Ventajas de un sitio Web

Un sitio Web de calidad le permitirá mejorar su imagen profesional, demostrando por encima de todo, el conocimiento y la aceptación de las nuevas tecnologías. Además, cada vez son más los miles de usuarios que acceden a Internet buscando información, un producto o servicio. Por eso, no tener una Web supone una gran desventaja competitiva con respecto a otras empresas que sí la tienen.

Internet se ha convertido en la forma más eficiente de transmitir información. Permite el contacto directo entre su empresa, cliente y proveedor de una manera más rápida y económica. Toda la información que usted incluya en su sitio Web estará disponible las 24 horas del día, 7 días a la semana y 365 días al año.

5.5.2 Domain Name Server (DNS)

Servidor de nombres. Es el equipo que resuelve los nombres de dominios indicando en donde se encuentran ubicados, Si usted contrata el servicio de Hosting con Azteca Hosting debe asociar su dominio con los DNS de Azteca Hosting.

5.5.3 Servicio de Hosting

El servicio de Hosting implica el alquiler de un espacio en el disco rígido de un servidor conectado directamente a la red de Internet para alojar su sitio web.

Cuanto mejor sea el servidor donde usted aloja su página Web usted se beneficiará con mayores prestaciones y mejor tiempo de respuesta de su sitio hacia sus visitantes en la web.

5.5.4 Dominio

Un nombre de Dominio es un nombre único proporcionado a una persona, empresa u organización, de forma que cuando un usuario teclea dicho nombre en un navegador de Internet, el navegador muestra la información específica para esa persona, empresa u organización. Algunos ejemplos nos ayudan a entenderlo mejor: Empresa Dominio IBM ibm.com Oracle oracle.com BBC bbc.co.uk Los nombres de dominio son familiares y fáciles de recordar. En el campo técnico, cada nombre de dominio está relacionado con una dirección IP, (poco amigable y difícil de recordar).

5.5.5 Redirección de Dominio

Una redirección permite que 2 o más dominios compartan la misma cuenta de hosting mostrando el mismo contenido. Suele ser la opción ideal para economizar cuando se quiere poder acceder al mismo sitio desde distintas direcciones en Internet.

5.5.6 Diferencia un Hosting de un Dominio

Un nombre de dominio es tan solo una dirección que nos permite acceder a la información proporcionada por la empresa o particular dueño del dominio. Por otro lado un Hosting o Alojamiento Web, permite almacenar la información que se mostrará cuando un usuario teclea un dominio en un navegador.

Ambos servicios, Dominio y Hosting son complementarios. Para que su página personal o su empresa se muestren en Internet necesitará ambos servicios.

5.5.7 File Transfer Protocol (FTP)

FTP son siglas que vienen del inglés: File Transfer Protocol o Protocolo para la transferencia de Ficheros.

FTP es básicamente un sistema que permite colocar los archivos de tu página Web en tu espacio de hosting. Muchos de los programas de diseño Web disponen de un cliente FTP incorporado. Así es el caso de (Dreamweaver), (NetObjects) y otros.

También están disponibles clientes FTP en el mercado. Desde utilidades gratuitas hasta de elevado precio.

POP3 (Post office protocol -versión 3)

Es un protocolo cliente/servidor que permite a su programa de correo interpretar y almacenar su cuenta e-mail, lo cual permite que usted pueda conectarse con el servidor POP3 en cualquier momento y leer su correo desde (Outlook), (Eudora), etc.

5.5.8 Lista de correo y sus ventajas

Pueden ser definidas como un sistema que distribuye mensajes a un grupo de personas que comparten un interés común pero que no necesariamente se conocen entre sí. Constituyen una importante forma de comunicación vía e-mail.

Se basan en un sistema centralizado (un programa que reside en un servidor) que realiza un proceso llamado "mailexplósión".

Con que solo enviemos un e-mail a la dirección de la lista, el programa se encarga de re-enviar el email a cada uno de los miembros de esa lista. Es decir que cualquier miembro de la lista puede mandar una sola copia del mensaje y esta llegará a todos los suscriptos.

Este proceso se centraliza para no tener que enviar tantos e-mails como miembros tenga la lista y para que el listado de quienes forman parte de la lista no tenga que ser mantenido por todos sus miembros, lo

que acarrearía errores ya que si alguien se quiere suscribir o borrar debería pedirle a todos los miembros que lo agreguen o borren y alguno se olvidaría de hacerlo. El resultado sería que al poco tiempo ninguna lista de suscriptos coincidiría.

5.5.9 Banner

El Banner es un gráfico con publicidad que aparece en la pantalla del navegante, los (banner) pueden ser estáticos o con movimiento. [19]

5.6 Etapa de análisis

En esta sección del documento se definen los conceptos de las herramientas utilizadas en la etapa de análisis que es la primera etapa en la que se cumple en la definición del ciclo de vida y desarrollo del software, con la definición de estas herramientas podemos comprender de donde hemos obtenido la información básica inicial necesaria para estructurar las bases de este proyecto.

5.6.1 Requerimientos del software

Normalmente, un tema de la Ingeniería de Software tiene diferentes significados. De las muchas definiciones que existen para requerimiento, a continuación se presenta la definición que aparece en el glosario de la IEEE.

- (1) Una condición o necesidad de un usuario para resolver un problema o alcanzar un objetivo.
- (2) Una condición o capacidad que debe estar presente en un sistema o componentes de sistema para satisfacer un contrato, estándar, especificación u otro documento formal.
- (3) Una representación documentada de una condición o capacidad como en (1) o (2).

Los requerimientos puedes dividirse en requerimientos funcionales y requerimientos no funcionales. Los requerimientos funcionales definen las funciones que el sistema será capaz de realizar. Describen las transformaciones que el sistema realiza sobre las entradas para producir salidas.

Los requerimientos no funcionales tienen que ver con características que de una u otra forma puedan limitar el sistema, como por ejemplo, el rendimiento (en tiempo y espacio), interfaces de usuario, fiabilidad (robustez del sistema, disponibilidad de equipo), mantenimiento, seguridad, portabilidad, estándares, etc.

5.6.2 Características de los requerimientos

Las características de un requerimiento son sus propiedades principales. Un conjunto de requerimientos en estado de madurez, deben presentar una serie de características tanto individualmente como en grupo. A continuación se presentan las más importantes. Necesario: Un requerimiento es necesario si su omisión provoca una deficiencia en el sistema a construir, y además su capacidad, características físicas o factor de calidad no pueden ser remplazados por otras capacidades del producto o del proceso. Conciso: Un requerimiento es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro. Completo: Un requerimiento está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión. Consistente: Un requerimiento es consistente si no es contradictorio con otro requerimiento. No ambiguo: Un requerimiento no es ambiguo cuando tiene una sola interpretación.

Verificable: Un requerimiento es verificable cuando puede ser cuantificado de manera que permita hacer uso de los siguientes métodos de verificación: inspección, análisis, demostración o pruebas.

5.6.3 Los requerimientos deben ser

- ❖ **Especificados por escrito.** Como todo contrato o acuerdo entre dos partes Posibles de probar o verificar. Si un requerimiento no se puede comprobar, entonces ¿cómo sabemos si cumplimos con él o no?
- ❖ **Descritos como una característica del sistema a entregar.** Esto es: que es lo que el sistema debe de hacer (y no como debe de hacerlo) Lo más abstracto y conciso posible. Para evitar malas interpretaciones.

5.6.4 Fundamentos del Análisis de Requerimientos

Definición: Es el conjunto de técnicas y procedimientos que nos permiten conocer los elementos necesarios para definir un proyecto de software. Es la etapa más crucial del desarrollo de un proyecto de software. La IEEE los divide en funcionales y no funcionales: Funcionales: Condición o capacidad de un sistema requerida por el usuario para resolver un problema o alcanzar un objetivo.

No Funcionales: Condición o capacidad que debe poseer un sistema para satisfacer un contrato, un estándar, una especificación u otro documento formalmente impuesto.

Para realizar bien el desarrollo de software es esencial realizar una especificación completa de los requerimientos de los mismos. Independientemente de lo bien diseñado o codificado que esté, un programa pobremente especificado decepcionará al usuario y hará fracasar el desarrollo.

La tarea de análisis de los requerimientos es un proceso de descubrimiento y refinamiento, El ámbito del programa, establecido inicialmente durante la ingeniería del sistema, es refinado en detalle. Se analizan y asignan a los distintos elementos de los programas las soluciones alternativas.

Tanto el que desarrolla el software como el cliente tienen un papel activo en la especificación de requerimientos. El cliente intenta reformular su concepto, algo nebuloso, de la función y comportamiento de los programas en detalles concretos, El que desarrolla el software actúa como interrogador, consultor y el que resuelve los problemas.

El análisis y especificación de requerimientos puede parecer una tarea relativamente sencilla, pero las apariencias engañan. Puesto que el contenido de comunicación es muy alto, abundan los cambios por mala interpretación o falta de información. El dilema con el que se enfrenta un ingeniero de software puede ser comprendido repitiendo la sentencia de un cliente anónimo: "Sé que crees que comprendes lo que piensas que he dicho, pero no estoy seguro de que lo que creíste oír sea lo que yo quise decir".

Los requerimientos de un sistema de software, cuando se ven en su conjunto son extensos y detallados, y además contienen múltiples relaciones entre sí. Lo que nos da a concluir, que el conjunto de requerimientos de un sistema computacional es complejo. Obtenemos la posibilidad de especificar sistemas complejos al documentar especificaciones simples y concisas para el sistema.

Esto se logra mediante al clasificar, estructurar y organizar todo lo que el sistema debe de hacer. En otras palabras al analizar sus requerimientos.

El análisis de requerimientos es la tarea que plantea la asignación de software a nivel de sistema y el diseño de programas. El análisis de requerimientos facilita al ingeniero de sistemas especificar la función y comportamiento de los programas, indicar la interfaz con otros elementos del sistema y establecer las ligaduras de diseño que debe cumplir el programa. El análisis de requerimientos permite al ingeniero refinar la asignación de software y representar el dominio de la información que será tratada por el programa. El análisis de requerimientos de al diseñador la representación de la información y las funciones que pueden ser traducidas en datos, arquitectura y diseño procedimental.

Finalmente, la especificación de requerimientos suministra al técnico y al cliente, los medios para valorar la calidad de los programas, una vez que se haya construido.

5.6.5 Tareas del Análisis

El análisis de requerimientos puede dividirse en cuatro áreas:

- ❖ Reconocimiento del problema
- ❖ Evaluación y síntesis
- ❖ Especificación ❖ Revisión.

5.6.6 Especificación de Requisitos de Software (SRS)

La especificación de requisitos de software es la actividad en la cual se genera el documento, con el mismo nombre, que contiene una descripción completa de las necesidades y funcionalidades del sistema que será desarrollado; describe el alcance del sistema y la forma en cómo hará sus funciones, definiendo los requerimientos funcionales y los no funcionales.

En la SRS se definen todos los requerimientos de hardware y software, diagramas, modelos de sistemas y cualquier otra información que sirva de soporte y guía para fases posteriores.

Es importante destacar que la especificación de requisitos es el resultado final de las actividades de análisis y evaluación de requerimientos; este documento resultante será utilizado como fuente básica de comunicación entre los clientes, usuarios finales, analistas de sistema, personal de pruebas, y todo aquel involucrado en la implementación del sistema.

Los clientes y usuarios utilizan la SRS para comparar si lo que se está proponiendo, coincide con las necesidades de la empresa. Los analistas y programadores la utilizan para determinar el producto que debe desarrollarse. El personal de pruebas elaborará las pruebas funcionales y de sistemas en base a este documento. Para el administrador del proyecto sirve como referencia y control de la evolución del sistema. La SRS posee las mismas características de los requerimientos: completa, consistente, verificable, no ambigua, factible, modificable, rastreable, precisa, entre otras. Para que cada característica de la SRS sea considerada, cada uno de los requerimientos debe cumplirlas; por ejemplo, para que una SRS se considere verificable, cada requerimiento definido en ella debe ser verificable; para que una SRS se considere modificable, cada requerimiento debe ser modificable y

así sucesivamente. Las características de la SRS son verificadas en la actividad de Validación, descrita en el punto.

La estandarización de la SRS es fundamental pues ayudará, entre otras cosas, a facilitar la lectura y escritura de la misma. Será un documento familiar para todos los involucrados, además de asegurar que se cubren todos los tópicos importantes.

Existen plantillas creadas para la SRS, sin embargo, cada uno tiene la potestad de crear su propia plantilla. [10]

5.7 Técnicas para hallar datos

Los analistas utilizan una variedad de métodos a fin de recopilar los datos sobre una situación existente, como entrevistas, cuestionarios, inspección de registros revisión en el sitio y observación.

Cada uno tiene ventajas y desventajas. Generalmente, se utilizan dos o tres para complementar el trabajo de cada una y ayudar a asegurar una investigación completa, en este para la realización de manuales de procesos y procedimientos de sistemas en una organización. [20]

5.7.1 Entrevista

Una Entrevista para la recopilación de información es una conversación dirigida con un propósito específico, que se basa en un formato de preguntas y respuestas.

En la entrevista se desea conocer tanto las opiniones como los sentimientos del entrevistado acerca del estado actual de los sistemas, sus metas personales, de la organización y de los procedimientos informales. Sobré todo esto, se busca la opinión de la persona entrevistada. Las opiniones pueden ser aún más importantes y reveladoras que los mismos hechos. Dentro de una organización, la entrevista es la técnica más significativa y productiva de que dispone el analista para recabar datos.

En otras palabras, la entrevista es un intercambio de información que se efectúa cara a cara. Es un canal de comunicación entre el analista y la organización; sirve para obtener información acerca de las necesidades y la manera de satisfacerlas, así como concejo y comprensión por parte del usuario para toda idea o método nuevo. Por otra parte, la entrevista ofrece al analista una excelente oportunidad para establecer una corriente de simpatía con el personal usuario, lo cual es fundamental en transcurso del estudio.

5.7.1.1 Resultados de la Entrevista

- Escribir los resultados (Documentación).
- Entregar una copia al entrevistado, solicitando su conformación, correcciones o adiciones. (Profesionalismo).
- Archivar los resultados de la entrevista para referencia y análisis posteriores (Documentación).

5.7.2 Observación

Otra técnica útil para el analista en su progreso de investigación, consiste en observar a las personas cuando efectúan su trabajo. Como técnica de investigación, la observación tiene amplia aceptación científica. Los sociólogos, psicólogos e ingenieros industriales utilizan extensamente ésta técnica con el fin de estudiar a las personas en sus actividades de grupo y como miembros de la organización. El propósito de la organización es múltiple: permite al analista determinar que se está haciendo, como se está haciendo, quien lo hace, cuando se lleva a cabo, cuanto tiempo toma, dónde se hace y por qué se hace. ¡Ver es creer! Observar las operaciones la proporciona el analista hechos que no podría obtener de otra forma.

5.7.3 Experiencia

Es una forma de conocimiento o habilidad derivados de la observación, de la vivencia de un evento o proveniente de las cosas que suceden en la vida. La historia de esta palabra se alinea con el concepto de experimento. Tanto el ser humano como también muchos animales pueden obtener esta forma de conocimiento llamada experiencia a lo largo de sus vidas. El concepto de experiencia generalmente se refiere al conocimiento procedimental (cómo hacer algo), en lugar del conocimiento factual (qué son las cosas). Los filósofos tratan el conocimiento basado en la experiencia como "conocimiento empírico" o "un conocimiento a posterior". Desde el punto de vista de la hermenéutica filosófica (Gadamer), solamente son posibles las experiencias si se tienen expectativas, por eso una persona de experiencia no es la que ha acumulado más vivencias (Erlebnis), sino la que está capacitada para permitirselas. La experiencia contribuye sensiblemente a la sabiduría. Aunque se puede obtener cierto grado de sabiduría al sufrir castigo u observar a otros recibirlo, una mejor manera de adquirir sabiduría, y que además ahorra tiempo, es beneficiarse y aprender de la experiencia de los que ya son sabios, prefiriendo su compañía a

la de “los inexpertos”. Una persona con considerable conocimiento en un área determinada puede ganar reputación como un experto. [22]

5.8 Diseño del sistema recomendado

En esta fase el analista utiliza la información recopilada en las primeras fases para realizar el diseño lógico del sistema de información. El analista diseña procedimientos precisos para la captura de datos que aseguran que los datos que ingresen al sistema de información sean correctos. Facilita la entrada eficiente de datos al sistema de información mediante técnicas adecuadas de diseño de formularios y pantallas. La concepción de la interfaz de usuario forma parte del diseño lógico del sistema de información. La interfaz conecta al usuario con el sistema y por tanto es sumamente importante. También incluye el diseño de archivos o bases de datos que almacenarán gran parte de los datos indispensables para los encargados de tomar las decisiones en la organización. [22].

5.8.1 Balsamiq Mockups

Es una herramienta que nos permite realizar Wireframes para webs fácilmente.

Un Wireframe (aplicado a la web) es una representación esquemática de la solución que llevaremos adelante, sin entrar en etapas posteriores como el diseño gráfico o la programación web. Podemos verlo como el esqueleto general visual de la solución. Nos permite acordar con el cliente aspectos clave de la solución a desarrollar, como la distribución general de los elementos, sus jerarquías y la navegación de los mismos.

Balsamiq Mockups nos provee de representaciones de todos los elementos utilizados para la construcción de una web, como pantallas de navegadores, títulos, menues, imágenes, videos, etc. Haciendo uso de ellos, sólo debemos organizarlos en un documento y ya podemos tener una primera aproximación de la solución a desarrollar.

Balsamiq Mockups es una herramienta que puede ser usada tanto por clientes como por desarrolladores, con el objetivo de acordar aspectos importantes del alcance de una solución con una mínima inversión de tiempo dedicada.

Los clientes pueden hacer uso de Balsamiq Mockups sin tener ningún tipo de conocimiento técnico especial. Gracias a ello, pueden comunicar de una manera más eficiente sus ideas y necesidades al grupo de trabajo que realizará las implementaciones. Se trata de una gran ayuda a la hora de definir el alcance de una solución.

Los desarrolladores pueden usar Balsamiq Mockups con el mismo propósito, pero al revés. Para comunicar rápidamente las propuestas de solución, sin invertir demasiada cantidad de tiempo en esta primera etapa (en el caso de un asesoramiento inicial no pago).

Inclusive durante el mismo proceso de trabajo, el equipo de desarrollo puede utilizar Balsamiq Mockups para generar versiones visuales sobre aspectos no definidos en un principio, de modo que las propuestas de solución sean fácilmente entendidas por el cliente. Este caso es especialmente relevante para cuando se desarrolla utilizando metodologías ágiles. [23]

5.8.2 StarUML

Es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture), que en un principio era un producto comercial y que hace cerca de un año paso de ser un proyecto comercial (anteriormente llamado plastic) a uno de licencia abierta GNU/GPL.

El software heredó todas las características de la versión comercial y poco a poco ha ido mejorando sus características, entre las cuales se encuentran:

- ❖ Soporte completo al diseño UML mediante el uso de:
 - Diagrama de casos de uso
 - Diagrama de clase
 - Diagrama de secuencia
 - Diagrama de colaboración.
 - Diagrama de estados
 - Diagrama de actividad. – Diagrama de componentes – Diagrama de despliegue.
 - Diagrama de composición estructural (UML 2.0)

- ❖ Definir elementos propios para los diagramas, que no necesariamente pertenezcan al estándar de UML,
- ❖ La capacidad de generar código a partir de los diagramas y viceversa, actualmente funcionando para los lenguajes c++, c# y java.
- ❖ Generar documentación en formatos Word, Excel y PowerPoint sobre los diagramas.
- ❖ Patrones GoF (Gang of Four) , EJB (Enterprise JavaBeans) y personalizados.
- ❖ Plantillas de proyectos.

- ❖ Posibilidad de crear plugins para el programa. . [24]

5.8.3 HTML

HTML es un lenguaje de programación que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponde a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

Se trata de un formato abierto que surgió a partir de las etiquetas SGML (Standard Generalized Markup Language). Concepto traducido generalmente como “Estándar de Lenguaje de Marcado Generalizado” y que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista. Este lenguaje es el que se utiliza para especificar los nombres de las etiquetas que se utilizarán al ordenar, no existen reglas para dicha organización, por eso se dice que es un sistema de formato abierto.

EL HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como fotografías, animaciones, etc).

Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos.

Para la escritura de este lenguaje, se crean etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >. Entre sus componentes, los elementos dan forma a la estructura esencial del lenguaje, ya que tienen dos propiedades (el contenido en sí mismo y sus atributos).

Por otra parte, cabe destacar que el HTML permite ciertos códigos que se conocen como scripts, los cuales brindan instrucciones específicas a los navegadores que se encargan de procesar el lenguaje. Entre los scripts que pueden agregarse, los más conocidos y utilizados son JavaScript y PHP.

El marcado estructural es el que estipula la finalidad del texto, aunque no define cómo se verá el elemento. El marcado presentacional, por su parte, es el que se encarga de señalar cómo se verá el texto más allá de su función. [25]

5.8.4 CSS

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C(World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "<style>".

El CSS sirve para definir la estética de un sitio web en un documento externo y eso mismo permite que modificando ese documento (la hoja CSS) podamos cambiar la estética entera de un sitio web, el mismo sitio web puede variar totalmente de estética cambiando solo la CSS, sin tocar para nada los documentos HTML o jsp o asp que lo componen.

CSS es un lenguaje utilizado para dar estética a un documento HTML (colores, tamaños de las fuentes, tamaños de elemento, con css podemos establecer diferentes reglas que indicarán como debe presentarse un documento. Podemos indicar propiedades como el color, el tamaño de la letra, el tipo de letra, si es negrita, si es itálica, también se puede dar forma a otras cosas que no sean letras, como colores de fondo de una página, tamaños de un elemento (por ejemplo el alto y el ancho de una tabla. [26]

5.8.4.1 Características principales de CSS3

- Atributo gradiente de colores en borde con CSS y Firefox: Posibilidad de definir el un gradiente de color en el borde de los elementos con CSS, en un atributo no estándar de Firefox.
- Bordos redondeados en CSS 3: Las características de CSS 3 incluyen bordes redondeados, a través del atributo border-radius, que define la curvatura que debe tener el borde del elemento.
- Múltiples imágenes de fondo con CSS: Cómo conseguir que un elemento de la página tenga varias imágenes de fondo a la vez, con CSS básico y con características de CSS 3.
- Colores RGBA en CSS 3: Veremos qué son los colores RGBA y su notación, que se incluyen en la especificación de Hojas de Estilo en Cascada CSS 3.
- Textos multi-columna con CSS 3: CSS 3 incorpora nuevos atributos para que el navegador se encargue de producir texto multicolumna, es decir, que maquete directamente el texto en varias columnas sin tener que hacer nosotros nada.
- Bordos con imágenes en CSS 3: El atributo border-image y varios otros de CSS 3 harán posible la utilización de imágenes como bordes de los elementos de la página, sin código HTML especial, simplemente con hojas de estilo.
- Sombras en CSS 3 con box-shadow: Crear sombras en CSS3 con el atributo box-shadow. Por fin podremos aplicar sombras a los elementos de la página, sin usar imágenes, Javascript ni nada extra, simplemente con un atributo de CSS 3.
- Resplandor exterior con CSS3: Cómo realizar un elemento que tenga un resplandor exterior con CSS3 y la propiedad box-shadow.
- Propiedad background-origin de CSS 3: La propiedad de CSS 3 background-origin permite decidir la posición de la imagen de fondo con respecto al borde, padding o el contenido del elemento.
- Sombras en el texto con text-shadow de CSS: Cómo aplicar sombras y otros efectos en los textos con CSS y el atributo text-shadow. [26]

5.8.5 Bootstrap

Twitter Bootstrap es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.

Es el proyecto más popular en GitHub y es usado por la NASA y laMSNBC entre otras organizaciones. [27]

5.8.6 Materialize CSS

Este framework ofrece una serie de recursos en CSS, Fuentes y JS así como su versión con Sass para personalizarlo y compilarlo nosotros mismos.

- Permite (al igual que otros frameworks como Bootstrap) crear interfaces que se adapten a distintas resoluciones y navegadores.
- Crear menú lateral abierto o desplegable en función de la resolución.
- Animaciones con filosofía Material Design.
- Crear diseños con Material Design como botones, tarjetas, colecciones, iconos, barras de navegación, etiquetas, ‘toast’, modales, etc.
- Integra Parallax, Pushpin y scrollspy para hacer un diseño más atractivo [28]

5.9 Desarrollo y documentación del sistema

En el mundo globalizado de hoy es necesario que las empresas, instituciones públicas, entre otros, cuenten con un sistema documental de sus procedimientos, servicios y funciones; ya que la documentación es el soporte del sistema de gestión de la calidad y planificación, pues en ella se plasman no sólo las formas de operar de la organización sino toda la información que permite el desarrollo de todos los procesos, la toma de decisiones y permitir las auditorías. Existen diversas técnicas, elementos y metodologías con los cuales se puede elaborar la documentación que nos permita atacar las necesidades que tenga cualquier tipo de organización o cliente; de manera tal que la documentación a implementar cumpla con los requisitos, permita el desarrollo y la permanencia en el mercado.

Con la documentación se puede mantener y actualizar los sistemas de información de una empresa frecuentemente, con el fin de brindar efectividad en el diseño de las estrategias; un buen sistema de documentación debe desarrollarse utilizando un método único, estandarizado que proporcione una

comunicación clara y rápida, ya que pueden reducir costos en el almacenamiento, adiestramiento de personal, adquisición de tecnologías y otros. A continuación señalaremos los aspectos que tienen que ver con las técnicas elementos de la documentación de sistema, tales como: funciones de la documentación, técnicas para hallar datos, entre otros.

5.9.1 La Documentación

En términos generales, la documentación es el acto de reunir documentos sobre un tema dado y el tratamiento de éstos en vistas a su difusión.

5.9.1.1 Campo de la documentación de sistemas

Son aquellos que esporádica o periódicamente tienen necesidad de documentarse con motivo de estudio, trabajo, investigación, docencia, formación permanente, decisión, etc.

5.9.1.2 Características de la documentación de sistemas

- **Pertinencia:** los documentos que se proporcionan deben responder a las necesidades específicas de los usuarios.
- **Exhaustividad:** deben proporcionarse al usuario todos los documentos que respondan a sus necesidades.
- **Rapidez:** los documentos pertinentes deben transmitirse al usuario interesado inmediatamente después de su publicación.
- **Economía:** el coste de la Documentación debe ser mínimo. [20]

5.9.2 Visual Studio

Es un conjunto completo de herramientas de desarrollo para la generación de aplicaciones web ASP.NET, Servicios Web XML, aplicaciones de escritorio y aplicaciones móviles. Visual Basic, Visual C# y Visual C++ utilizan todos el mismo entorno de desarrollo integrado (IDE), que habilita el uso compartido de herramientas y facilita la creación de soluciones en varios lenguajes. Asimismo, dichos lenguajes utilizan las funciones de .NET Framework, las cuales ofrecen acceso a tecnologías clave para simplificar el desarrollo de aplicaciones web ASP y Servicios Web XML. [29]

5.9.3 Arquitectura Cliente servidor

Este sistema utilizará una arquitectura cliente – servidor, la cual se detalla de manera concreta en la sección de definiciones del Marco Teórico.

5.9.4 Programación en capas

Cuando se construye software como producto empresarial o comercial, se llevan a cabo varias técnicas de manera que el desarrollo se haga en forma ordenada y así poder asegurar un avance continuo del proyecto, un producto final de calidad, y además realizar posteriores mejoras para que sea una tarea más fácil.

Existen muchas prácticas de programación, dependiendo del tipo de software que se va a desarrollar y de la disciplina o disciplinas de programación que se utilicen en el desarrollo del producto. Una de las más utilizadas se llama programación por capas, que consiste en dividir el código fuente según su funcionalidad principal.

Para lograr sacarle el mayor provecho a la programación por capas se necesita seguir una serie de pasos complejos los cuales primeramente deben ser definidos para cada proyecto en específico, luego deben ser revisados para asegurarse de que el modelo adoptado cumpla con las normas necesarias para que la aplicación sea del agrado del usuario, y por último debe ser implementado por el grupo de desarrollo encargado para tal fin, los cuales siguiendo el modelo propuesto obtienen una herramienta útil para facilitar la labor de programación dividiendo la aplicación en módulos y capas fáciles de pulir.

5.9.4.1 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.

5.9.5 Prueba y Mantenimiento del Sistema

Tres medios instrumentales de minimizar los riesgos de la tecnología son la verificación, prueba y mantenimiento de los sistemas. Cada componente de un sistema de cómputo -equipo, comunicaciones y programas- debe ser verificado y probado rigurosamente antes de utilizarlo para un evento electoral. Después de una prueba exitosa, los sistemas requieren mantenimiento regular para asegurarse que funcionarán de manera efectiva cuando se requieran.

Es probable que el nivel de importancia tecnológica determine el grado de rigor aplicado a la verificación, prueba y mantenimiento de la tecnología. Para un sistema que va a ser utilizado en una función electoral clave, como una votación electrónica, el nivel de rigor requerido será mayor.

5.9.5.1 Verificación

Para un sistema muy importante, como uno de votación electrónica, es conveniente que una autoridad independiente aplique las pruebas de verificación. Para sistemas de menor importancia, la verificación puede realizarse internamente.

Las pruebas de verificación (también conocidas como pruebas de calidad) pueden incluir:

- Probar los equipos bajo condiciones que simulen las de operación real.

- Probar los programas para asegurar que se siguen los estándares apropiados y que desempeñan las funciones esperadas.
- Asegurar que la documentación sea la adecuada y esté completa.
- Asegurar que los sistemas de comunicación se ciñan a los estándares establecidos y funcionen de manera efectiva.
- Verificar que los sistemas sean capaces de operar bajo condiciones normales, pero también bajo potenciales condiciones inesperadas.
- Asegurar que se cuente con las debidas medidas de seguridad y que estas se ciñan a las normas establecidas.

5.9.5.2 Prueba

La prueba de los sistemas es usualmente más detallada y rigurosa que la verificación. Se requiere para asegurar que cada componente del sistema esté en operación como debe y que el sistema en su conjunto se desempeñe exactamente de acuerdo con los requerimientos locales específicos.

Para un sistema importante, como el de votación electrónica, un programa estructurado de prueba constituye un medio para asegurar que todos sus componentes sean evaluados. Las medidas de prueba que se pueden seguir incluyen:

- Desarrollar un conjunto de criterios para la prueba.
- Examinar todos los códigos no estandarizados para garantizar su lógica y que se hayan seguido los estándares debidos de diseño y construcción.
- Aplicar pruebas "no operativas" para asegurar que el equipo puede tolerar los niveles de manejo físico esperado.
- Aplicar pruebas funcionales para determinar si se han satisfecho los criterios de prueba.
- Aplicar evaluaciones de calidad para determinar si se han satisfecho los criterios de prueba.
- Conducir pruebas en condiciones de "laboratorio" y en una variedad de condiciones "reales".
- Conducir pruebas durante un periodo prolongado, para cerciorarse que los sistemas pueden funcionar de manera consistente.

- Conducir "pruebas de carga", simulando tanto como sea posible una variedad de condiciones reales utilizando o excediendo los volúmenes de información que se pueden esperar en una situación concreta.
- Verificar que lo que entra es lo que sale, introduciendo información conocida y verificando que el resultado sea consecuente con ella.

5.9.5.3 Mantenimiento

Después de que los sistemas han sido verificados, probados e implantados, se les debe seguir dando mantenimiento para asegurar que continúen operando en el nivel mostrado durante la etapa de prueba. Las rutinas de mantenimiento variarán de acuerdo con el tipo y complejidad de la tecnología. Los fabricantes o proveedores suelen indicar en muchos productos el programa o calendario de mantenimiento requerido. El mantenimiento también puede ser realizado por el fabricante o el proveedor como parte del acuerdo de compra.

El monitoreo permanente de los sistemas necesita ser sistematizado para asegurar que las necesidades de mantenimiento sean identificadas y satisfechas cuando resulte necesario. Cuando los sistemas son de uso prolongado, se puede establecer un mecanismo para recibir retroalimentación de los usuarios como otra forma de determinar las necesidades de mantenimiento y modificación.

Cuando se realicen modificaciones al equipo, programa o comunicaciones como resultado de programas de mantenimiento o actualización, puede ser necesario promover rondas adicionales de verificación y prueba del sistema para asegurarse que sigue cumpliendo las normas exigidas. [30]

VI. Marco Metodológico

A continuación se detalla la metodología empleada en el desarrollo del sistema de información web de gestión Académica de Academia Bilingüe Amerrisque.

6.1 Sistema de Variables

El diseño de la investigación fue de tipo no experimental: correlacional- transversal ya que no se manipuló ni se sometió a prueba las variables de estudio.

“Este diseño describe relaciones entre dos o más variables en un momento determinado. Los diseños transaccional descriptivos tienen como objetivo indagar la incidencia y valores en que se manifiesta uno o más variables”. [31]

6.2 Hipótesis

Al implementar el “Sistema para la gestión Académica en Academia Bilingüe Amerrisque”, se facilitan los procesos de matrícula, disminuyendo: El tiempo que se toma en relación a la forma manual tradicional, tiempo en consulta de notas y los costos en materiales de oficina, papelería y recursos humanos. De esta forma aumenta el nivel de conformidad de los clientes y se agilizan las consultas de información.

6.3 Operacionalización de las variables.

| Objetivo | Variable | Definición. |
|---|--|--|
| 1. <i>Identificar los requerimientos funcionales y no funcionales que necesita la aplicación web para la gestión académica, utilizando estándar de especificación de requisitos de software IEEE 830.</i> | 1. Requerimientos Funcionales y no funcionales. | Se pretende identificar cada uno de los procesos que el cliente y los usuarios necesitan para el eficaz funcionamiento del sistema, así mismo dejar claras las limitaciones del software, lo que se puede realizar, las restricciones, los roles de los usuarios, la tecnología que se va a utilizar, etc. |
| 2. <i>Diseñar las interfaces y procesos teniendo en cuenta los requerimientos identificados, aplicando herramientas de diseño.</i> | 2. Diseño en base a los requerimientos que han sido solicitados. | Diseño lógico de los requisitos recolectados. |
| 3. <i>Implementar la codificación del diseño obtenido, a través de lenguajes de programación, orientado a arquitectura distribuida.</i> | 3. Codificación en base a Diseño obtenido. | Modelado y escritura de códigos que construyan, diseñen y hagan funcionales cada uno de los requerimientos. |
| 4. <i>Evaluar el desempeño del sistema web, de acuerdo a los requerimientos establecidos.</i> | 4. implementación del sistema. | Instalación del Software en el lugar donde será utilizado y los equipos tecnológicos que se necesitan. |

Tabla 1 - Operacionalización de las variables

| Variable | Dimensiones | Indicadores |
|--|---|--|
| <i>Requerimientos funcionales y no funcionales.</i> | <ul style="list-style-type: none"> • Organizacional. • Procesos. | <ul style="list-style-type: none"> • Roles. • Datos entrada. • Datos de salida. • Asociaciones de procesos. • Procesos. • Restricciones. |
| <i>Diseño en base a los requerimientos que han sido solicitados.</i> | <ul style="list-style-type: none"> • Tecnología. • Procesos. | <ul style="list-style-type: none"> • Casos de uso. • Diagrama E/R. • Bocetos. |
| <i>Codificación en base a Diseño obtenido.</i> | <ul style="list-style-type: none"> • Documentación • Codificación | <ul style="list-style-type: none"> • Plataforma de • Programación • Navegadores. • Gestor de Base de datos. |
| <i>Implementación del sistema.</i> | <ul style="list-style-type: none"> • Procesos • Normas | <ul style="list-style-type: none"> • Manual de usuario. • Manual Técnico. • Estándar ISO • Usuario final |

Tabla 2 – Identificación de requerimiento

| Variables | Indicadores | Instrumentos. |
|--|---|---|
| <i>Requerimientos Funcionales y no funcionales.</i> | <ul style="list-style-type: none"> • Roles • Datos entrada • Datos de salida • Asociaciones de procesos • Restricciones | <ul style="list-style-type: none"> • SRS • Diagrama ER. • Bocetos. • Mozilla Firefox. • Microsoft Edge. • Google Chrome. • Cristal Reports. • Encuesta • Observación • Entrevista |
| <i>Diseño en base a los requerimientos que han sido solicitados.</i> | <ul style="list-style-type: none"> • Casos de uso • Diagrama E/R • Bocetos | <ul style="list-style-type: none"> • Diagramas UML • Star UML • Embarcadero • MockupBuilder |
| <i>Codificación en base a Diseño obtenido.</i> | <ul style="list-style-type: none"> • Plataforma de Programación • Navegadores • Sistema Gestor de Base de datos • Manual de usuario • Manual Técnico | <ul style="list-style-type: none"> • Microsoft Office Word 2013 • IEEE 1063. • Visual Studio.Net • SQL Server 2012 • HTML5 • CSS3 |
| <i>Implementación del sistema.</i> | <ul style="list-style-type: none"> • Manual de Pruebas | <ul style="list-style-type: none"> • Plantilla de Pruebas de requerimientos funcionales del software. |

Tabla 3 - Diseño de las interfaces

6.4 Tipo de investigación.

La presente documentación del proyecto software, aplicación web para la Gestión Académica del Colegio A.B.A., está basada en el tipo de investigación aplicada, ya que en esta investigación utilizamos los conocimientos obtenidos en diferentes investigaciones y realizamos una práctica sobre un proyecto que tenemos en desarrollo, siguiendo un diseño transversal el cual determina el tiempo en que se realiza esta investigación. [32].

También es una investigación de campo según Razo:

“En la ejecución de los trabajos de este tipo, tanto el levantamiento de información como el análisis, comprobaciones, aplicaciones prácticas, conocimientos y métodos utilizados para obtener conclusiones, se realizan en el medio en el que se desenvuelve el fenómeno o hecho en estudio.

La presentación de resultado se complementa con un breve análisis documental. En estas investigaciones, el trabajo se efectúa directamente en el campo (80 a 90 por ciento) y solo se utiliza un estudio de carácter documental para avalar o complementar los resultados (20 a 10 por ciento).” [33]

6.5 Métodos y técnicas de recolección.

Para obtener e identificar el tipo de información que se necesitó para realizar este proyecto de software se han utilizado fuentes y técnicas que se emplearon para la recolección de la información.

6.5.1 La Observación: Es la acción de observar, mirar detenidamente, en el sentido del investigador es la experiencia, es el proceso de mirar detenidamente, o sea, en sentido amplio, el experimento, el proceso de someter conductas de algunas cosas o condiciones manipuladas de acuerdo a ciertos principios para llevar a cabo la observación. Observación significa también el conjunto de cosas observadas, el conjunto de datos y conjunto de fenómenos. En este sentido, que pudiéramos llamar objetivo, observación equivale a dato, a fenómeno, a hechos. [34]

6.5.2 La entrevista: Es la técnica de recolección de información empleada para identificar ciertos requerimientos que se recopilan únicamente a través de la narración del cliente y que para su entendimiento el cliente debe de aclarar y describir claramente lo que él desea que haga el sistema.

Como modelo para el desarrollo de un Software, este proyecto está siguiendo las técnicas que se encuentran en el modelo de desarrollo del sistema que sugiere Ian Sommerville, encaminando este proyecto cumpliendo con los cuatro pasos que propone este paradigma: *Análisis, diseño, desarrollo y validación*. [1].

Para el análisis de los requerimientos funcionales y no funcionales, que son todas las necesidades, roles y característicos que requiere el cliente y los usuarios para el correcto funcionamiento del sistema, se utilizó un documento de especificación de requerimientos (SRS) el cual está basado en el Estándar IEEE 830 – 1998.

6.6 Técnicas de procesamiento y análisis de datos.

Una vez recolectados los datos proporcionados por los instrumentos, se procederá al análisis estadístico respectivo realizado con la herramienta SPSS programa estadístico muy usado en las ciencias exactas, sociales y aplicadas, además de las empresas de investigación de mercado. Los datos serán evacuados y presentados en tablas y gráficos de distribución de frecuencias.

6.7 Modelo de desarrollo del Software

Un modelo del proceso del software es una representación abstracta de un proceso del software. Cada modelo de proceso representa un proceso desde una perspectiva particular, y así proporciona solo información parcial sobre ese proceso.

Estos modelos generales no son descripciones definitivas de los procesos del software. Más bien, son abstracciones de los procesos que se pueden utilizar para explicar diferentes enfoques para el desarrollo de software. Puede pensarse en ellos como marcos de trabajo del proceso que pueden ser extendidos y adaptados para crear procesos más específicos de ingeniería de software.

Los modelos de proceso que se incluyen en el libro de Ingeniería de Software 7ma edición de Ian Sommerville son:

- 1. El modelo en cascada. Considera las actividades fundamentales del proceso de especificación, desarrollo, validación y evolución, y los representa como fases separadas del proceso, tales como la especificación de requerimientos, el diseño del software, la implementación, las pruebas, etcétera.*

2. *Desarrollo evolutivo. Este enfoque entrelaza las actividades de especificación, desarrollo y validación. Un sistema inicial se desarrolla rápidamente a partir de especificaciones abstractas. Este se refina basándose en las peticiones del cliente para producir un sistema que satisfaga sus necesidades.*
3. *Ingeniería del software basado en componentes. Este enfoque se basa en la existencia de un número significativo de componentes reutilizables. El proceso de desarrollo del sistema se enfoca en integrar estos componentes en el sistema más que en desarrollarlos desde cero. [35]*

En este punto tomando como referencia los modelos propuestos en este libro, se ha decidido tomar como modelo a seguir en este proyecto de software el modelo de *Desarrollo evolutivo* ya que se basa en elaborar un proyecto inicial partiendo de las necesidades del usuario y exponiendo el proyecto a los puntos de vista del cliente con el fin de satisfacerlo cumpliendo con los requisitos especificados.

VII. Marco administrativo

A continuación se presenta en esta sección del documento el cronograma de trabajo con el cual se detallan las fechas en las que se cumplirá cada etapa del desarrollo del software. También se define el presupuesto de trabajo que conlleva el desarrollo del software tomando en cuenta todos los gastos identificados.

Cabe mencionar que el presupuesto definido para este proyecto ha sido elaborado tomando como referencia una parte del modelo de estimación de costes COCOMO en el cual se detallan cada uno de los gastos y esto nos permite realizar una sumatoria del costo total del proyecto de forma aproximada.

7.2 Presupuesto del trabajo

Entre los distintos métodos de estimación de costes de desarrollo de software, el modelo COCOMO (COntstructive COst MOdel) desarrollado por Barry M. Boehm, se engloba en el grupo de los modelos algorítmicos que tratan de establecer una relación matemática la cual permite estimar el esfuerzo y tiempo requerido para desarrollar un producto.

Por un lado, COCOMO define tres modos de desarrollo o tipos de proyectos:

- **Orgánico:** proyectos relativamente sencillos, menores de 50 KDLC líneas de código, en los cuales se tiene experiencia de proyectos similares y se encuentran en entornos estables.
- **Semi-acoplado:** proyectos intermedios en complejidad y tamaño (menores de 300 KDLC), donde la experiencia en este tipo de proyectos es variable, y las restricciones intermedias.
- **Empotrado:** proyectos bastantes complejos, en los que apenas se tiene experiencia y se engloban en un entorno de gran innovación técnica. Además, se trabaja con unos requisitos muy restrictivos y de gran volatilidad.

Y por otro lado existen diferentes modelos que define COCOMO:

- **Modelo básico:** Se basa exclusivamente en el tamaño expresado en LDC.
- **Modelo intermedio:** Además del tamaño del programa incluye un conjunto de medidas subjetivas llamadas conductores de costes.
- **Modelo avanzado:** Incluye todo lo del modelo intermedio además del impacto de cada conductor de coste en las distintas fases de desarrollo.

Para nuestro caso el modelo Básico será el que usaremos, dado que realiza las estimaciones con bastante facilidad y precisión.

El Modelo Básico de COCOMO'81 estima el esfuerzo y el tiempo empleado en el desarrollo de un proyecto de software usando dos variables predictivas denominadas factores de costo (cost drivers): el tamaño del software y el modo de desarrollo. [36]

Las ecuaciones básicas son:

Esfuerzo:

| | |
|-------------------------|------------------------------|
| $E = a \times (KLDC)^B$ | <i>Ecuación 1 - Esfuerzo</i> |
|-------------------------|------------------------------|

Cronograma:

| | |
|----------------------|--------------------------------|
| $D = C \times (E)^D$ | <i>Ecuación 2 - Cronograma</i> |
|----------------------|--------------------------------|

Donde:

- *E* es el esfuerzo estimado. Representa los meses- persona necesarios para ejecutar el proyecto
- *KLDC* es el tamaño del software a desarrollar en miles de líneas de código
- *A* y *B* son coeficientes que varían según el Modo de Desarrollo (*Orgánico, Semiacoplado, Empotrado*)

Donde:

- *D* representa los meses de trabajo que se necesitan para ejecutar el proyecto
- *C* y *D* son coeficientes que varían según el Modo de Desarrollo (*Orgánico, Semiacoplado, Empotrado*)

En la tabla se muestra la variación de la fórmula de estimación de esfuerzo y cronograma para los tres Modos de Desarrollo:

Tabla 1- Variaciones de las formulas

| Modo de Desarrollo | Esfuerzo | Cronograma |
|---------------------|--------------------------------|-----------------------------|
| Orgánico | $E = 2.4 \times (KLDC)^{1.05}$ | $D = 2.5 \times (E)^{0.38}$ |
| Semiacoplado | $E = 3.0 \times (KLDC)^{1.12}$ | $D = 2.5 \times (E)^{0.35}$ |
| Empotrado | $E = 3.6 \times (KLDC)^{1.20}$ | $D = 2.5 \times (E)^{0.32}$ |

Tomado del estándar de estimación de costos COCOMO. [37]

Tanto a, b, c y d son constantes y presentan un valor diferente para cada modo de desarrollo, siendo nuevamente el orgánico en aplicado a continuación, conocemos que:

- a = 2,4
- b = 1,05
- c = 2,5
- d = 0,38

Obtenemos el Total de líneas de código (*KLDC*) escritas en los diferentes lenguajes de programación.

Tabla 4 - Total de líneas de códigos

| Lenguaje | Líneas de Código |
|--------------|------------------|
| C# | 3259 |
| HTML | 2593 |
| Sql | 1342 |
| JavaScript | 525 |
| CSS | 3522 |
| ASP.NET | 854 |
| Total | 12095 |

Así pues, tras saber que son 12095 líneas de código, sustituyendo la fórmula para encontrar el esfuerzo y el tiempo:

$$E = 2.4 \times (KLDC)^{1.05}$$

$$E = 2.4 * (12095)^{1.05}$$

Se obtiene el resultado del esfuerzo:

$$E = 48524$$

$$D = 2.5 \times (E)^{0.38}$$

$$D = 2.5 * (48524)^{0.38}$$

Se obtiene el resultado del tiempo de trabajo en relación con el cronograma:

$$D = 85.48$$

Justificación de los valores:

Atributos de software

- **Fiabilidad requerida del software:** Si se produce un fallo por el pago de un pedido, o fallo en alguna reserva, etc.... puede ocasionar grandes pérdidas a la empresa (Valoración Alta).
- **Tamaño de la base de datos:** La base de datos de nuestro producto será de tipo estándar (Valoración Nominal).
- **Complejidad del producto:** La aplicación no va a realizar cálculos complejos (Valoración Baja).

Atributos de hardware

- **Restricciones del tiempo de ejecución:** En los requerimientos se exige alto rendimiento (Valoración Alta).
- **Restricciones del almacenamiento principal:** No hay restricciones al respecto (Valoración Nominal).
- **Volatilidad de la máquina virtual:** Se usarán sistemas de la “Familia Windows” (Valoración Nominal).
- **Tiempo de respuesta del ordenador:** Deberá ser interactivo con el usuario (Valoración Alta).

Atributos del personal

- **Capacidad del analista:** Capacidad alta relativamente, debido a la experiencia en análisis en proyecto similar (Valoración Alta)
- **Experiencia en la aplicación:** Se tiene cierta experiencia en aplicaciones de esta envergadura (Valoración muy alta).
- **Capacidad de los programadores:** Teóricamente deberá tenerse una capacidad muy alta por la experiencia en anteriores proyectos similares (Valoración muy alta).
- **Experiencia en S.O. utilizado:** Con Windows 2000 Professional la experiencia es a nivel usuario (Valoración Nominal).
- **Experiencia en el lenguaje de programación:** Es relativamente alta, dado que se controlan las nociones básicas y las propias del proyecto (Valoración Alta).

Atributos del proyecto

- **Prácticas de programación modernas:** Se usarán prácticas de programación mayormente convencional (Valoración Nominal).
- **Utilización de herramientas software:** Se usarán herramientas estándar que no exigirán apenas formación, de las cuales se tiene cierta experiencia (Valoración Alta).
- **Limitaciones de planificación del proyecto:** Existen pocos límites de planificación. (Valoración Baja).

Productividad:

$$PR = KLDC / E$$

Ecuación 3 - Tiempo del cronograma

$$PR = 12095 / 85.48 = 141.50 \text{ LDC/ Por persona}$$

Según estas cifras será necesario un equipo de 2.25 personas trabajando alrededor de 3.40 meses, Dado que las personas no se pueden dividir de esa manera y el tiempo límite es de 4 meses, se hace un cálculo de costo en base a esto, sabiendo que según el COCOMO el tamaño del equipo desarrollador y la cantidad de tiempo son coherentes con lo obtenido.

Costo:

Cada persona trabaja en promedio 4 horas al día de lunes a viernes, teniendo en el mes un total de 70 horas al mes por persona.

La fórmula sería la siguiente:

$$\text{Costo} = \text{Horas trabajadas} * \text{Personas} * \text{meses}$$

Sustituyendo:

$$\text{Costo} = 70 \times (2) \times (4) = 560 \text{ horas totales}$$

Dando como resultados:

$$\text{Horas trabajadas por persona} = 70 \text{ H} * 4 \text{ meses} = 280 \text{ H.}$$

$$\text{Total LDC por persona} = LDC / 2 \text{ personas} \quad \text{esto es igual a} \quad 12095 / 2$$

$$\text{Total LDC por persona} = 6047.5 \text{ LDC}$$

$$\text{LDC por hora por persona} = 6047.5 \text{ LDC} / 280 \text{ H. por persona}$$

$$= 21.6 \text{ LDC por hora por persona}$$

Costo de desarrollo del proyecto

| Gasto | Precio | Cantidad | Total |
|-----------------------------------|--------|------------|-----------------|
| Pago a desarrolladores | \$15 | 560 Horas | \$ 8400 |
| Viáticos | \$50 | 2 Personas | \$ 100 |
| Electricidad | | | \$100 |
| Costo total de desarrollo: | | | \$ 8,600 |

Tabla 5 - Costo de desarrollo

Costo de licencias de Software

| Software | Costo |
|----------------------------|------------------|
| Visual Studio 2012 | \$ 40,00 |
| Microsoft SQL Server 2012 | \$ 50,00 |
| Embarcadero E/R Studio 8.0 | \$ 10,00 |
| StarUml | \$ 10,00 |
| Balsamiq Muckps 3 | \$ 10,00 |
| Project 2016 | \$ 20,00 |
| TOTAL | \$ 140,00 |

Tabla 6 - Costo de licencia

Costo del documento

| Gasto | Precio Unitario | Cantidad | Precio Total |
|---|------------------------|-----------------|---------------------|
| Horas trabajadas en doc. | C\$ 50 | 50 horas | C\$ 2,500 |
| Electricidad | C\$ 5 | 50 horas | C\$ 250 |
| Papelería | | | C\$ 100 |
| Impresión | | | C\$ 300 |
| Fotocopias | C\$ 100 | 2 | C\$ 200 |
| Varios (Transporte, comunicación, internet) | | | C\$ 500 |
| Precio Total de documentación | | | C\$ 3,750 |
| Cambio de dólar (C\$ 30.00) | | | \$ 125 |

Tabla 7 - Costo del documento

Costo total del proyecto

| Concepto | Costo |
|--------------------------------|-----------------|
| Costo de desarrollo | \$ 8.600 |
| Costo de documentación | \$ 125 |
| Costo de licencias de software | \$ 140 |
| Total | \$ 8,865 |

Tabla 8 - Costo total del Proyecto

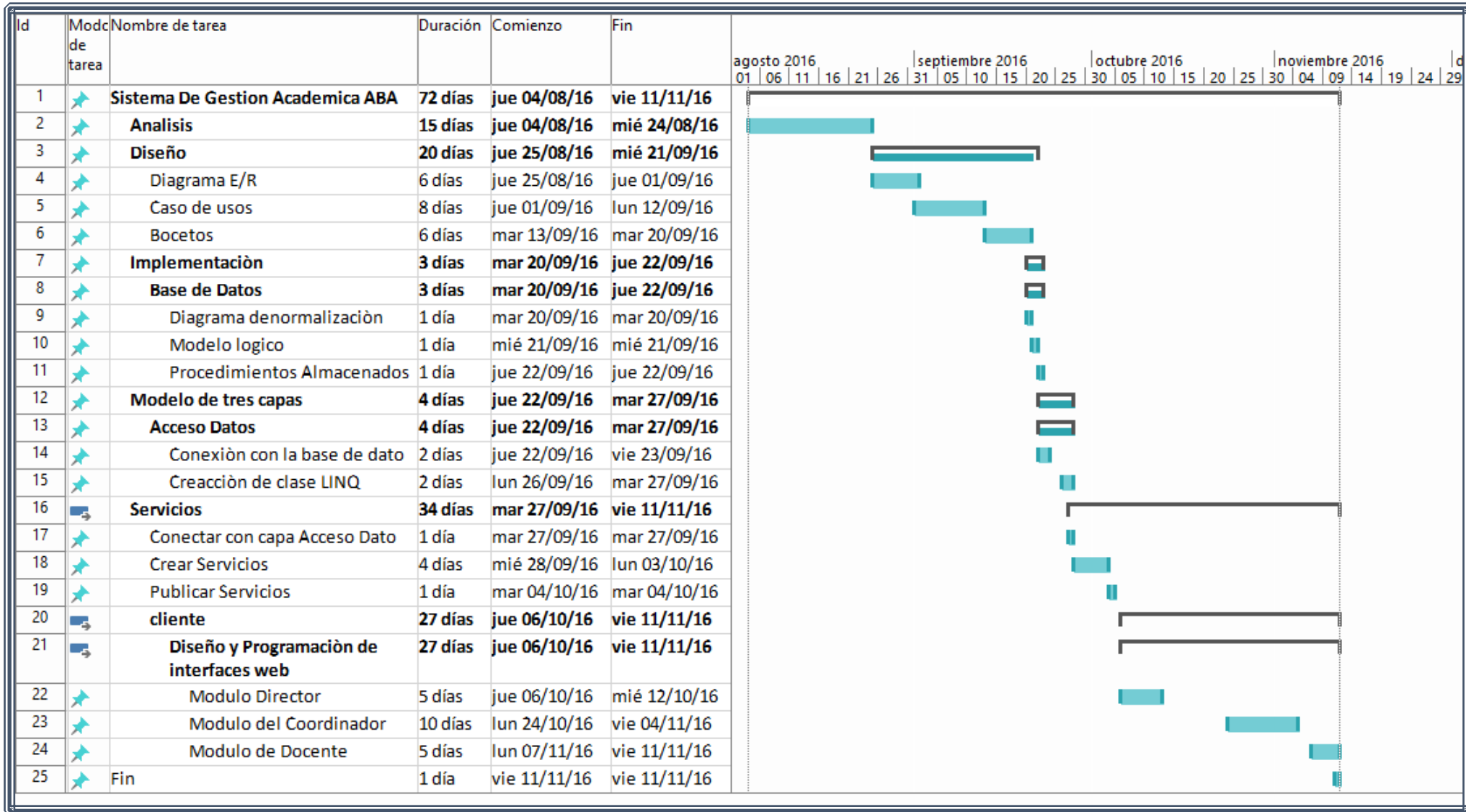
VIII. Marco de resultado

En esta sección del documento se presentaran los resultados obtenidos de las herramientas aplicadas correspondientes a los objetivos del sistema que fueron definidos tomando en cuenta el modelo de desarrollo del software que se ha establecido para este proyecto.

8.1 Cronograma del desarrollo del software

En este cronograma se definen los entregables del proyecto que genera cada una de las etapas de desarrollo del software de acuerdo al modelo utilizado. Todos estos entregables son los que se realizan a través de las herramientas de diseño y desarrollo de software que hemos explicado en el Marco teórico, cada una de estas herramientas tienen cierto nivel de dificultad para quienes la utilizan (teniendo experiencia o no), por este motivo se detalla en este cronograma el tiempo que demora cada entrega de los resultados que se obtienen al hacer uso de cada una de estas herramientas.

Ilustración 2- Cronograma del desarrollo del software



8.2 Descripción de los resultados por objetivos

Para mostrar los resultados dividiremos estos de acuerdo a los objetivos que se han planteado para este proyecto, de esta manera se brindan soluciones a la problemática planteada que pretende resolver el sistema.

8.2.1 Análisis de requerimientos del software

El primer objetivo que plantea este proyecto es, identificar los requerimientos funcionales y no funcionales que necesita la aplicación web para la gestión académica del colegio A.B.A., utilizando el estándar de especificación de requisitos del software IEEE830.

Además se aplicó una entrevista como herramienta de recolección de información. Con estas herramientas se obtuvieron los resultados que se muestran a continuación.

8.2.2 Entrevista realizada a dirección del colegio A.B.A.

La entrevista como herramienta de recolección de información ha sido aplicada personalmente de forma grabada a la directora del colegio quien no se negó a brindarnos sus opiniones y responder a todas las interrogantes e iniciativas propuestas en la entrevista. Cabe mencionar que al momento de hacer la pregunta del inciso número ocho que dicta: ¿Considera necesario un sistema de información que automatice y agilice la mayoría de estos procesos en A.B.A.? A lo que la directora ha dicho que Sí, que la Academia está abierta a la implementación de nuevas tecnologías que ayuden al desarrollo del centro.

Para ver la encuesta completa que ha sido aplicada ir a la sección de Anexos N° 1.1

8.2.3 Documento de especificación de requerimientos del software

Este documento está basado en el estándar de especificación de requisitos del software IEEE830. Tiene la finalidad de recopilar los requisitos especificados por el cliente los cuales se obtienen mediante otras técnicas de recolección de información tales como la entrevista, la observación y una técnica antiquísima para recolectar datos que es la experiencia, de la cual se obtuvieron muchos requisitos por la práctica de los procesos que se hacen manualmente en la academia y que necesitan ser actualizados porque así lo demanda el cliente, en este documento S.R.S. se plantean también los procesos que realizan cada uno de

los usuarios del sistema y la forma en los que estos se deberían de automatizar con el software.

Para ver el documento completo ir a sección de Anexos N° 7

8.3 Diseño y modelado del software

El segundo objetivo que plantea este proyecto es el diseño de las interfaces y procesos teniendo en cuenta los requerimientos identificados aplicando herramientas de diseño tales como Balsamiq Mockups, Embarcadero, Star UML, SQL Server, entre otros. En este objetivo se pretende modelar o diseñar las interfaces y los procesos con los que va a contar el sistema partiendo de los requerimientos identificados en la etapa anterior, para esto se han elaborado diagramas de casos de uso, bocetos, base de datos relacional, diccionario de datos, mapa de navegación, entre otros.

8.3.1 Diagramas de Casos de uso

En la elaboración de los diagramas de caso de uso se ha utilizado la herramienta Star UML definida en el marco teórico con la cual se han identificado y graficado los principales procesos y acciones que realiza cada usuario al utilizar este sistema. Al dividir las acciones por cada usuario se logra identificar algunos de los menús que muestra la aplicación a los usuarios que pueden acceder al sistema.

Para ver los diagramas de casos de uso ir a sección de Anexos, N° 2

8.3.2 Bocetos del Sistema

Para la realización del modelado del sistema se han generado algunos bocetos como punto de partida para la estructura de la interfaz gráfica del sistema tomando en cuenta los procesos identificados y graficados en los casos de uso, se utilizó el software Balsamiq Mockups y se obtuvieron como resultados los diseños de las interfaces para los usuarios que se muestran en los anexos, sección 6

8.3.3 Diseño de la base de datos

Como se puede observar en la imagen del diagrama de la base de datos del sistema, esta cuenta con diecisiete tablas con sus respectivos atributos y relaciones que permite la dinámica en las transacciones que se generan al ser consultada.

Se observan algunas tablas Maestro – Detalle tales como “ingreso_socios” o “ingreso_matricula” cuyo objetivo consiste en mostrar datos de orígenes de datos diferentes en un único informe. Por ejemplo la tabla “ingreso” posee un campo que se llama “tipoIngreso” el cual contiene ciertos valores predeterminados para los tipos de ingresos fijos de colegio (Pago de mensualidad, pago de matrícula, pago de uniforme, etc.) estos “tipos de ingresos” que posee su precio predefinido y constante están almacenados precisamente en la tabla maestro – detalle nombrada “tipoIngreso”. Ver imagen del diagrama en Anexos, sección 3.

8.3.4 Diccionario de datos

El diccionario de datos muestra todas las tablas con sus atributos, tipos de datos, valores y una breve descripción por tabla de lo que se guarda en la base de datos del sistema, la finalidad de este diccionario es mostrar de forma clara los campos de las tablas y al mismo tiempo definir algunos ejemplos de cómo serán llenados estos campos y la descripción de ellos para tener una mejor comprensión de la información que será almacenada.

Para ver el diccionario de datos completo con cada una de sus tablas, ver Anexos, sección 4.

8.4 Desarrollo del software

El proceso del desarrollo del software se da con la utilización de las herramientas antes mencionadas y gracias a las posibilidades de las tecnologías aplicadas al proyecto y utilizando métodos de programación “en capas”. Con esto se ha logrado diseñar y desarrollar un sistema rápido, fiable y seguro.

Existen 2 proyectos, El “cliente” que controla todo lo relacionado con la presentación y captura de información del usuario final, y el “servidor” el cual es encargado de obtener información de la base de datos y enviarla según las solicitudes al “cliente”.

Las capas que existen en el software están distribuidas de la siguiente forma:

- Cliente: Capa presentación
- Servidor: Capa de datos (Conexión a la BD)
Capa transporte

Como primera instancia y luego de poseer la estructura final que tendrá la base de datos e incluir algunos datos de pruebas para el desarrollo, se creó la conexión a la BD almacenada en un servidor local de la pc de desarrollo, luego se he creado la clase “.oldb” para la conexión a la misma y se han instanciado todos los objetos (procedimientos almacenados, funciones, tablas) que están presentes, esta capa también cuenta con clases “.cs” de C# que funcionan como propiedades para dar formato a la información.

La capa transporte la cual contiene todos los [WebMethod] disponibles mediante el uso del Webservice creado, contiene conexión a la BD mediante el uso de la capa de datos y obtiene y envía datos según las peticiones del cliente.

Para el desarrollo de la capa presentación dentro del proyecto cliente, se utilizaron las herramientas de AXML de ASP.NET y el control mediante programación brindada por el lenguaje C#, el cual da la forma final a la información obtenida por el servidor y enviada al cliente para seguidamente ser presentada al usuario.

IX. Conclusiones.

- A través del estándar IEEE 830 para la especificación de requerimientos del software, se logró identificar los requerimientos funcionales del software aquí descrito, de acuerdo con las necesidades de los interesados del software. También se determinaron los requerimientos no funcionales acorde con las necesidades del software y los usuarios. De igual manera se utilizaron otras herramientas que fueron de mucha ayuda para la recolección de dichos requerimientos, por ejemplo: La Observación, la Entrevista y la Encuesta a padres de familia o tutores del centro, de la cuales pudimos identificar algunos posibles requerimientos de los usuarios finales del sistema.
- Mediante las herramientas de trabajo para el desarrollo de la aplicación web, dentro de las cuales resaltamos las siguientes:
 - Microsoft Visual Studio .Net 2012
 - HTML, lenguaje que se emplea para el desarrollo de páginas de internet.
 - Bootstrap: framework creado por Twitter que incluye varias herramientas de software libre para el diseño y desarrollo web.
 - SQL Server, sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial.

Se logró desarrollar una aplicación con todas las funciones requeridas por el cliente de tal manera que trabaja eficiente y eficazmente y a la vez es amigable, accesible y fácil de usar para los usuarios finales del sistema.

A través de la plantilla de pruebas de requerimientos funcionales se logró evaluar sistemáticamente si el software cumple con los requisitos solicitados. Esta plantilla genera casos de pruebas para cada requerimiento, y estos casos de pruebas generan los resultados que se obtienen cuando se hacen las consultas al sistema y la base de datos.

X. Recomendaciones

- Lograr un nivel de compromiso de la institución con el proyecto SIS-I, definiendo las áreas y procesos que serán afectados en el centro educativo (Implantación de Área TIC), destinando un porcentaje de ingresos a los fondos del proyecto para sus posteriores mantenimientos y mejoras, entre otros.
- Tener a cargo a una persona altamente capacitada y motivada a cargo del sistema.
- Se recomienda brindar datos de usuarios de forma digital, y al correo electrónico de cada Tutor.
- Verificar los grados académicos a los que los estudiantes serán matriculados.
- En caso de tener alguna duda al momento del registro o al momento de reservar su cupo de matrícula, Cancelar la gestión y acudir al manual de usuario o contactar al personal de soporte técnico.

XI. Glosario de términos.

Cortafuegos o firewall:

Programa de seguridad que protege al equipo del acceso desde internet de intrusos no deseados. Puede venir integrado con el sistema operativo. También se le denomina Firewall.

Disco duro:

Disco que se encuentra en el interior de la CPU y que almacena, tanto la información generada por el usuario, como los archivos necesarios para que los programas funcionen. Se le identifica por la letra C en el sistema operativo, y se le suele representar por las iniciales HD (hard disk). Su capacidad es variable, pero en la actualidad es del orden de GB.

DNS:

Servidor de Nombres de Dominio. Identificación del equipo servidor que nos da acceso a la red internet. Es necesario conocerlo para configurar un acceso telefónico a redes.

DVD:

Dispositivo de almacenamiento externo en forma de disco óptico que sólo permite la lectura de datos (no la escritura) en nuestro equipo. Es el idóneo para almacenar vídeo, dado que éste requiere mucho espacio y la capacidad de un DVD es de 4,7 GB. También los hay grabables y regrabables.

Enlace:

Elemento de una página web que da acceso a otro documento (o a otra parte del mismo documento) al hacer clic sobre él con el botón izquierdo del ratón. Es la base del acceso a la información en la World Wide Web. Un enlace puede estar sobre texto o sobre una imagen. Se reconocen porque el puntero del ratón se convierte en una mano al pasar sobre un enlace.

También se les llama vínculos o hipervínculos.

Firefox:

Navegador gratuito de la red. Es una de las alternativas más habituales a Internet Explorer, el navegador de Microsoft.

Gigabyte:

Múltiplo de Byte, unidad de almacenamiento de información. Se utiliza para dispositivos de gran capacidad, como discos duros y DVD. Véase Byte.

Hardware:

Cualquiera de los elementos físicos que componen un ordenador: disco duro, placa base, tarjeta gráfica, puertos...

HTML:

Lenguaje de marcas de hipertexto. Es el lenguaje en el que están escritas las páginas web. Realmente se trata de un texto en el que hay insertadas etiquetas que comienzan con el símbolo < y acaban con los símbolos /> Cuando nuestro ordenador descarga una página para visualizarla, descarga este código y el navegador interpreta el significado de las etiquetas mostrándonos en pantallas las páginas tal y como lo vemos.

Icono:

Pequeña imagen que representa un elemento de software (un programa, un archivo, una carpeta, un acceso directo...) Al hacer doble clic sobre él se accede al elemento.

INTERNET:

Es una red de ordenadores conectados entre sí que intercambian información a través de las líneas telefónicas. Ofrece multitud de servicios como la world wide web (www), la transferencia de ficheros, las charlas en tiempo real o chats, los foros, el correo electrónico, etc...

IP:

La IP o Dirección IP es un código numérico compuesto por cuatro grupos de hasta tres números separados por puntos, que identifica a cada ordenador conectado a la red internet, que utiliza el protocolo (código de comunicación) TCP/IP

Las siglas responden a Internet Protocol. La IP de un ordenador que servicio de acceso a internet tiene una IP fija, es decir, asignada de forma permanente. Es la manera de reconocerlo en la red. Para facilitar la identificación, en la dirección de internet se sustituye la IP por un nombre de dominio, La traducción de IP a nombre de dominio la realizan los servidores de nombres de dominio (DNS) Sin embargo, cuando este servidor nos da acceso a un ordenador personal a internet, nos asigna una IP temporalmente para que nos identifique mientras estamos conectados. Si nos desconectamos y nos volvemos a conectar, se nos asignará otra diferente. Es lo que se llama IP dinámica.

Ejemplo de IP: 164.14.153.25

Kilobyte:

Unidad de almacenamiento múltiplo del byte. Se abrevia como KB, y su equivalencia es:

$$1\text{KB} = 1024\text{B}$$

Si pensamos en la capacidad de almacenamiento de las unidades actuales, el KB es una unidad pequeña.

LAN:

Siglas correspondientes a Local Area Network, (Red de Área local). Es una red compuesta por ordenadores físicamente cercanos, conectados entre sí por un cable de red o por una WIFI (red inalámbrica). Si los ordenadores están lejanos unos de otros, forman una WAN.

Megabyte:

Unidad de almacenamiento múltiplo del byte. Se abrevia como MB, y a veces se le llama simplemente "mega". Su equivalencia es:

$$1\text{MB} = 1024\text{KB} = 1048576\text{ B}$$

Como ejemplo diremos que un CD-ROM puede albergar 700MB.

Memoria RAM:

Parte de la memoria de un ordenador en la que éste almacena información de modo temporal y automático para poder realizar sus operaciones. Cuando apagamos el equipo, esta

información se borra, volviéndose a cargar la memoria RAM con lo necesario al encenderlo de nuevo. Las siglas corresponden a "Random Access Memory" (que se traduce por memoria de acceso aleatorio).

Navegador:

Programa que interpreta el código (HTML y más) en el que están escritas las páginas web y nos las muestra tal y como las vemos en el monitor.

On-line:

Estado de un ordenador cuando está conectado a internet. La traducción literal es "en línea". Esta expresión se usa cuando se habla de actividades en las que se obtiene una respuesta del servidor que nos da acceso a internet.

Página web:

Es un archivo, escrito en código HTML, que se encuentra en el disco duro de un ordenador conectado a la red internet. Estos archivos se transfieren por la línea telefónica y, si se ven a través de un programa navegador (como Internet Explorer o FireFox) muestran texto, imágenes, animaciones, sonidos, vídeos... Pero lo más característico es que contienen enlaces (también llamados vínculo o hipervínculos) a otras páginas o documentos, de manera que podemos ir "saltando" por la información que nos interesa.

Password:

Literalmente "palabra de paso". También se le llama contraseña. Es una palabra, conocida sólo por el usuario, que le permite el acceso y uso a zonas privadas, bien de la web, bien de programas específicos. Cuando se escribe, en la pantalla aparecen asteriscos o puntos en lugar de los caracteres, para evitar que otras personas puedan conocerla.

Procesador:

Circuitos electrónicos incluidos en una pastilla que ejecutan las instrucciones básicas de un ordenador. Es el núcleo que le permite realizar todas las operaciones que nosotros utilizamos. También se le llama microprocesador, debido a su pequeño tamaño.

Red:

Conjunto de dos o más ordenadores conectados entre sí, que son capaces de intercambiar datos e información.

Servidor:

Se llama así a un ordenador central de un sistema de red que proporciona servicios y programas a otros ordenadores conectados. En el caso de estar trabajando con servicios de internet, se habla de "servidor de correo" si ofrece cuentas de correo electrónico, "servidor web" si almacena y ofrece acceso a páginas con hipertexto, "servidor de FTP" si proporciona acceso a archivos, etc...

Sistema operativo:

Programa fundamental que administra los demás programas en un ordenador.

Software:

Parte lógica del ordenador. Se trata de un conjunto de órdenes lógicas cuya ejecución permite al usuario realizar un trabajo con el ordenador. Son los llamados "programas". Se almacenan en la memoria y pueden ser muy variados: de tratamiento de texto, de tratamiento de imágenes, de control numérico, de reproducción multimedia... El propio sistema operativo del ordenador se considera software.

TCP/IP:

Son las siglas en inglés de Protocolo de Control de la Transmisión / Protocolo de Internet. Es el código que emplean los ordenadores que están conectados por la red Internet para comunicarse entre sí. Sin este código común, sería imposible el intercambio de datos.

URL:

Siglas de la expresión en inglés Uniform Resource Locator (Localizador Uniforme de Recursos). Es la dirección de una página web en internet. Por ejemplo:
<http://www.catedu.es/aratecno>

User ID:

Identificador de usuario. Conjunto de caracteres alfanuméricos que sirven para identificar a un usuario cuando quiere acceder a algún sistema, por ejemplo web sites, banca electrónica, emails, etc. Suelen ir acompañados de una contraseña o password.

XII. Referencias

- [1] I. Sommerville, Ingeniería de Software, Madrid: Pearson Educación S.A., 2005.
- [2] M. d. E. d. Colombia, "MINEDEDUCACION," 06 Mayo 2016. [Online]. Available: <https://goo.gl/U3TRBE>.
- [3] UTP, "Universidad Tecnológica de Panamá," 25 03 2015. [Online]. Available: <http://www.utp.ac.pa/matricula>. [Accessed 21 01 2017].
- [4] C. e. linea, "Biblioteca Digital UNA," [Online]. Available: <https://goo.gl/KEiSCQ> [Accessed 21 01 2017].
- [5] F. d. H. y. d. c. Jurídicas, "Universidad Nacional Autónoma de Nicaragua," [Online]. Available: <https://goo.gl/PJrs0E>. [Accessed 21 01 2017].
- [6] UCA, "Servicios en Línea UCA," 2010. [Online]. Available: <https://sistemasweb.uca.edu.ni/inicio.jsp>. [Accessed 21 01 2017].
- [7] UNI, "Universidad Nacional de Ingeniería," 18 02 2016. [Online]. Available: <https://goo.gl/KEiSCQ> [Accessed 21 01 2017].
- [8] U. F. Chontales, "Universidad Nacional Autónoma de Nicaragua," 2016. [Online]. Available: <http://www.chontales.unan.edu.ni/>. [Accessed 21 01 2017].
- [9] S. RedAccenir, "PortalProgramas," 2003-2017. [Online]. Available: <https://goo.gl/gKlfvX> [Accessed 21 01 2017].
- [10] J. A. Senn, Analisis y Diseño de SIstemas de Información, McGraw Hill, 1992.
- [11] A. M. e. D. d. apps, "Yeeply," 15 Septiembre 2015. [Online]. Available: <https://goo.gl/uSZjX4> [Accessed 04 Octubre 2016].
- [12] A. Mocholí, "Yeeply," 15 Septiembre 2015. [Online]. Available: <https://www.yeeply.com/blog/6-tipos-desarrollo-de-aplicaciones-web/>.
- [13] A. Lafuente, "Euskara," [Online]. Available: <http://www.sc.ehu.es/acwlaroa/SDI/Apuntes/Cap1.pdf>.
- [14] G. Coulouris, DISTRIBUTED SYSTEMS Cocepts and Design, Boston, Massachusetts: Pearson Education, 1994.
- [15] L. Alegs, "Alegs," 14 07 2016. [Online]. Available: http://www.alegsa.com.ar/Dic/cliente_servidor.php.

- [16] O. P. Records. Oregon Patent M0018, 1977-1994.
- [17] S. M. Lewandowski, Component-Based Client/Server Computing, ACM Computing Surveys, 1998.
- [18] R. H. D. a. E. J. Orfali, The Essential Distributed Objects Survival Guide, John Wiley & Sons., 1996.
- [19] EcuRed, "EcuRed Conocimiento con todos y para todos," 2 Octubre 2016. [Online]. Available: https://www.ecured.cu/Sitio_Web.
- [20] L. Simoes, "Scribd.," 16 Marzo 2009. [Online]. Available: <https://goo.gl/h8iMkc>
- [21] R. S. Pressman, Ingenieria del Software, un enfoque práctico, Mc Graw Hill, 1993.
- [22] K. E. y. K. J. E. Kendall, Analisis y diseño de Sistema, México: PEARSON EDUCATION, 2011.
- [23] J. M. Guerrero, "Glidea," google ad words, 13 02 2011. [Online]. Available: <https://goo.gl/xFtrb0> [Accessed 08 10 2016].
- [24] Jotatsu, "BLACK-BYTE," 19 08 2006. [Online]. Available: <http://blackbyte.com/review/staruml/>. [Accessed 08 10 2016].
- [25] J. P. P. y. A. Gardey, "Definicion.de," 2008. [Online]. Available: <http://definicion.de/html/>.
- [26] J. A. F. Sergio Luján Mora, "Universidad de Alicante," 2012. [Online]. Available: <https://goo.gl/PQxdPM> [Accessed 19 Dic 2016].
- [27] J. Solis, "arweb.com," 26 09 2014. [Online]. Available: <https://goo.gl/79TdPn> [Accessed 19 Dic 2016].
- [28] D. Iel, "Maquetando," 07 may 2015. [Online]. Available: <https://goo.gl/eulbsK> [Accessed 19 Diciembre 2016].
- [29] Microsoft, "Microsoft Developer Network," 2016. [Online]. Available: <https://goo.gl/x72Utl> [Accessed 08 10 2016].
- [30] C. F. Electoral, Performance and test standars for Punchcard, Markense, and directing recording voiting system, Washington D.C.: US Government Printing Office, 1990.
- [31] R. Sampieri, Metodología de la investigación, Mexico: Mc Graw Hill, 2006.
- [32] R. H. Sampieri, Metodologia de la investigación., México.: McGraw- Hill, 2010.
- [33] C. M. Razo, Como elaborar y asesorar una investigación de tesis, Naucalpan de Juárez, Edo. de México: Prentice Hall Hispanoamericana, S.A., 1998.

- [34] C. Sabino, El proceso de investigación, Caracas: Ed. Panamericana, Bogotá, y Ed. Lumen, Buenos Aires. , 1992.
- [35] I. Somerville, Ingenieria del Software Septima edicion, Madrid, España: PEARSON EDUCATION, S.A., 2005.
- [36] B. a. W. R. Boehm, Ada COCOMO and the Ada Process Mode, Institute, Pittsburgh, PA, 1989.
- [37] C. B. H. E. W. C. M. R. S. R. Boehm B.W., "Universidad del sur de california," 1995. [Online]. Available: <https://goo.gl/mTIMkr> [Accessed 24 01 2017].

XIII. ANEXOS

ANEXOS

Contenido

| | | |
|------------|---|-----------|
| 1. | INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN..... | 1 |
| 1.1 | Entrevista:..... | 1 |
| 2. | DIAGRAMAS DE CASO DE USO..... | 4 |
| 3. | BASE DE DATOS (DIAGRAMA ENTIDAD – RELACIÓN)..... | 8 |
| 4. | DICCIONARIO DE DATOS | 9 |
| 5. | MAPA DE NAVEGACIÓN..... | 15 |
| 6. | BOCETOS DEL SISTEMA. | 19 |

Índice de ilustraciones

| | |
|---|----|
| Ilustración 1 - Caso de uso tutor | 4 |
| Ilustración 2 - Caso de uso docente | 5 |
| Ilustración 3 - Caso de uso administrador | 6 |
| Ilustración 4 - Diagrama de entidad relación..... | 8 |
| Ilustración 5 – Mapa de navegación Usuario tutor..... | 15 |
| Ilustración 6 - Mapa de navegación usuario docente | 16 |
| Ilustración 7 - Mapa de navegación usuario administrador..... | 17 |
| Ilustración 8 - Bocetos de la página principal | 19 |
| Ilustración 9 - Boceto inicio de sesión..... | 20 |
| Ilustración 10 - Boceto página principal administrador | 21 |
| Ilustración 11 - Bocetos página tipo de matricula | 22 |
| Ilustración 12 -Boceto tipo de matricula | 22 |
| Ilustración 13 - Boceto Datos Tutor | 23 |
| Ilustración 14 – Bocetos Nuevo estudiante | 24 |
| Ilustración 15 – Boceto Datos de socio | 25 |
| Ilustración 16 - Pago de socios | 26 |
| Ilustración 17 - Listado de estudiante..... | 27 |
| Ilustración 18 - Bocetos Ingreso..... | 27 |
| Ilustración 19 - Bocetos Egreso..... | 28 |
| Ilustración 20 - Lista de datos de estudiante..... | 29 |
| Ilustración 21 - Boceto de Calificación | 30 |
| Ilustración 22 - Lista de grado a su cargo..... | 31 |
| Ilustración 23 Bocetos grado a cargo | 31 |
| Ilustración 24 - Bocetos Crear Calificación | 32 |

Índice de tablas

| | |
|---------------------------------------|----|
| Tabla 1 - Administrador | 9 |
| Tabla 2 - Asignación..... | 9 |
| Tabla 3 - Asignatura | 9 |
| Tabla 4 - Calificación | 9 |
| Tabla 5 - Egreso..... | 10 |
| Tabla 6 - Estudiante..... | 10 |
| Tabla 7 - Factura..... | 11 |
| Tabla 8 - Grado..... | 11 |
| Tabla 9 - Ingreso..... | 11 |
| Tabla 10 - Ingreso de Matricula | 12 |
| Tabla 11 - Ingreso socio | 12 |
| Tabla 12 - Matricula | 12 |
| Tabla 13 - Profesor | 13 |
| Tabla 14 - Socio..... | 13 |
| Tabla 15 -Ingreso..... | 13 |
| Tabla 16 - Tutor..... | 14 |
| Tabla 17 - Usuario | 14 |

1. Instrumentos de recolección de información.

Se aplicó la técnica de la Entrevista, realizada al personal directivo y administrativo de la institución. Esta contiene las siguientes interrogantes:

1.1 Entrevista:

UNIVERSIDAD NACIONAL AUTÓNOMA DE NICARAGUA

UNAN, FAREM - Chontales.



ENTREVISTA A: ACADEMIA BILINGÜE AMERRISQUE (ABA).

Entrevista realizada a personal directivo y administrativo de la Academia Bilingüe Amerrisque con el objetivo de Identificar y analizar los procesos de llenado de hojas de matrículas y recepción de información de Academia Bilingüe Amerrisque.

PREGUNTAS

1. ¿En qué fecha comienzan los procesos de Prematrículas y matrículas en la Academia?

R: Estamos establecido conforme al ministerio de educación, años anteriores el inicio de la matricula normalmente iniciaba en el mes de enero. A partir del año pasado el inicio de las prematricula se hacen en octubre, noviembre y enero, para un inicio de clase en el mes de febrero, 15 de octubre a 15 de noviembre son las fechas de pre matricula y 16 de noviembre al 16 de enero son las matricula.

2. ¿Cuántos formularios son llenados por cada estudiante de nuevo ingreso?

R: son dos formulario para los alumnos de nuevo ingreso, uno propio de la academia bilingüe Amerrisque y el segundo que lo proporciona el ministerio de educación.

3. ¿Cuántos formularios son llenados por cada estudiante de re-ingreso?

R: son dos formularios para los alumnos de re ingreso, uno propio de la academia bilingüe Amerrisque llamado formulario de continuidad de matrícula y el segundo que lo proporciona el ministerio de educación.

4. Después de ser llenados los formularios ¿Dónde se almacena esa información y que se hace con ella?

R: una vez llenado todos los formularios se llena un cuaderno de registro se le da apertura como matricula 2016, se dejan los numero de teléfono por que el libro de registros nos sirve para cualquier emergencia que vayamos a tener con los niños, además de eso se guardan se archivan en un campo donde van a estar todas las matricula de registro de los expedienté de los niños matriculado en el mismo año e igual los registros de matrícula del ministerio de educación

5. ¿Cuáles son los principales requisitos para la matricula?

R: Partida de nacimiento, Fotocopia de cedula de los padres, numero cedula del responsable que retire el niño, tarjeta de vacuna, llenado de formulario exigido, firma según como sale en su cedula, las notas del último año aprobado, epicrisis de estudiante que presente problemas de salud para su posterior atención en la academia.

6. ¿Existen políticas internas de la academia para poder tener derecho a matricula, tales como política, religión, promedio de notas, conducta, etc.?

R: no existes políticas que impida ingresar al colegio siempre y cuando cumpla con los lineamientos y reglamento de manera general como interno, hacer su pago debidamente, traer su nota de su último año aprobado.

7. ¿Existen reglas internas para la limitación de cupos de matrícula por grados?

R: dentro de la academia las aulas que tenemos están capacitada para no más de 18 estudiante por grado, la limitación de cupo de alumno matriculado no pasa de 18 por que la prioridad número uno de la academia es la atención individual, por ese mismo motivo no puede sobrepasar de un cupo de 18 estudiante.

8. Tomando en cuenta todos los procesos que se realizan para matricular a un estudiante; el tiempo que esto conlleva, la seguridad de la información, la facilidad para consultar cualquier información. ¿Considera necesario un sistema de información que automatice y agilice la mayoría de estos procesos en ABA?

R: si se ha considerado la opción de establecer un sistema para mejor facilidad de los procesos en la institución, pero por el costo de los sistemas no se ha implementado por que es una institución que está en crecimiento y no hay presupuesto para implementarlo.

9. Tomando en cuenta que ningún otro colegio en Juigalpa ha implementado este tipo de tecnología, ya sea por el temor al cambio, por falta de recursos, etc. ¿Cuáles son las probabilidades de que Academia Bilingüe Amerrisque (ABA) logre implementar la matrícula online, siendo el primer colegio de Juigalpa Chontales que impulse este desarrollo tecnológico?

R: siempre y cuando sea accesible, con beneficio y desarrollo para la academia, existen probabilidades de que sea implementado en la misma para un mejor beneficio, una seguridad para la información, beneficios de búsqueda de datos y acortar presupuestos de papelería como también la facilidad de matrícula para los padres de familia y queremos implementarlo en la academia.

2. Diagramas de Caso de Uso

- a. *Caso de uso para el usuario Tutor:* El usuario inicia sesión con sus credenciales las cuales son validadas automáticamente identificando los tipos de usuarios que van a acceder al sistema. El usuario puede ejecutar las opciones permitidas para un “Tutor” tales como, <Modificar Contraseña> esta opción está disponible igual para todos los usuarios. La opción de <Lista de estudiante> se da automáticamente al acceder al sistema, es decir, en la pantalla principal se muestra al tutor una tabla con sus estudiantes matriculados para los cuales tiene las opciones de “Matricular” en caso de que comience un nuevo año lectivo y la opción de “Ver Notas” que permite al usuario consultar en línea las calificaciones individuales de sus estudiantes como una entrega de boletines digital, esto en dependencia de cada corte evaluativo. De esta manera el usuario podrá descargar en formato PDF su hoja de calificaciones por estudiante y también estará disponible la opción de mandar a imprimir.

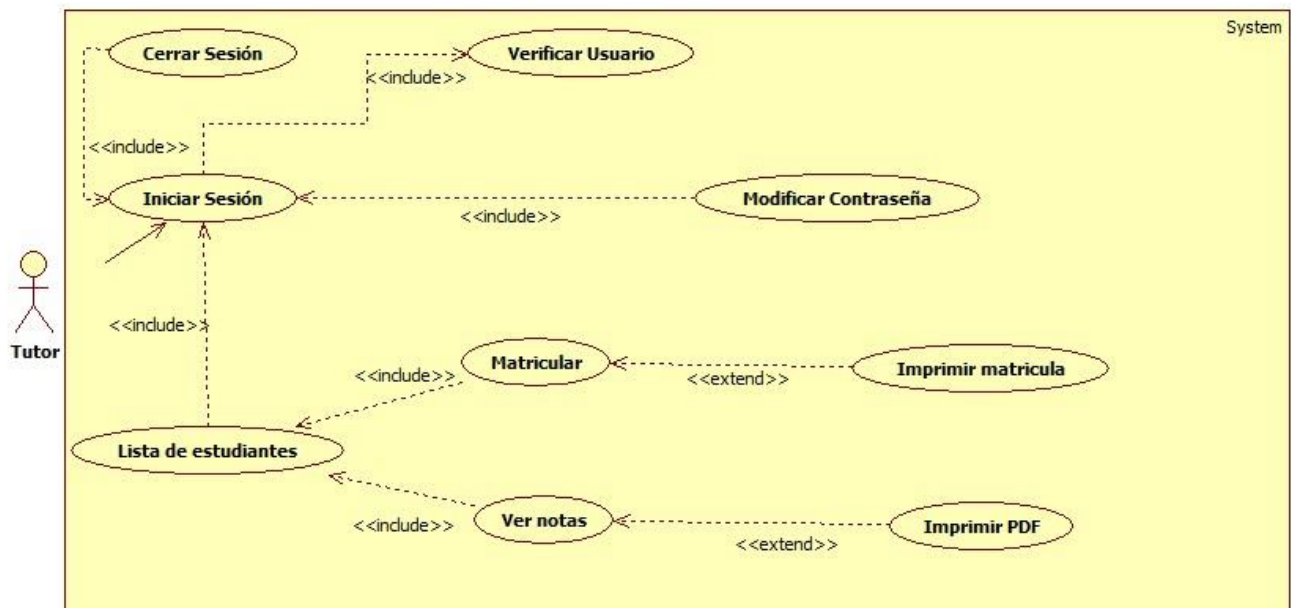


Ilustración 1 - Caso de uso tutor

- b. *Caso de uso de usuario “Docente”*: Al acceder al sistema se le presenta al usuario la lista con los grados que le fueron asignados (Pueden ser más de uno) el usuario selecciona el grado que le interesa y se le muestra una lista con los estudiantes de ese grado, para los cuales tiene la opción de “Calificar” esta opción le permite al usuario escribir la calificación individual para cada

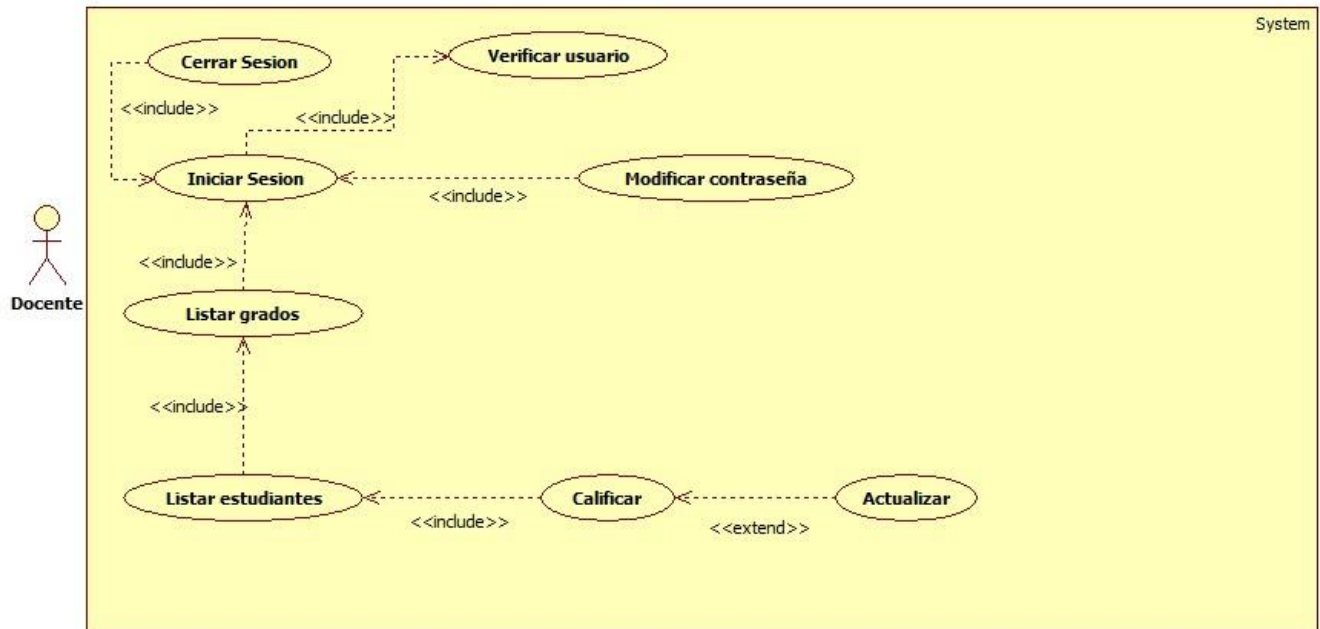


Ilustración 2 - Caso de uso docente

estudiante y para cada asignatura de acuerdo con el corte evaluativo.

- c. *Casos de uso de Usuario “Administrador”*: En esta primera imagen se muestra una parte de las operaciones que puede realizar el Usuario tales como “Iniciar Sesión” y “Cerrar sesión” que son las básicas que todo usuario deberá realizar en este sistema.

Como se puede observar en la imagen el Usuario “Administrador” se encarga de los registros de los demás usuarios del sistema así como el registro de los estudiantes y de su primera matrícula o podrá realizar una matrícula directamente en caso que el Tutor no pueda hacerlo él mismo. Para ello se presenta al usuario la opción de “Matricular” lo que le permite seleccionar el tipo de la matrícula que serían “Reingreso”, “Nuevo ingreso” y “Repitente” y estos tipos de matrícula permiten a su vez la agilización del proceso, por ejemplo, si el tipo es “Reingreso” entonces el usuario no tiene que escribir todos los datos del estudiante o tutor

debido a que sus datos están almacenados en la base de datos del sistema y simplemente se puede buscar al estudiante o tutor por nombre o código y el siguiente paso es actualizar la matrícula e imprimirla si eso se requiere.

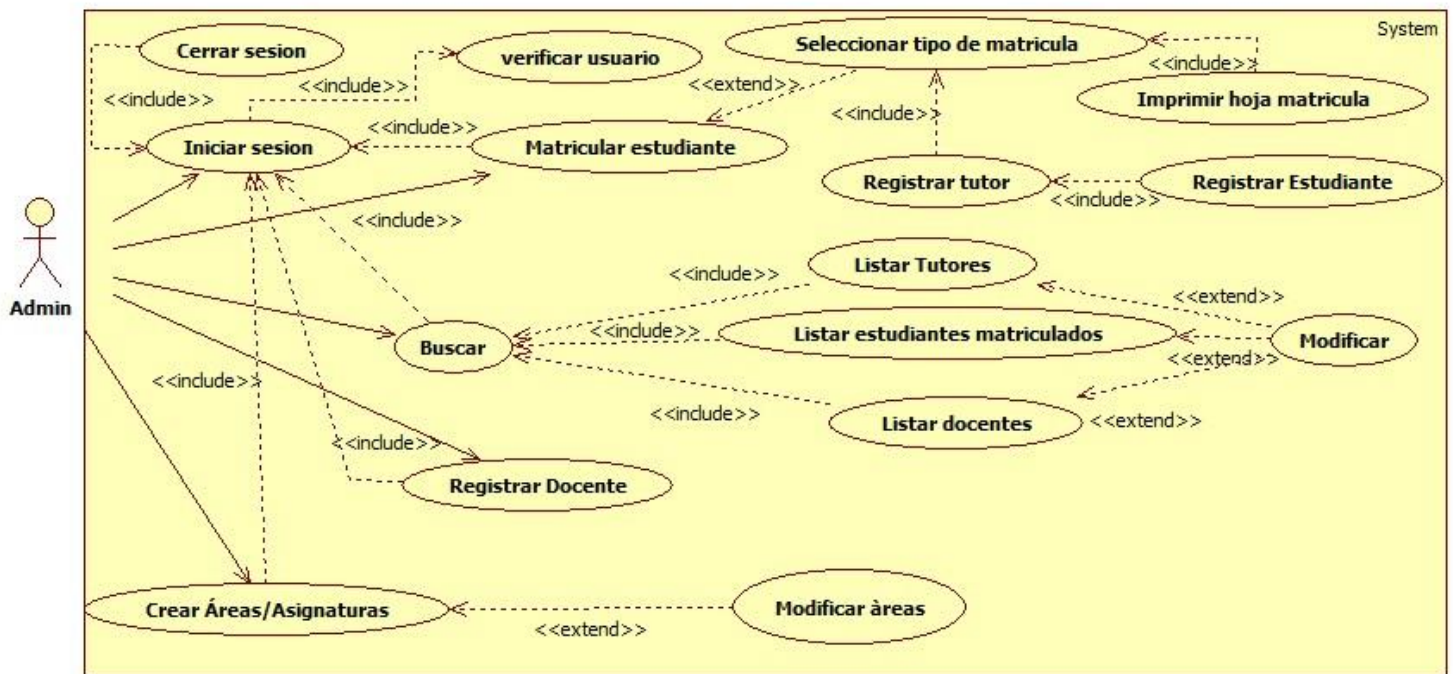


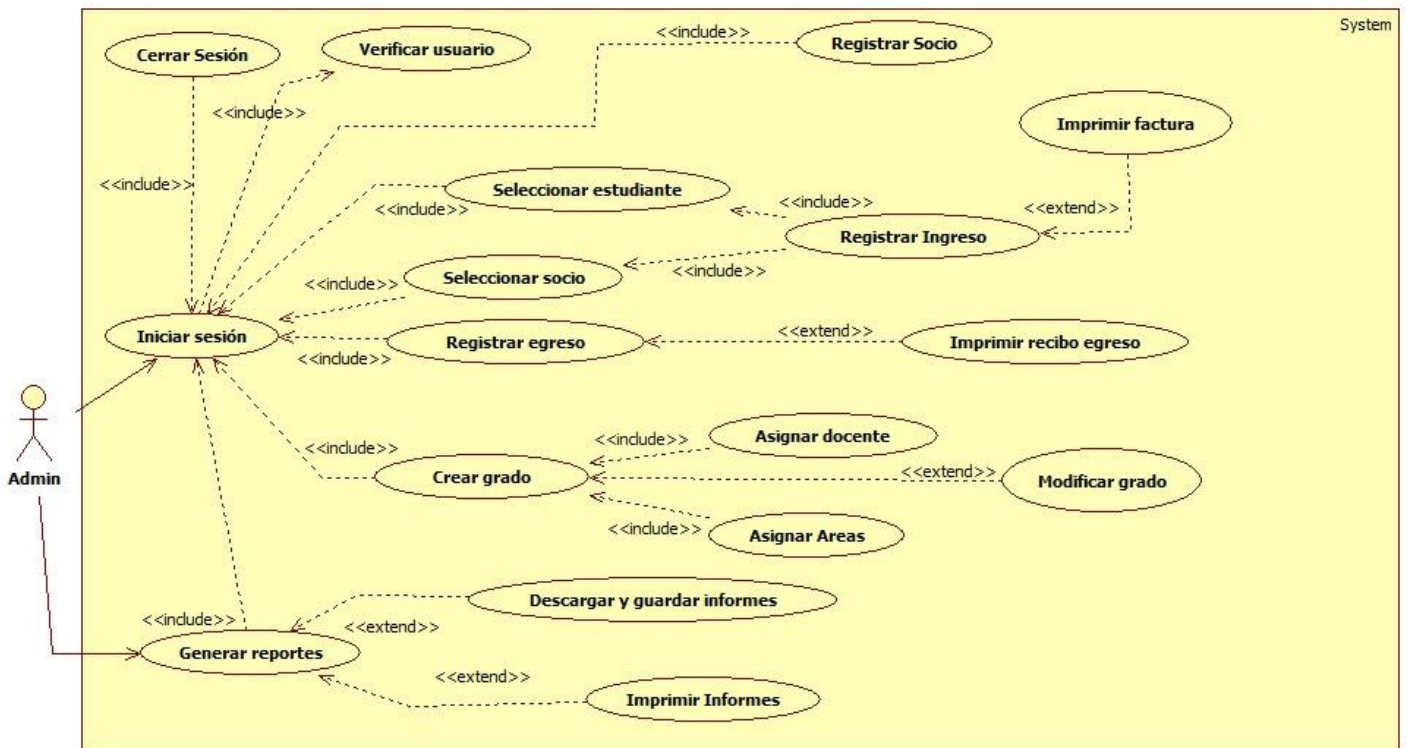
Ilustración 3 - Caso de uso administrador

En la opción de “Buscar” el usuario puede ver en una lista todos los registros que almacena el sistema hasta la actualidad usando diferentes filtros de búsqueda y otras herramientas, después de listar y encontrar la entidad de interés, existe la opción de “Modificar” en caso que se requiera.

La opción “Crear áreas/Asignaturas” se refiere a la creación de las asignaturas que reciben los estudiantes en el año lectivo que cursan generalmente en su grado, se ha decidido manejar las asignaturas en una tabla aparte para controlar las notas por cortes evaluativos para la funcionalidad del software.

Para el módulo de caja que el sistema requiere, se controlan los ingresos y egresos de manera general en donde el usuario Administrador registra los pagos y los gastos de la siguiente manera:

Al registrar una entrada o Egreso los cuales se aplican individualmente por cada estudiante se listan en una tabla a todos los estudiantes en donde el usuario deberá buscar por nombre o por grado al estudiante que vaya a realizar el pago, después se hace un registro del pago para ese estudiante, este recibo o factura se deberá guardar y posteriormente imprimir.



En el caso de los egresos simplemente se registra el gasto con la cantidad y el concepto y se imprime el recibo si así se desea.

Finalmente el Administrador puede generar los reportes o informes que necesite consultando la base de datos de los registros que han sido ejecutados correctamente estos informes pueden ser de Usuarios, de tutores, docentes, estudiantes (por grado, por año lectivo, con cuotas pendientes, etc.), informes de ingresos, informes de egresos, informes de aportes de los socios. Todos estos reportes podrán ser descargados en formato PDF y cuentan con la opción de ser impresos.

3. Base de datos (Diagrama Entidad – Relación)

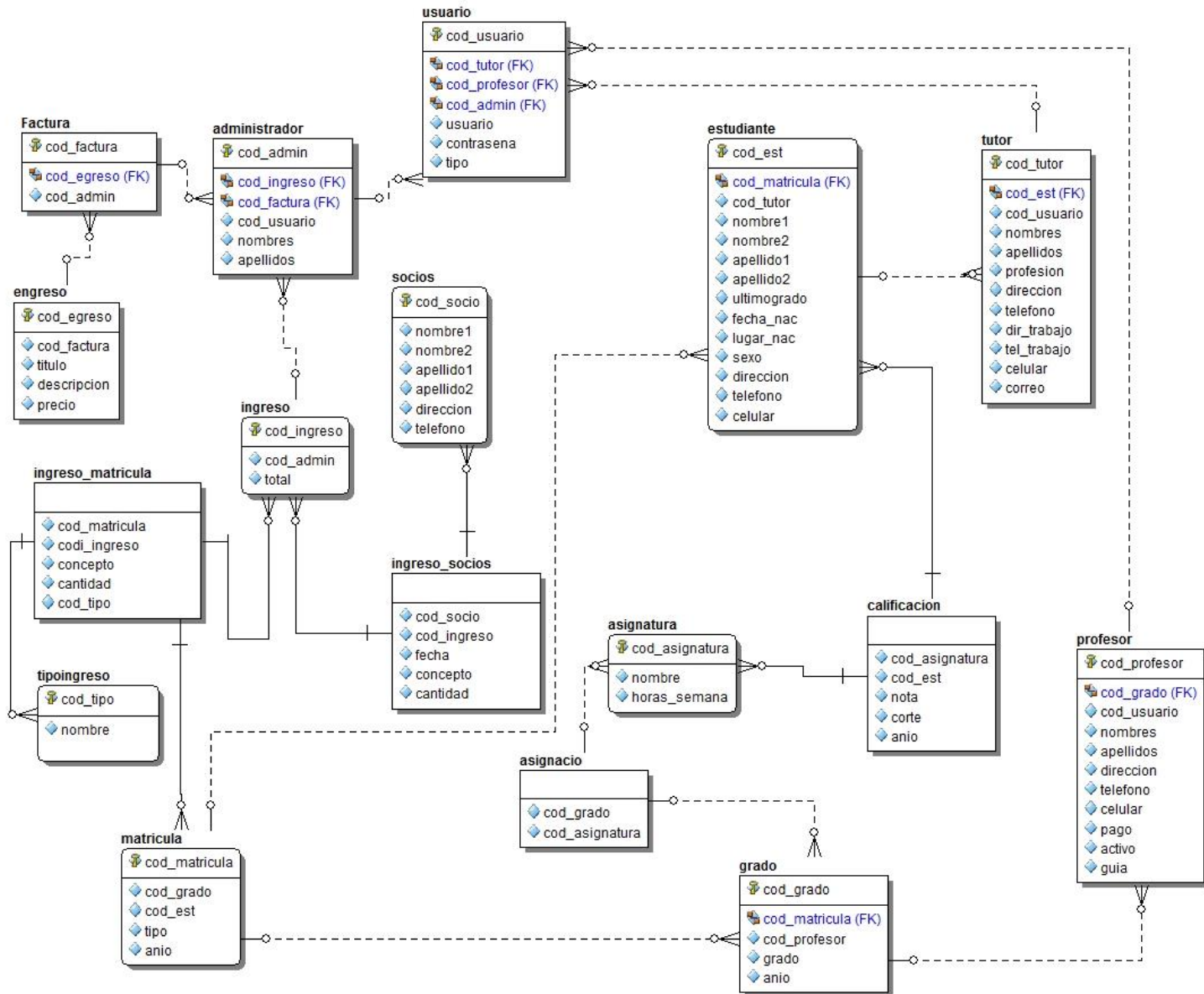


Ilustración 4 - Diagrama de entidad relación

4. Diccionario de Datos

1) Tabla Administrador

| Atributo | Tipo Dato | valores | Descripción |
|-------------|-----------|----------|--|
| cod_admin | Int | 00023 | Código único del administrador |
| cod_usuario | Int | 00012 | Código del usuario el cual controla el administrador |
| Nombres | varchar | Joaquín | Nombre del administrador |
| Apellidos | varchar | Espinoza | Apellidos del administrador |

Tabla 1 - Administrador

2) Tabla asignación

| Atributo | Tipo Dato | valores | Descripción |
|----------------|-----------|---------|--|
| cod_grado | Int | 0004 | Código del grado de la asignación |
| cod_asignatura | Int | 0006 | Código de la asignatura de la asignación |

Tabla 2 - Asignación

3) Tabla asignatura

| Atributo | Tipo Dato | Valores | Descripción |
|----------------|-----------|----------|---|
| cod_asignatura | Int | 0006 | Código único de la asignatura |
| nombre | Text | Robinson | Nombre de la asignatura |
| horas_semana | Int | 4 | Horas a la semana en que se imparte la asignatura |

Tabla 3 - Asignatura

4) Tabla Calificación

| Atributo | Tipo Dato | valores | Descripción |
|----------------|-----------|---------|--|
| cod_asignatura | Int | 0006 | Código de la asignatura calificada |
| cod_est | Int | 0005 | Código del estudiante al que pertenece la calificación |
| Nota | Int | 89 | Calificación asignada en la clase |
| Corte | int | 2 | Corte parcial al que pertenece la calificación |
| Anio | int | 5 | Año lectivo de la calificación |

Tabla 4 - Calificación

5) Tabla Egreso

| Atributo | Tipo Dato | Valores | Valores |
|-------------|-----------|-------------------|---|
| cod_egreso | int | 0006 | Código único del egreso |
| cod_factura | int | 0007 | Código de la factura a la que pertenece el egreso |
| Titulo | text | Pago | Titulo corto del egreso |
| descripcion | text | Pago de matricula | Descripción detallada del egreso |
| Precio | float | 1000 | Precio total del egreso o subtotal de la factura |

Tabla 5 - Egreso

6) Tabla Estudiante

| Atributo | Tipo Dato | valores | Descripción |
|-------------|-----------|-----------------------------------|---|
| cod_est | int | 0006 | Código único del estudiante |
| cod_tutor | int | 0007 | Código del tutor al que está asignado el estudiante |
| nombre1 | varchar | Robinson | Primer nombre del estudiante |
| nombre2 | varchar | Akiles | Segundo nombre del estudiante |
| apellido1 | varchar | Obando | Primer apellido del estudiante |
| apellido2 | varchar | Aragón | Segundo apellido del estudiante |
| ultimogrado | int | 4 | Ultimo grado aprobado por el estudiante |
| fecha_nac | datetime | 12-04-1990 | Fecha de nacimiento del estudiante |
| lugar_nac | varchar | Juigalpa | Lugar de nacimiento del estudiante |
| Sexo | char | Masculino | Sexo del estudiante |
| direccion | text | De la casa de la mujer 2 c al sur | Dirección de domicilio del estudiante |
| telefono | varchar | 51598731 | Teléfono del estudiante |
| Celular | varchar | 88798234 | Celular del estudiante |

Tabla 6 - Estudiante

7) Tabla factura

| Atributo | Tipo Dato | Valores | Descripción |
|-------------|-----------|---------|---|
| cod_factura | Int | 0006 | Código único de la factura |
| cod_admin | Int | 0003 | Código del administrador que realiza la factura |

Tabla 7 - Factura

8) Tabla Grado

| Atributo | Tipo Dato | Valores | Descripción |
|--------------|-----------|---------|---------------------------------------|
| cod_grado | int | 0006 | Código único del grado |
| cod_profesor | int | 0007 | Código del profesor a cargo del grado |
| Grado | int | 4 | Grado que representa el registro |
| Anio | int | 2016 | Año lectivo y de creación del grado |

Tabla 8 - Grado

9) Tabla Ingreso

| Atributo | Tipo Dato | Valores | Descripción |
|-------------|-----------|---------|--|
| cod_ingreso | Int | 0006 | Código único del ingreso |
| cod_admin | Int | 0004 | Código del administrador que registra el ingreso |
| Total | Int | 20000 | Total del ingreso |

Tabla 9 - Ingreso

10) Tabla Ingreso Matricula

| Atributo | Tipo Dato | Valores | Descripción |
|---------------|-----------|---------------------|--|
| cod_matricula | int | 0006 | Código de la matricula a la que pertenece el detalle |
| cod_ingreso | int | 0007 | Código del ingreso al que pertenece el detalle |
| concepto | text | Pago de mensualidad | Descripción del ingreso |
| cantidad | float | 1000 | Total del ingreso |
| cod_tipo | int | 0008 | Código del tipo de ingreso |

Tabla 10 - Ingreso de Matricula

11) Tabla ingreso socio

| Atributo | Tipo Dato | Valores | Descripción |
|-------------|-----------|-----------------|--|
| cod_socio | int | 00002 | Código del socio que realiza el ingreso |
| cod_ingreso | int | 00003 | Código del ingreso al que pertenece el detalle |
| Fecha | datetime | 12-01-2016 | Fecha de pago de ingreso |
| concepto | text | Aporte de socio | Concepto del pago |
| cantidad | float | 2000 | Cantidad pagada |

Tabla 11 - Ingreso socio

12) Tabla Matricula

| Atributo | Tipo Dato | valores | Descripción |
|---------------|-----------|-----------|------------------------------------|
| cod_matricula | int | 0003 | Código único de la matricula |
| cod_grado | int | 0004 | Código del grado a matricular |
| cod_est | int | 0006 | Código del estudiante a matricular |
| Tipo | int | Reingreso | Tipo de la matricula |
| Anio | int | 2016 | Año en que se realiza la matricula |

Tabla 12 - Matricula

| Atributo | Tipo Dato | valores | Descripción |
|--------------|-----------|---------|---|
| cod_profesor | Int | 0008 | Código único del profesor |
| cod_usuario | Int | 0009 | Código del usuario al que pertenece el profesor |
| Nombres | Varchar | Saira | Nombres del profesor |
| Apellidos | Varchar | López | Apellidos del profesor |

| | | | |
|-----------|---------|--------------------------------------|-------------------------------------|
| Dirección | Text | Colegio Regina mundi 1/5 al norte | Dirección de domicilio del profesor |
| Teléfono | Varchar | 51297654 | Teléfono del profesor |
| Celular | Varchar | 88756432 | Celular del profesor |
| Pago | money | 4000 | Pago mensual del profesor |

13) Tabla Profesor

Tabla 13 - Profesor

14) Tabla Socio

| Atributo | Tipo Dato | valores | Descripción |
|-----------|-----------|--------------------------|----------------------------------|
| cod_socio | Int | 0006 | Código único del socio |
| nombre1 | Varchar | fernando | Primer nombre del socio |
| nombre2 | Varchar | Josue | Segundo nombre del socio |
| apellido1 | varchar | Perez | Primer apellido del socio |
| apellido2 | varchar | Treminio | Segundo apellido del socio |
| direccion | text | Frente a la cruz roja | Dirección de domicilio del socio |
| telefono | varchar | 58128790 | Teléfono del socio |

Tabla 14 - Socio

15) Tablas Tipo Ingreso

| Atributo | Tipo Dato | valores | Descripción |
|----------|-----------|----------------------|----------------------------------|
| cod_tipo | int | 10 | Código único del tipo de ingreso |
| nombre | text | Pago de matricula | Nombre del tipo de ingreso |

Tabla 15 -Ingreso

16) Tabla Tutor

| Atributo | Tipo Dato | Valores | Descripción |
|----------|-----------|---------|-------------|
|----------|-----------|---------|-------------|

| | | | |
|-------------|----------|---------------------------|---|
| cod_tutor | int | 0008 | Código único del tutor |
| cod_usuario | int | 0009 | Código del usuario que pertenece al tutor |
| nombres | vvarchar | Juan José | Nombres del tutor |
| apellidos | vvarchar | Espinoza Lara | Apellidos del tutor |
| profesion | text | Doctor | Profesión del tutor |
| direccion | text | De la curasao 2 al sur | Dirección de domicilio del tutor |
| telefono | vvarchar | 51286754 | Teléfono del tutor |
| dir_trabajo | text | Del cementerio 3 al sur | Dirección del trabajo del tutor |
| tel_trabajo | vvarchar | 51280967 | Teléfono del trabajo del tutor |
| Celular | vvarchar | 88723872 | Celular del tutor |
| Correo | text | robinson9obando@gmail.com | Correo electrónico del tutor |

Tabla 16 - Tutor

17) Tabla Usuario

| Atributo | Tipo Dato | valores | Descripción |
|-------------|-----------|--------------|--------------------------|
| cod_usuario | Int | 00008 | Código único del usuario |
| Usuario | Vvarchar | Robin23 | Nombre de usuario |
| Contraseña | Vvarchar | Megafacil123 | Contraseña del usuario |
| Tipo | Int | Alumno | Tipo del usuario |

Tabla 17 - Usuario

5. Mapa de Navegación

El mapa de navegación muestra la ruta que toman las opciones a las que puede optar el usuario al momento de utilizar el sistema, se establece un orden jerárquico que inicia desde la pantalla principal dependiendo del tipo de usuario y luego los menús y submenús para las transacciones que le permite el software al usuario que lo está utilizando.

a) Usuario “Tutor”

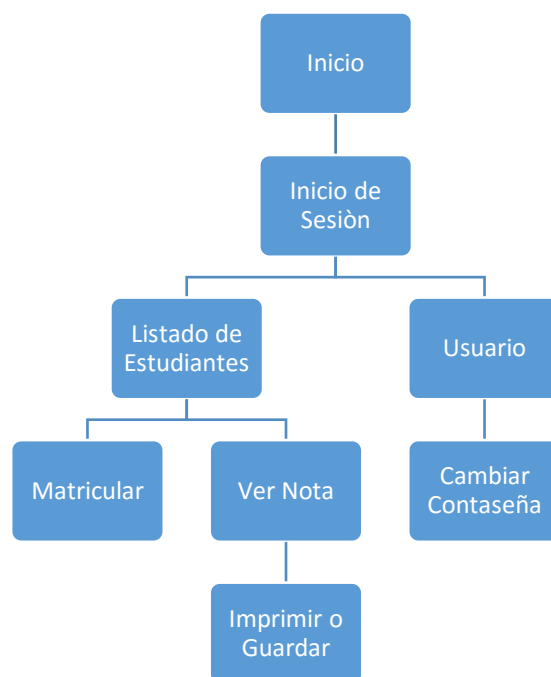


Ilustración 5 – Mapa de navegación Usuario tutor

Para este usuario la ruta de la aplicación es la que podemos observar en la ilustración, en la cual el primer nivel en orden jerárquico es la página de inicio que muestra información sobre la academia y a la vez un poco de publicidad. Es esta página el usuario activa el inicio de sesión y accede al sistema con sus credenciales, automáticamente se le presentan las opciones que le permite el sistema a un tutor. Para este usuario las opciones disponibles son las de Cambiar sus datos (Contraseña) y ver el listado de sus estudiantes, en este punto se muestran otras dos opciones “Matricular” y “ver notas”, la opción de ver notas lleva al usuario a una tabla con las calificaciones del estudiante que ha seleccionado y le permite guardar esa información o imprimir si lo requiere.

b) Usuario “Docente”

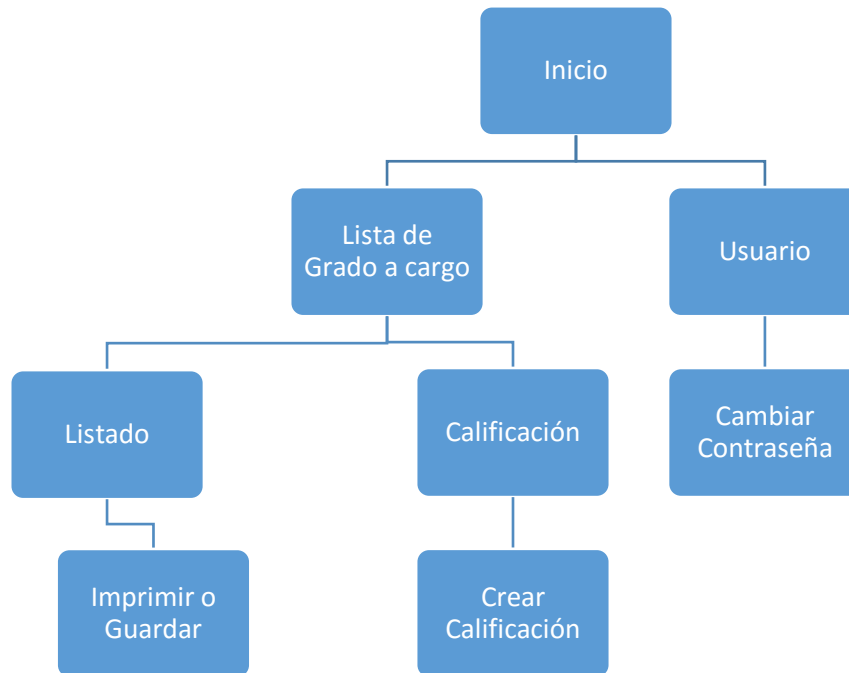


Ilustración 6 - Mapa de navegación usuario docente

Del mismo modo el usuario Docente accede siguiendo el orden jerárquico que va desde la página de inicio del sistema, introduce sus credenciales de usuario y se le presentan la opción de cambiar su contraseña y listar sus grados. Al seleccionar un grado de la lista se muestra al usuario las opciones de listado que es una tabla en donde aparecen todos los estudiantes de ese grado en una eschela la cual puede imprimir para uso didáctico (Asistencia, Registro de calificaciones, etc.). En la opción de “Calificación” el sistema lleva al docente al crear las calificaciones para cada estudiante.

c) Usuario “Administrador”

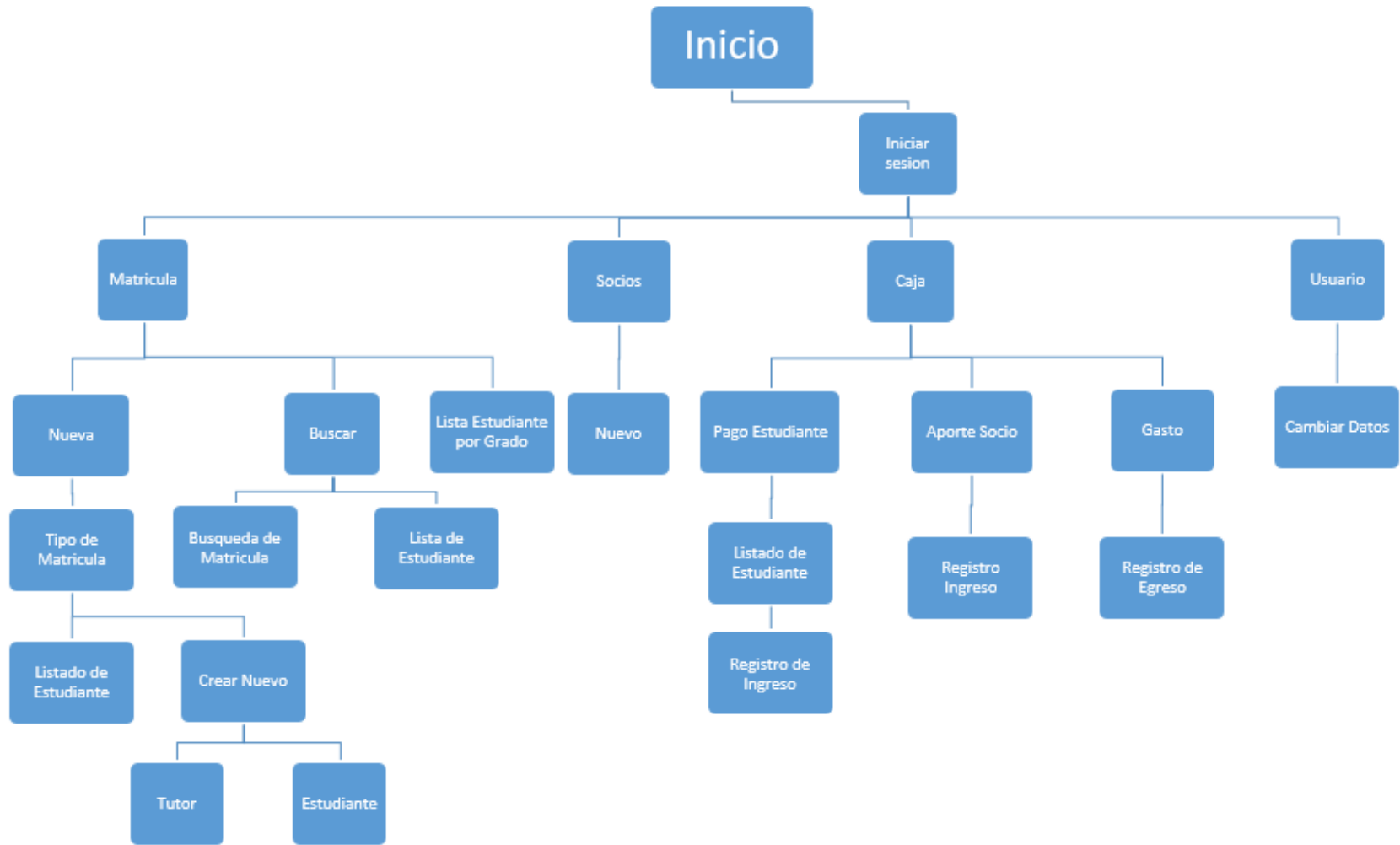


Ilustración 7 - Mapa de navegación usuario administrador

El inicio de sesión para este usuario es el mismo orden jerárquico que siguen los demás usuarios. El administrador posee más opciones como se puede observar en la ilustración, en el tercer nivel de la jerarquía se muestran los menús de la página principal del administrador que conllevan a las demás opciones.

Matricula: Esta opción despliega opciones de Nueva, Buscar y Lista de estudiante. El submenú de “Nueva” dirige a la página de Tipo de Matricula la cual presenta otras dos nuevas opciones, Listado de estudiantes y “Crear nuevo” para crear un nuevo estudiante se necesita primero crear a un tutor, aquí se llenan los formularios de matrícula con los datos generales del tutor y des nuevo estudiante. El submenú de “Buscar” genera las opciones de búsqueda de matrículas, los cuales son informes de matrículas que ya fueron creadas, y la opción de listar estudiantes genera también una búsqueda conglomerada de los estudiantes que ya están registrados en el sistema.

Socios: Este menú direcciona a un formulario que permite al administrador registrar a un nuevo socio en el sistema, los cuales no desempeñan ninguna función directa en el manejo del software pero han sido requeridos por el cliente.

Caja: Este menú permite al administrador controlar los pagos y gastos de la institución de él se despliegan las opciones de Pago de estudiantes, aporte de socios y Gastos. La opción de pago de estudiante lleva al usuario a un listado de estudiantes para los cuales se permite Registrar un ingreso por estudiante de forma individual. El submenú “Aporte de socios” representa otro tipo de ingreso que lleva al administrador a registrar este aporte en el sistema. La opción de “Gastos” permite al administrador registrar este egreso y generar una factura.

Usuario: Permite al administrador el cambio de contraseña.

6. Bocetos del Sistema.

- a. **La pantalla de presentación** llevará el logo y el lema de la Academia. De igual forma se agregará si así se prefiere un carrusel con imágenes actualizadas de las principales actividades realizadas en la academia, cuyas imágenes se podrán visualizar antes de iniciar sesión en el sistema como un tipo de página web la cual brindará a su vez publicidad e información de la Academia a todo aquel que desee consultar.

Esta página mostrará información como: Misión y Visión de la Academia, Imágenes actualizadas de las actividades de la Academia, Excelencias Académicas (Nombres de los estudiantes con sus fotografías), Noticias y Novedades con respecto a eventos y actividades generales que se celebran en la Academia y cursos alternos extracurriculares que se imparten con frecuencia en este colegio, Promociones de inscripciones, descuentos de matrícula, etc.

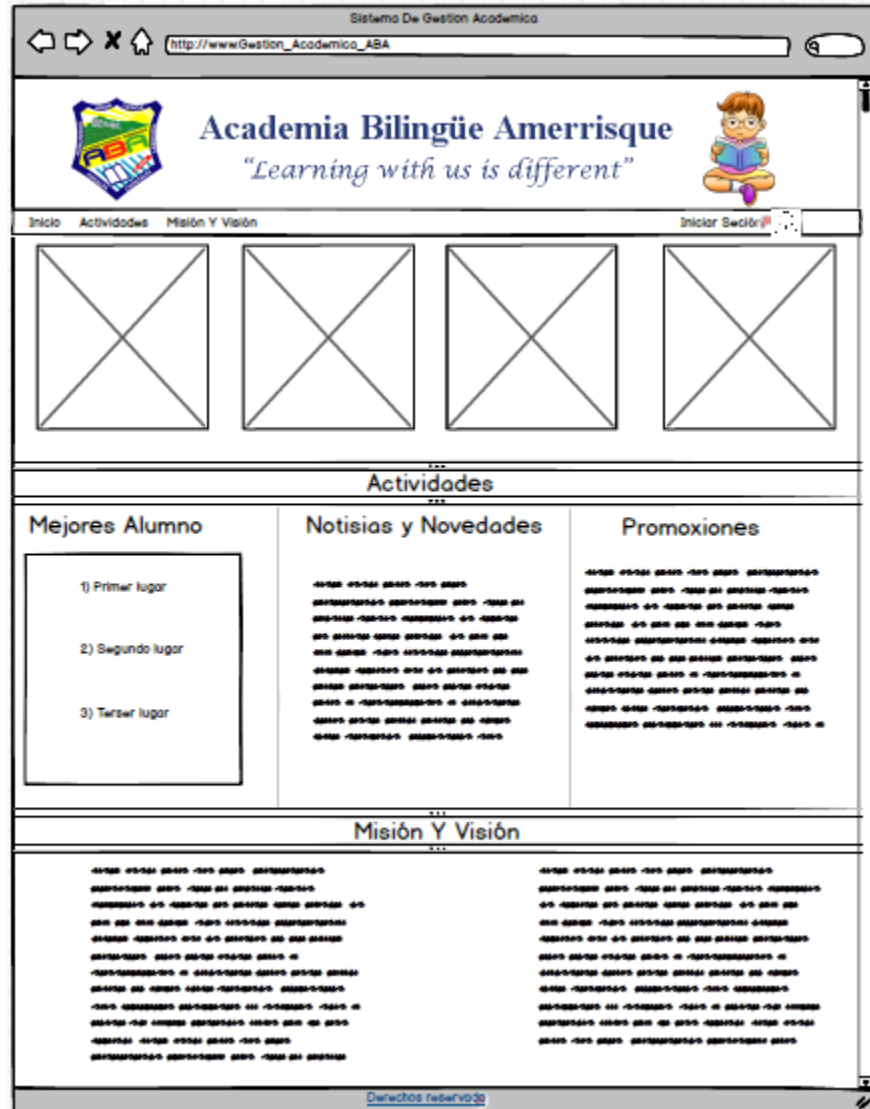


Ilustración 8 - Bocetos de la página principal

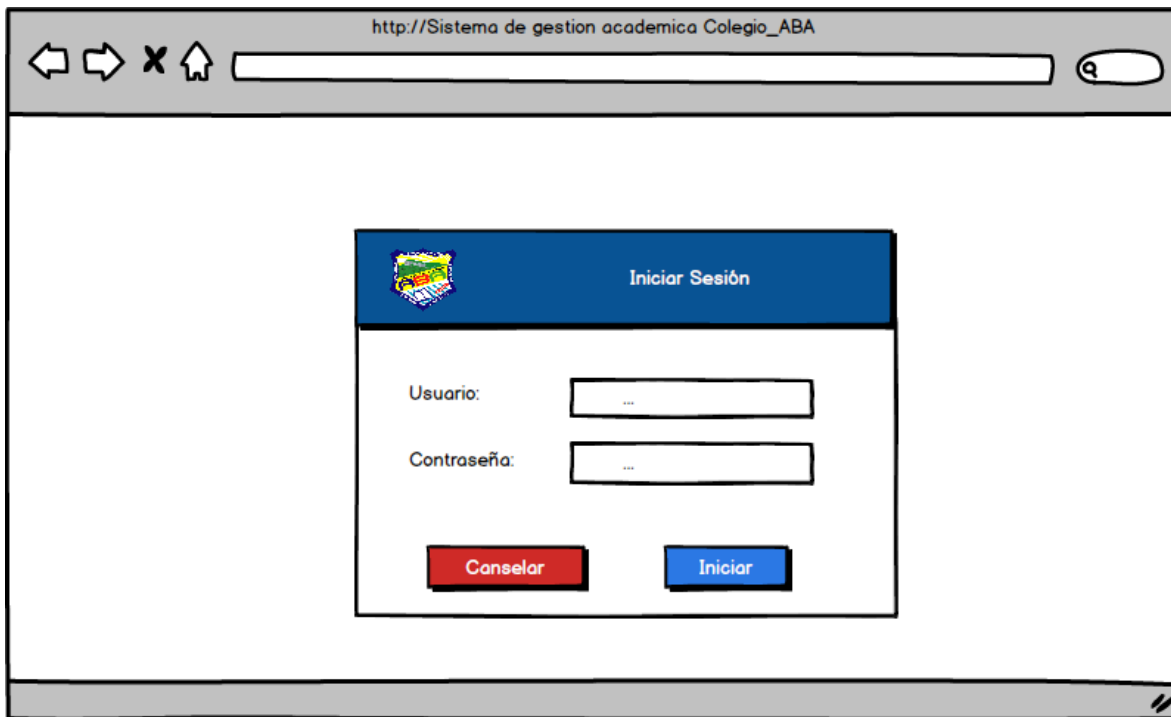


Ilustración 9 - Boceto inicio de sesión

- b. **La pantalla del login:** Esta pantalla será desplegada al dar clic en un botón que indica el *Inicio de Sesión*, en donde el usuario deberá escribir sus credenciales para acceder al sistema a realizar sus transacciones.

De igual forma la pantalla muestra el logo y el lema de la Academia como en todas las pantallas.



Ilustración 10 - Boceto página principal administrador

- c. **Página Principal del Usuario Administrador:** En esta página del sistema, a la cual solo podrá acceder el Administrador con sus credenciales, se permitirá gestionar todos los Registros de Usuarios, de la misma manera el Administrador puede acceder al control de caja en donde podrá manejar las operaciones financieras de la institución.

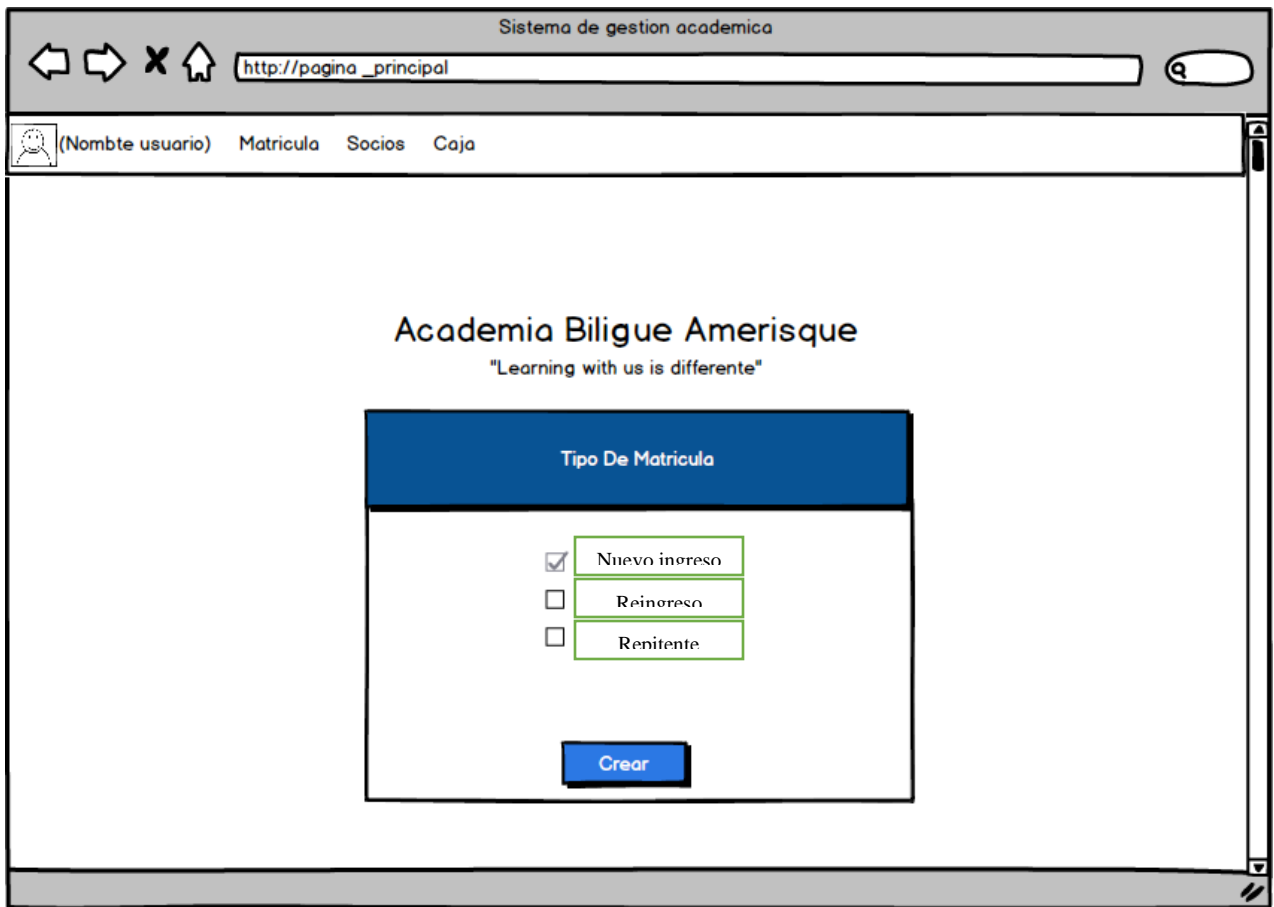


Ilustración 11 - Bocetos página tipo de matricula

Al seleccionar una nueva matricula se le pregunta al usuario el tipo de matrícula que en este caso pueden ser “Nuevo ingreso”, “Reingreso”, y “Repitente” para poder pasar a la siguiente pantalla en caso de que sea un Nuevo ingreso:

Esta pantalla pregunta si se va a registrar un Nuevo Tutor o un Nuevo estudiante para realizar la vinculación entre el estudiante y el tutor.

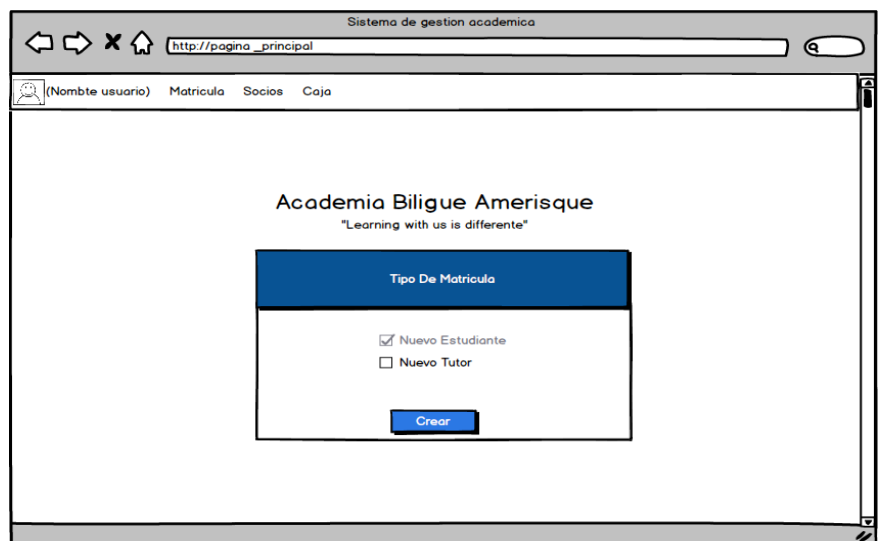


Ilustración 12 -Boceto tipo de matricula

Esta es la pantalla para registrar a un Nuevo Tutor para el cual se llena la información y como podremos observar en la imagen además en este formulario se llenan los datos de usuario con nombre de usuario y contraseña y se crea un nuevo usuario del sistema.

The image shows a web browser window with the address bar displaying "http://Sistema de gestion academica Colegio_ABA". The main content area contains a registration form titled "Datos Del Tutor". The form has a blue header with a small logo on the left and the title "Datos Del Tutor" on the right. Below the header, there are several input fields for personal and contact information. A red rectangular box highlights the "Usuario", "Contraseña", and "Confirmar Contraseña" fields. At the bottom of the form, there are two buttons: a red "Cancelar" button and a blue "Registrar" button.

| | |
|------------------------|----------------------|
| Nombres | |
| ... | |
| Apellidos | |
| ... | |
| Usuario | |
| ... | |
| Contraseña | Confirmar Contraseña |
| ... | ... |
| Telefono | Celular |
| ... | ... |
| Correo Electronico | |
| ... | |
| Direccion de domicilio | |
| ... | |
| Ocupacion | Telefono del trabajo |
| ... | ... |
| Direccion del trabajo | |
| ... | |

Ilustración 13 - Boceto Datos Tutor

Automáticamente después de guardar los datos del Tutor el sistema dirige al usuario a llenar los datos del estudiante para concretar la primera matrícula.

Como se puede observar en la imagen en el formulario para registrar a un Nuevo estudiante se hace la asignación del Tutor al estudiante. Y al registrar al estudiante este se agrega a la lista de estudiantes del tutor al que fue asignado.

The image shows a web browser window with the URL `http://Sistema de gestion academica Colegio_ABA`. The browser's address bar contains navigation icons (back, forward, stop, home) and a search icon. The main content area displays a form titled "Nuevo Estudiante" with a blue header bar. The form is organized into two columns of input fields. The left column contains: "Primer Nombre", "Primer Apellido", "Sexo:" with radio buttons for "Femenino" (checked) and "Masculino", "Direccion", "Fecha De Nacimiento", and "Lugar De Nacimiento". The right column contains: "Segundo Nombre", "Segundo Apellido", "Ultimo Grado Apro", and three "Contraseña:" fields. At the bottom of the form, a dropdown menu is highlighted with an orange border, containing the text "-Seleccione un Tutor-". Below the form are two buttons: a red "Cancelar" button and a blue "Registrar" button.

Ilustración 14 – Bocetos Nuevo estudiante

La siguiente pantalla del sistema registra los datos personales de un Nuevo Socio, los cuales realizan un aporte a la institución que genera un “Nuevo Ingreso” y estos deben ser controlados por el sistema.

The image shows a web browser window with the address bar displaying "http://Sistema de gestion academica Colegio_ABA". The page title is "Academia Biligue Amerisque" with the tagline "Learning with us is diferente". The main content area features a blue header with a logo and the text "Datos De Socios". Below this, there is a form with six input fields arranged in two columns. The left column contains "Primer Nombre", "Primer Apellido", and "Direccion". The right column contains "Segundo Nombre", "Segundo Apellido", and "Telefono". Each field has a small "..." placeholder. At the bottom of the form are two buttons: a red "Cancelar" button and a blue "Registrar" button.

Ilustración 15 – Boceto Datos de socio

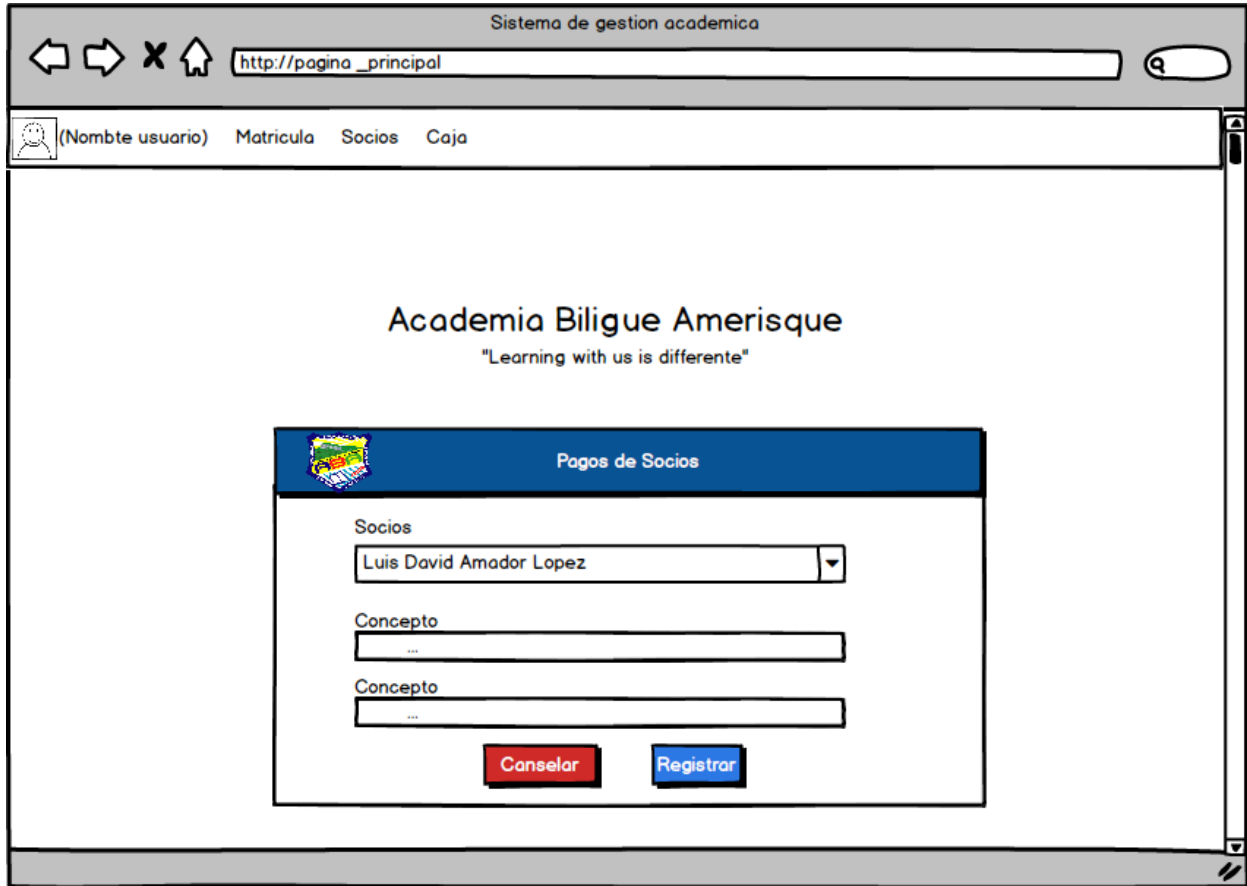


Ilustración 16 - Pago de socios

Como lo muestra la imagen un aporte de los socios se registra en el sistema de esta manera, primero selecciona el nombre del socio que debió haber sido registrado en el sistema anteriormente y después se realiza la operación simplemente escribiendo el concepto y la cantidad.

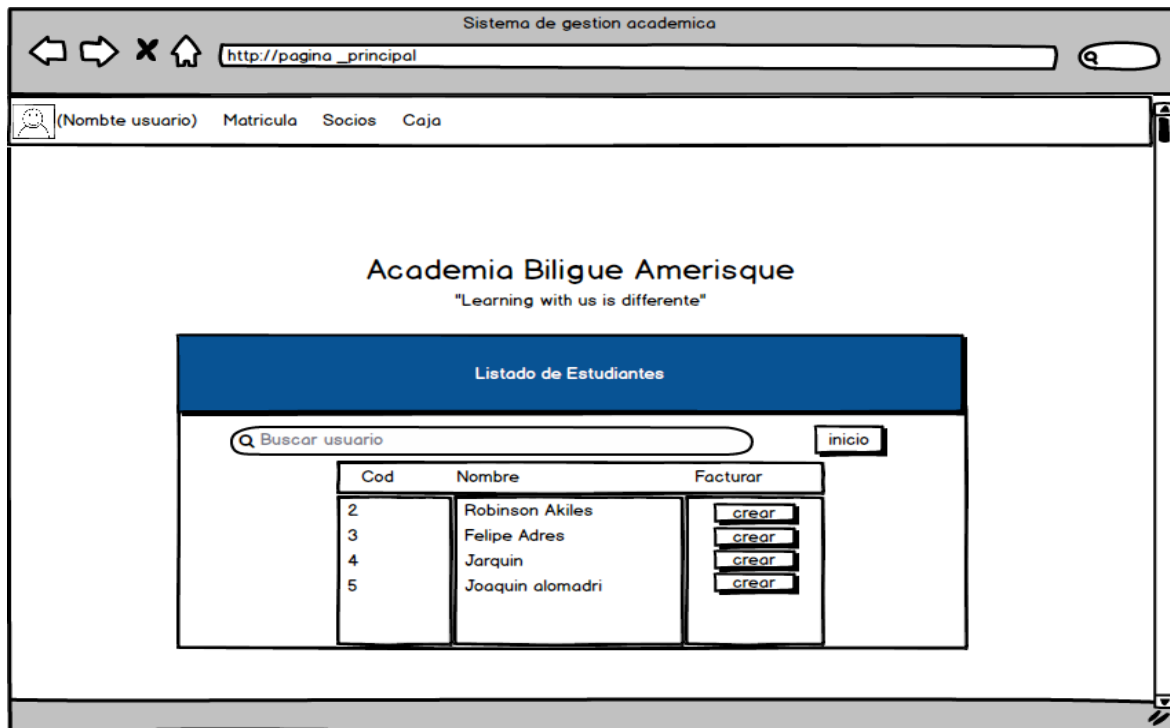


Ilustración 17 - Listado de estudiante

Esto mismo sucede con el registro de un Nuevo ingreso, para este el usuario selecciona el estudiante que realiza en pago y se procede a crear una nueva factura en donde se puede seleccionar el tipo de pago que realiza, los cuales están definidos en un combobox y contienen los precios ya definidos.

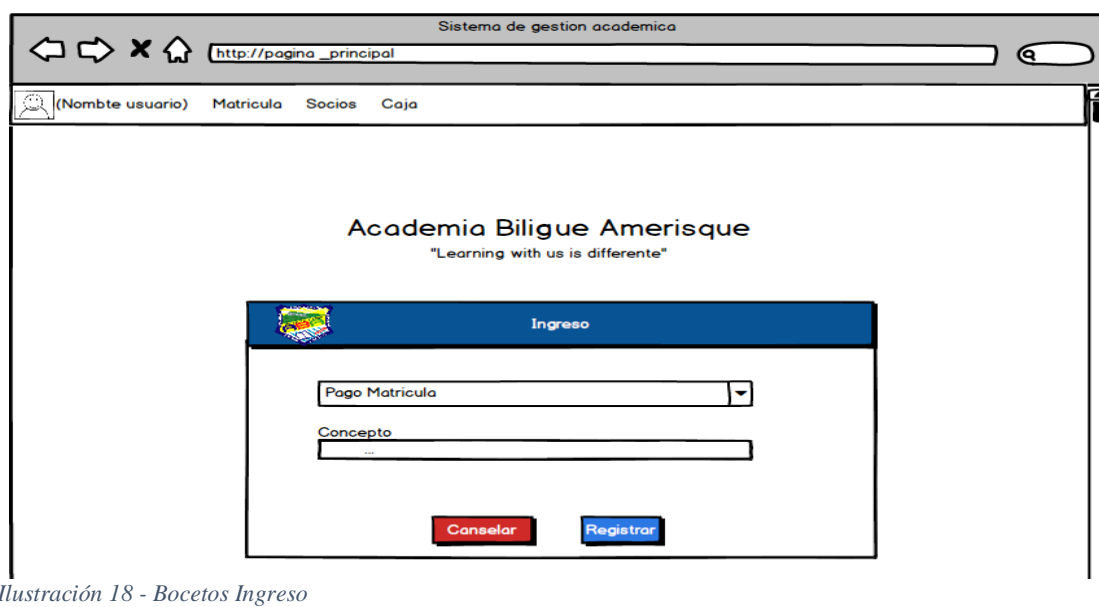


Ilustración 18 - Bocetos Ingreso

De igual forma para registrar los Egresos se deberá llenar el formulario con concepto y cantidad y dar clic en el botón Registrar.

The image shows a web browser window titled "Sistema de gestion academica". The address bar contains "http://pagina_principal". The browser's navigation bar includes a search icon and a search input field. Below the navigation bar, there is a user profile section with a placeholder for a user icon and the text "(Nombte usuario) Matricula Socios Caja". The main content area displays the logo and name of "Academia Biligue Amerisque" with the tagline "Learning with us is diferente". Centered on the page is a form titled "Egreso" with a blue header. The form contains two input fields, each labeled "Concepto" with a small "..." icon to its right. At the bottom of the form are two buttons: a red "Cancelar" button and a blue "Registrar" button.

Ilustración 19 - Bocetos Egreso

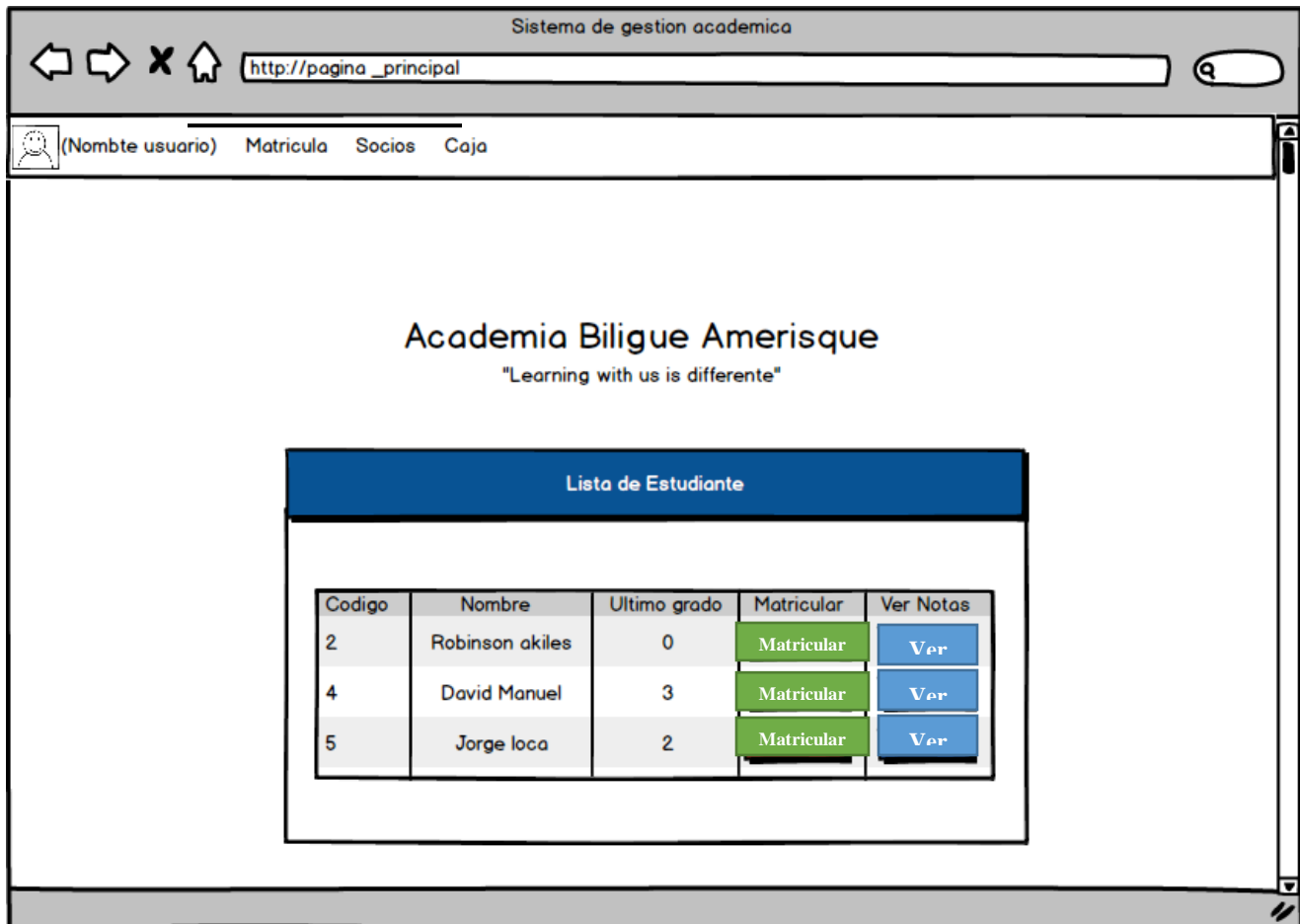


Ilustración 20 - Lista de datos de estudiante

- d. **Página principal del Usuario Tutor:** En esta página, el Tutor, accede e inmediatamente se le muestra una tabla con la lista de los estudiantes que están registrados bajo su responsabilidad, para los cuales puede visualizar a la par del nombre las opciones de Matricular y Ver en formas de botones que les permite Matricular al estudiante consultar calificaciones en línea de la siguiente manera:

A Web Page

http://

LOGO

Academia Biligüe Amerisque
"Learning with us is diferente"

Calificación Estudiantil

Nombre: Robinson Akiles Obando Aragon

Numero _Id: 080893 **Año Matriculado:** 5to

| Materia: | 1er C. | 2do C. | I Semestre | 3er C. | 4to C. | II Semestre | Nota Final ▼ |
|-------------|--------|--------|------------|--------|--------|-------------|--------------|
| Español | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| Matemática | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| Ciencia | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| Sociales | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| Computación | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| Cívica | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| EE-FF | 80 | 80 | 80 | 80 | 80 | 80 | 80 |
| Inglés | 80 | 80 | 80 | 80 | 80 | 80 | 80 |

Docente a cargo: 20 / 09 / 2016

Directora: Mirian Patrisia Telléz

[Imprimir](#) [Guardar](#)

Ilustración 21 - Boceto de Calificación

Como podemos observar en la imagen con la opción de “Ver” se muestra al tutor una especie de boletín escolar en el que se muestran el nombre del estudiante, el grado que cursa, las calificaciones de las asignaturas, los cortes evaluativos y al final el usuario tendrá la opción de guardar en formato PDF y si lo requiere imprimir su acta de calificaciones.

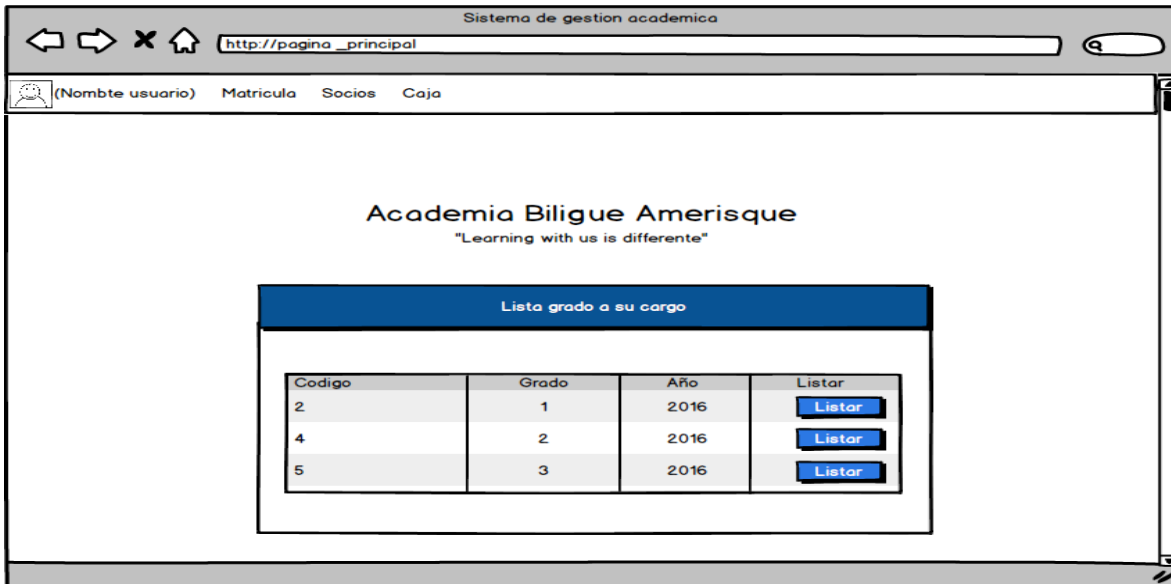


Ilustración 22 - Lista de grado a su cargo

En el caso de Usuario “Docente” al acceder al sistema le seran mostrados los grados que tiene a su cargo, en ellos estan las listas de sus estudiantes entonces al dar clic al Boton “Listar” del grado que le interese es sistema redirecciona el usuario a la siguiente pantalla

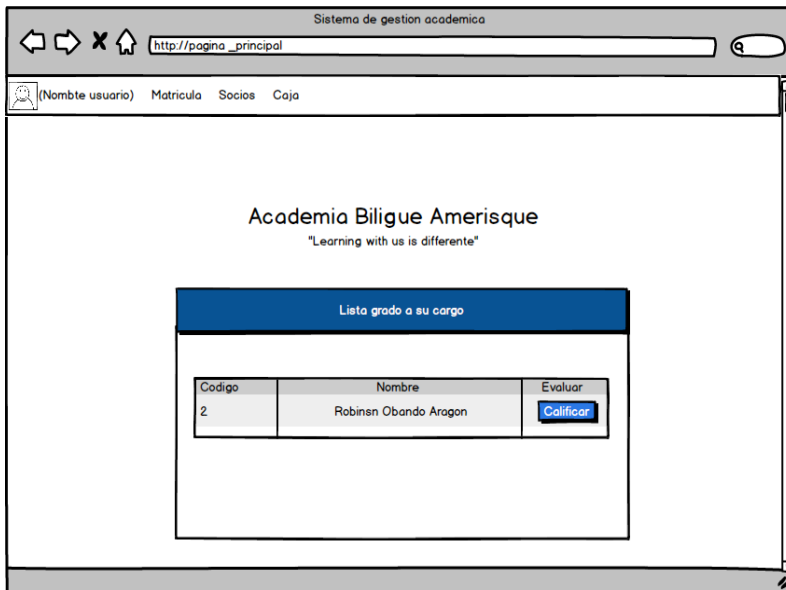


Ilustración 23 Bocetos grado a cargo

El Sistema muestra al Docente un listado de sus estudiantes para los cuales puede escribir la calificacion correspondiente a los cortes evaluativos y se muestra una pantalla con el siguiente formulario:

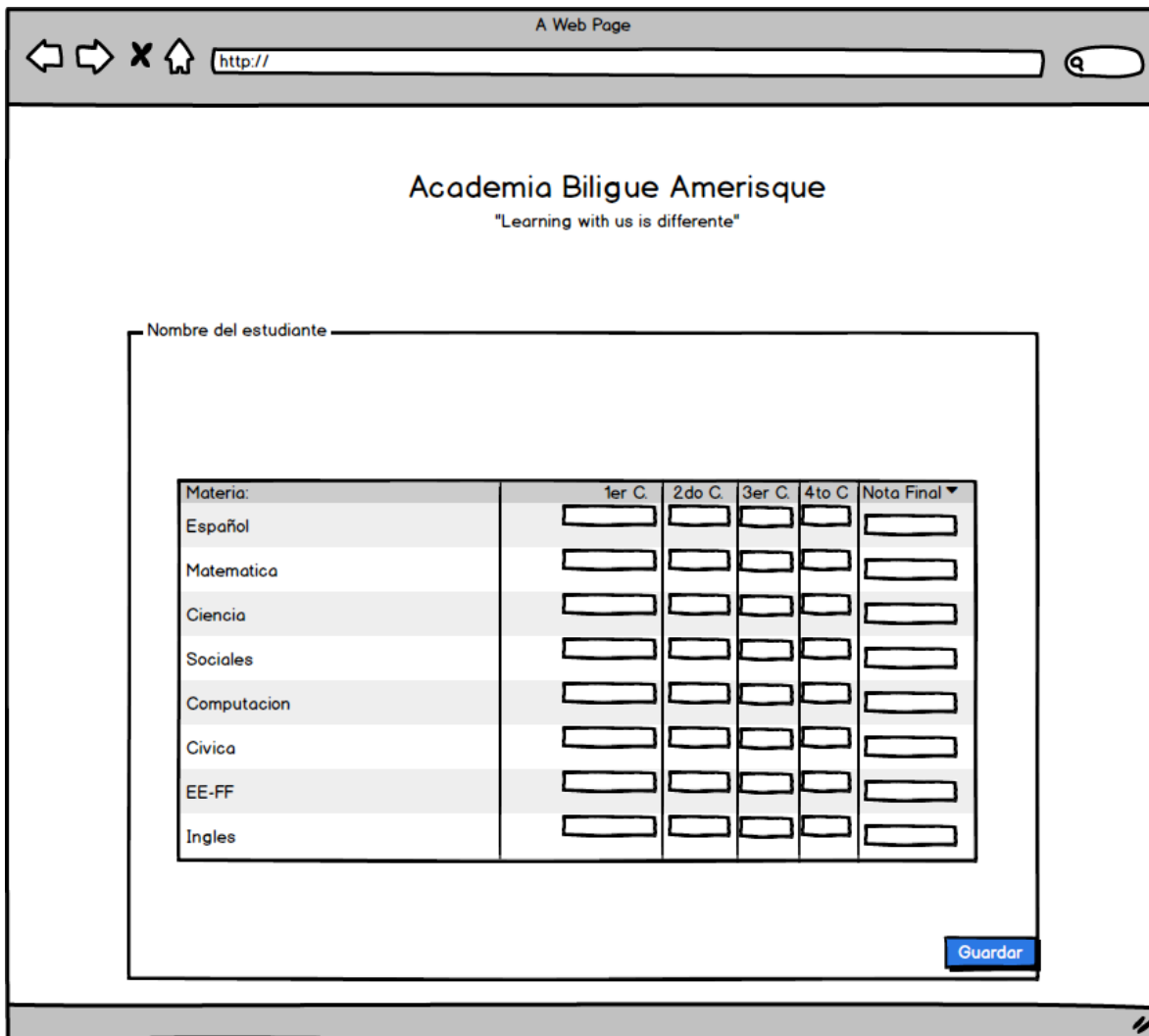
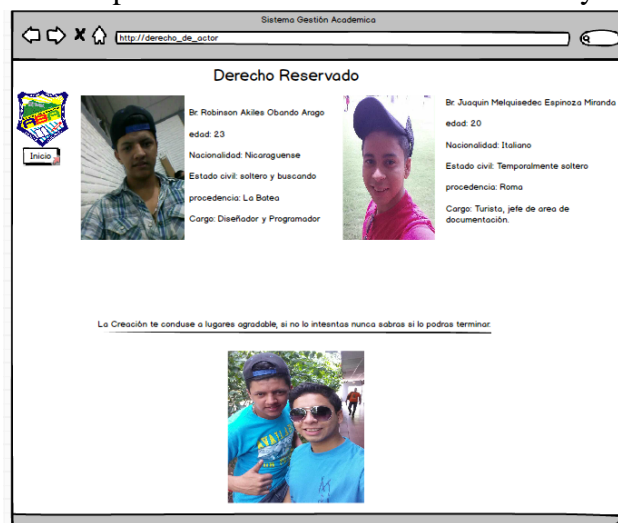


Ilustración 24 - Bocetos Crear Calificación

En esta pantalla el usuario puede escribir la calificación para cada estudiante y luego subirla para que la pueda ver el Tutor. Esto se hace en dependencia de cada corte evaluativo y el sistema calcula automáticamente la nota de los semestres y la nota final.

Información adicional de los creadores del sistema.



Documento SRS

Especificación de requisitos de software

Proyecto: Sistema de Información Web para la gestión Académica del Colegio ABA.

Revisión 010.



Enero del 2017

Contenido

| | |
|---|----|
| 1. (SRS) Ficha del documento..... | 1 |
| 2. Introducción | 2 |
| 3. Propósito..... | 2 |
| 4. Alcance..... | 2 |
| 5. Personal involucrado..... | 3 |
| 6. Definiciones, acrónimos y abreviaturas | 4 |
| 7. Referencias..... | 5 |
| 8. Resumen..... | 5 |
| 9. Descripción general..... | 5 |
| a. Perspectiva del producto | 5 |
| 10. Funcionalidad del producto. | 6 |
| 11. Características de los usuarios..... | 8 |
| 12. Restricciones..... | 8 |
| 13. Suposiciones y dependencias..... | 9 |
| 14. Evolución previsible del Sistema | 9 |
| 15. Requisitos específicos..... | 9 |
| 16. Interfaces de usuario..... | 9 |
| 17. Interfaces de hardware..... | 10 |
| 18. Interfaces de software..... | 10 |
| 19. Interfaces de comunicación | 10 |
| 20. Requisitos funcionales..... | 10 |
| 21. Requisitos no funcionales..... | 14 |

Índice de tablas

| | |
|---|---|
| Tabla 1 - Ficha del documento | 1 |
| Tabla 2 - Documento Validado | 1 |
| Tabla 3 - Persona Involucrado..... | 3 |
| Tabla 4 - Persona Involucrado..... | 3 |
| Tabla 5 - Persona Involucrado..... | 3 |
| Tabla 6 - Persona Involucrado..... | 3 |
| Tabla 7 - Referencia | 5 |
| Tabla 8 - Característica del usuario | 8 |
| Tabla 9 - Característica del usuario | 8 |
| Tabla 10 - Característica del usuario | 8 |

1. (SRS) Ficha del documento

Tabla 1 - Ficha del documento

| Fecha | Revisión | Autor | Verificado dep. Calidad. |
|------------|----------|--|--------------------------|
| 06/01/2017 | | <ul style="list-style-type: none">✓ Joaquín Melquisedec Espinoza.✓ Robinson Akiles Obando Aragón. | |

Documento validado por las partes en fecha: 06/01/2017

Tabla 2 - Documento Validado

| Por el cliente | Por la Universidad |
|-------------------------------|---|
| Academia Bilingue Amerrisque. | Universidad Nacional Autónoma de Nicaragua. FAREM - Chontales |
| | |

2. Introducción

Este documento es una Especificación de Requisitos de Software (ERS) para el Sistema de Información web para la Gestión Académica del Colegio A.B.A. de Juigalpa Chontales, primera versión. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830. 1998.

3. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema de información Web que permitirá gestionar distintos procesos administrativos y académicos. Éste será utilizado por estudiantes, docentes y directivos.

4. Alcance

El SIS-I será un sistema que funcionará en un entorno Web basado en las TIC que permitirá administrar y consultar la información de los estudiantes matriculados y de sus tutores, de igual manera permitirá que el tutor gestione la matrícula y las notas en línea. Esta aplicación apoyará los siguientes procesos:

- Recopilación de datos personales.
- Administración de inscripciones.
- Administrar usuarios del sistema.
- Consultar datos.
- Modificar registros.
- Generar notificaciones.
- Subir Calificaciones.
- Consultar informe de calificaciones según los cortes evaluativos.
- Generar reportes.
- Matricula en línea.
- Generar recibos de pagos en caja.

- Generar informes de ingresos y egresos.

El SIS-I permitirá inscripciones descentralizadas a través de una estación de cómputo dotada con una conexión a internet.

5. Personal involucrado

Tabla 3 - Persona Involucrado

| | |
|-------------------------|--|
| Nombre | Joaquín Melquisedec Espinoza. |
| Rol | Analista, diseñador y programador. |
| Categoría profesional | Ing. En Sistemas de Información. |
| Responsabilidades | Análisis de Información, diseño y programación del SIS-I |
| Información de contacto | jmelkisedek26@gmail.com |
| Aprobación | |

Tabla 4 - Persona Involucrado

| | |
|-------------------------|--|
| Nombre | Robinson Akiles Obando Aragón |
| Rol | Analista, diseñador y programador. |
| Categoría profesional | Ing. En Sistemas de Información. |
| Responsabilidades | Análisis de Información, diseño y programación del SIS-I |
| Información de contacto | robin93obando@gmail.com |
| Aprobación | |

Tabla 5 - Persona Involucrado

| | |
|-------------------------|--|
| Nombre | Miriam Tellez |
| Rol | Docente Tutor |
| Categoría profesional | Master |
| Responsabilidades | Revisar los procesos de diseño y desarrollo de SIS-I |
| Información de contacto | mpatricia@gmail.com |
| Aprobación | |

Tabla 6 - Persona Involucrado

| | |
|-------------------------|--|
| Nombre | Saira Urbina Cienfuegos |
| Rol | Docente Tutor |
| Categoría profesional | Master |
| Responsabilidades | Revisar los procesos de diseño y desarrollo de SIS-I |
| Información de contacto | sairacien5@hotmail.com |
| Aprobación | |

6. Definiciones, acrónimos y abreviaturas

- **Usuario:** Persona que usará el sistema para gestionar procesos.
- **SIS-I:** Sistema de información Web para la gestión Académica.
- **ERS:** Especificación de Requisitos de Software.
- **RF:** Requerimiento Funcional.
- **RNF:** Requerimiento no Funcional.
- **FTP:** Protocolo de Transferencia de Archivos.
- **BD:** Base de datos

7. Referencias

Tabla 7 - Referencia

| Referencia | Título | Ruta | Fecha | Autor |
|------------|---------|------|-------|-------|
| 1 | IEEE830 | | 1998 | IEEE |

8. Resumen

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que este debe realizar, los datos y los factores, restricciones, supuestos y dependencias que afectan el desarrollo, sin entrar en excesivos detalles.

Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

9. Descripción general

A continuación se describen las principales funcionalidades del Sistema.

a. Perspectiva del producto

El sistema SIS-I será un producto diseñado para trabajar en entornos Web, lo que permitirá su utilización de forma rápida y eficaz, además permitirá generar reportes por medio de consultas para imprimir documento directamente desde la página web. Trabjará de manera independiente por lo tanto no interactuará con otros sistemas.

10. Funcionalidad del producto.

El SIS-I permitirá las siguientes funciones:

- a) **Identificar usuarios:** El sistema pedirá al usuario identificarse antes de entrar pidiendo llenar un formulario con datos sencillos (Usuarios, Password).
- b) **Administración de Usuarios:** El Administrador del Sistema podrá gestionar los usuarios (Agregar, modificar, eliminar, buscar).
- c) **Registrar Usuarios:** El usuario de Tipo Administrador se encargará de registrar los datos personales de los usuarios (Nombre, Apellidos, Cédula, Dirección, teléfono, parentesco, Nacionalidad,).
- d) **Asignar Usuario y Password:** Seguidamente de registrar los datos del usuario el sistema proporcionará Nombre de Usuario Y Password dependiendo el tipo de Usuario, datos que serán asignados por defecto, los cuales podrá cambiar el usuario al ingresar al sistema si así lo desea.
- e) **Editar Password:** El sistema permitirá a los usuarios cambiar su contraseña cuando lo estimen conveniente para ello pedirá ingresar contraseña actual y una nueva si los datos son válidos entonces se guardaran los cambios en la base de datos.
- f) **Modificar Datos personales:** El sistema permitirá a cada usuario realizar una edición de sus datos personales Editando el formulario que contiene: Nombre, Apellido, Cedula de identidad, numero de Carnet, Fecha de Nacimiento, Dirección, teléfono, Nacionalidad.
- g) **Eliminar Usuarios:** El usuario de tipo Administrador podrá dar de baja o eliminar a un usuario que ya no seguirá haciendo uso del sistema. Buscando usuarios por nombres o por Numero de Cedula aparecerá próximo a sus datos Las opciones Modificar, Dar de Baja.
- h) **Llenar recibo de pago:** El usuario Administrador accediendo al módulo de caja realiza el llenado del recibo escogiendo que tipo de pago se realiza.
- i) **Imprimir recibo:** El sistema permite la impresión del recibo de caja después de su llenado.

- j) **Imprimir informes:** El sistema muestra reportes de entradas y salidas (ingresos y egresos), dicho reporte será generado en formato PDF y se podrá imprimir directamente.
- k) **Gestión de Clientes:** Para realizar los pagos el sistema buscará a los clientes por nombre de sus estudiantes, los cuales están almacenados en la base de datos.
- l) **Matricular:** Proceso que permitirá a una persona matricular al estudiante, esto incluye: Nombre de Estudiante, Nombre del tutor, Grado que será matriculado, Fecha de Reservación (Generada por el Sistema).
- m) **Ver estado de Matrícula:** El usuario de tipo Tutor podrá ver el número de estudiantes que tiene a cargo y su estado de matrícula, y el sistema le permitirá la opción de: Matricular, Cancelar matrícula.
- n) **Ver calificaciones:** El usuario “Tutor” podrá tener acceso a sus estudiantes a cargo para los cuales podrá consultar sus calificaciones a través de un informe que mostrará las notas por cada corte evaluativo y la nota final.
- o) **Ver Notificaciones:** El usuario tipo “Tutor” recibirá notificaciones enviadas por los usuarios “docentes” y “Administrador” con respecto a las actividades que se estarán realizando en el centro de estudio de forma general y específica de acuerdo a cada grado.
- p) **Subir calificaciones:** El usuario tipo “Docente” deberá subir las calificaciones de sus estudiantes de acuerdo al grado que imparte en fechas de cortes evaluativos.
- q) **Enviar Notificaciones:** El Docente o Administrador deberá crear un nuevo evento, este deberá contener toda la información sobre la actividad que se realizará, este evento será enviado como notificación al tutor.
- r) **Navegación:** Procesos por el cual una persona podrá utilizar el SIS-I.

11. Características de los usuarios

Tabla 8 - Característica del usuario

| | |
|-----------------|---|
| Tipo de usuario | Administrador |
| Formación | Técnico en Computación |
| Habilidades | Manejo de Ventanas de Windows, familiarización con navegadores de internet. |
| Actividades | Control y manejo del Sistema en General |

Tabla 9 - Característica del usuario

| | |
|-----------------|--|
| Tipo de usuario | Tutor |
| Formación | NA |
| Habilidades | Manejo de ventanas de Windows. Experiencia con uso de navegadores de internet. |
| Actividades | Llenado de formularios de datos personales. |

Tabla 10 - Característica del usuario

| | |
|-----------------|--|
| Tipo de usuario | Docente |
| Formación | NA |
| Habilidades | Manejo de ventanas de Windows. Experiencia con uso de navegadores de internet. |
| Actividades | Llenado de formularios de notas. |

12. Restricciones

- Interfaz para ser usada con internet.
- El servidor debe ser capaz de atender consultas concurrentes.
- El sistema se diseñará según un modelo de arquitectura cliente/Servidor.
- El sistema deberá tener un diseño e implementación sencilla, independientemente de la plataforma o del lenguaje de programación.
- Todo material que se realice para el usuario y la aplicación debe estar en idioma Español.

13. Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables.
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos antes indicados para garantizar una ejecución correcta de la misma.
- Se usaran Los Lenguajes y tecnología siguiente: Visual Studio.Net 2012; SQL Server 2012.
- Se usará protocolos de comunicación TCP/IP y HTTP.
- Debe realizarse una capacitación adecuada y acorde a lo que cada usuario va a realizar. Su capacitación de hará en el momento que sea necesaria y a las persona indicada.

14. Evolución previsible del Sistema

El sistema tendrá que mostrar los datos del estudiante y el tutor y de otros administradores a los usuarios de tipo administrador, de igual manera deberá permitir la modificación dentro de la misma consulta. MEJORAS FUTURAS

15. Requisitos específicos

RE001: Aporte de socios: En el módulo de caja deberá haber un formulario llamado: “Aporte de Socios” el cual se llena ocasionalmente.

16. Interfaces de usuario

RI001: Vínculos de Navegación: El sistema debe presentar o contener vínculos de navegación entendibles.

RI002: Imagen Empresarial: El sistema debe en cada una de las páginas contener la imagen del logo de la Academia.

17. Interfaces de hardware

RH001: Conexión de impresora: La máquina del administrador deberá tener conectada una impresora con papel disponible para la generación de recibos de caja e informes.

18. Interfaces de software

- **RFIS01:** Sistema Operativo, windows XP o superior
- **RFIS02:** Explorador Mozilla o Chrome

19. Interfaces de comunicación

Los servidores y aplicaciones se comunicaran entre sí, mediante protocolos de red ya sea web o local, siempre que sea posible.

20. Requisitos funcionales

RF001: Apertura a la aplicación: El Sistema debe permitir ingresar por medio de la cabecera HTTP.

RF002: Autenticación de usuarios: Se pedirá Usuario y Password, las cuales serán portadas por los usuarios que anteriormente habrán sido registrados al sistema por el administrador. De igual manera en este paso se identificará de manera interna que tipo de usuario accede al sistema y que tipo de acciones puede realizar en dependencia de su nivel de accesibilidad.

RF003: Registro de Nuevo Tutor: El Usuario de Tipo Administrador registra los datos del tutor: Nombre, Apellido, Cédula de identidad, Parentesco, Dirección domiciliar, dirección de trabajo, teléfono personal, teléfono del trabajo, Nombre de usuario, contraseña.

RF004: Registro de Nuevo Docente: El administrador podrá registrar los datos del nuevo usuario tipo “Docente” de la misma manera que al tutor, pidiendo los siguientes datos: Nombre, Apellido, Cédula de identidad, Fecha de Nacimiento, teléfono, dirección, Usuario, Contraseña.

RF005: Asignar Usuario y Password: Al momento de registrar los datos de los tutores o profesores también serán registrados los datos de usuario quedando asignados los tipos de usuario en dependencia de a quien se está registrando en ese momento.

RF006: Actualización Password de Usuarios: El sistema debe permitir a los usuarios actualizar su Password si así lo desea.

RF007: Modificar datos personales: El sistema permitirá al usuario administrador la edición de los datos personales de los usuarios o entidades registradas y almacenadas en el sistema. El usuario busca a quienes desea consultar información de sus datos generales y el sistema lista en una tabla los nombres con la opción de “Editar” luego de la edición las opciones que muestra el sistema son “Cancelar” y “Guardar”.

RF008: Modificar datos de Usuarios: El sistema debe permitir a cada usuario editar sus datos de Usuario los cuales son “Usuario” y “Contraseña”. También el administrador puede cambiar los datos de cualquier usuario con la opción de “Editar” la cual también habilita la edición de los datos de usuario y la demás información personal.

RF009: Habilitar o Deshabilitar Usuarios: El sistema debe permitir al usuario Administrador dar de baja o eliminar a los usuarios que ya no seguirán haciendo uso del sistema, para esto se debe entrar a la opción Buscar de la Barra de Menús y elegir el tipo de usuario a buscar luego seleccionar de la tabla el nombre del usuario y junto a sus datos aparecerá la opción de Modificar y Eliminar.

RF010: Matricular: El sistema deberá permitir realizar la matrícula, mostrando al tutor el estado en el que se encuentra su estudiante que tiene a cargo, de esta manera el tutor podrá Matricular al estudiante con solamente hacer clic en un botón el cual verifica si el estudiante está o no matriculado en el año lectivo en curso.

RF011: Cancelar Matrícula: El Usuario de tipo Tutor tendrá la opción de ver el estado de su estudiante “Matriculado” “Sin matricular” al iniciar sesión en el sistema y podrá Eliminar la matricula presionando el botón Cancelar Matricula.

RF012: Listar registros: El sistema deberá generar el listado de la consulta que se haga por medio de una vista de impresión.

RF013: Llenar recibo de caja: El Usuario Administrador tendrá un formulario el cual reflejara el nombre del estudiante, tipo de pago, (Si es Mensualidad también el Mes a pagar), cantidad, Observaciones.

RF014: Imprimir Recibo: Al terminar de llenar el recibo habrá un botón para cancelar, y otro para Imprimir, este último enviará a imprimir el recibo de forma inmediata.

RF015: Anulación de recibo: Tras haber un error, cambio de parecer o algún tipo de problema ajeno, el sistema deberá permitir al administrador la anulación de un recibo.

RF0016: Imprimir Informes de ingresos e informes de egresos: Todos los días 7 de cada mes el sistema deberá detallar en un informe las entradas, informe de gastos e informe de clientes con mora.

RF0017: Imprimir hoja de matrícula: El sistema debe permitir la impresión de la hoja de matrícula del estudiante o en su defecto ser descargado y almacenado en formato PDF.

RF0018: Consultar calificaciones: El Tutor tiene la opción de consultar calificaciones en línea de cada estudiante para ello habrá un botón con la opción “Ver” que lleva al usuario a una pantalla que muestra una hoja de calificaciones con las notas del estudiante para cada asignatura y cada corte evaluativo.

RF0019: Imprimir hoja de calificaciones: Después de consultar las calificaciones en línea de cada estudiante el sistema debe permitir que el usuario descargue en formato PDF su hoja de calificaciones y allí mismo tenga la opción de imprimir como una especie de boletín escolar.

RF0020: Asignar grados a los docentes: Después de haber registrado a los usuarios “Docentes” que van a interactuar con el sistema, se debe hacer una asignación de grados a cada docente en donde se asignen las asignaturas a cada grado y uno o más grados a un docente específico.

RF0021: Listar estudiantes de cada grado: Al Docente se le muestran en la pantalla principal de su cuenta de usuario una lista con los grados que le fueron asignados con la opción de “Listar” que genera una lista con los estudiantes de ese grado.

RF0022: Imprimir lista de estudiantes por grado: El sistema permite al Docente imprimir una esquila con la lista de los estudiantes de un grado.

RF0023: Calificar: A la par del nombre de cada estudiante del grado que le haya sido asignado a ese Docente aparecerá la opción de “Calificar” la cual permite al Docente escribir las calificaciones individuales de sus estudiantes por cada materia o área de evaluación en dependencia de los cortes evaluativos que se evalúan en un periodo determinado de tiempo y posteriormente aparece la opción “Guardar” que permite que el sistema actualice las notas de los estudiantes y que el usuario “Tutor” pueda consultar las calificaciones de sus estudiantes desde su cuenta.

RF0024: Registro de Socios: Los socios hacen aportes a la institución los cuales cuentan como ingresos que debe de manejar el sistema, para este tipo de ingreso se crea una tabla de socios en donde se registran los datos generales de los asociados pidiendo datos como: Nombres y Apellidos, dirección, teléfono.

21. Requisitos no funcionales

RNF001 Cantidad de Usuarios Concurrentes: El número máximo dependerá de la capacidad del servidor. El sistema debe soportar un número de usuarios de acuerdo a los recursos de infraestructura.

RNF002 Cantidad de información almacenada: El número máximo de información a registrar dependerá de la capacidad del servidor. El sistema debe soportar un número de usuarios de acuerdo a los recursos de infraestructura.

RNF003 Base de Datos: Las consultas que permiten la interacción de los scripts con la base de datos deben permitir interactuar con el motor de base de datos.

RNF004 Sistema Operativo: El Sistema debe permitir ser instalado en el Sistema Operativo Windows, y los clientes podrán ejecutar y correr el sistema correctamente en Versiones De Windows 7 en Adelante.

RNF006 Herramientas para Diagramas: Se utilizarán Diagramas UML para graficar los casos de uso, diagramas de actividades, etc.

RNF007 Seguridad de Password: El Sistema debe permitir encriptar los Password para mayor seguridad.

RNF008 Disponibilidad del Sistema: El Sistema deberá estar disponible a menos que sucedan causas externas, como fallas en el flujo eléctrico.

Manual Técnico

Contenido

| | |
|--|----|
| 1. Introducción | 1 |
| 2. Objetivo:..... | 1 |
| 3. Requerimientos técnicos..... | 1 |
| 3.1 Requerimientos mínimos de hardware | 1 |
| 3.2 Requerimientos mínimos de software | 1 |
| 4. Herramientas utilizadas para el desarrollo..... | 2 |
| 4.1 ASP | 2 |
| 4.2 SQL Server..... | 2 |
| 5. Contenido:..... | 3 |
| 5.1 Diseño de la aplicación..... | 3 |
| 5.2 Desarrollo de la Aplicación | 9 |
| 5.2.1 Programación del lado del Servidor | 9 |
| 5.2.2 Servicio: | 10 |
| 5.2.3 Bases de datos | 14 |
| 5.2.3.1 Administración de las bases de datos..... | 15 |
| 5.2.4 Cliente: | 15 |

Índice de ilustraciones

| | |
|--|----|
| Ilustración 1 - Programación de página principal..... | 3 |
| Ilustración 2 - Interfaz pantalla de inicio..... | 4 |
| Ilustración 3 - Interfaz de menú Actividades..... | 4 |
| Ilustración 4 - Programación del diseño..... | 5 |
| Ilustración 5 - Interfaz de usuario administrador..... | 6 |
| Ilustración 6 - Programación de las interfaz usando bootstrap..... | 6 |
| Ilustración 7 - Código ASP..... | 7 |
| Ilustración 8 - Código para la conexión..... | 8 |
| Ilustración 9 - Código para cargar los datos..... | 8 |
| Ilustración 10 - Servicios del proyecto..... | 10 |
| Ilustración 11 – Captura de datos..... | 11 |
| Ilustración 12 - ER Capa de datos..... | 11 |
| Ilustración 13 – Estructura de la clase “Servicios asmx”..... | 12 |
| Ilustración 14 - Código de los WebMethod..... | 13 |
| Ilustración 15 – Código de capa de comunicación..... | 14 |
| Ilustración 16 - Elemento de la capa de presentación..... | 15 |
| Ilustración 17 - Código de la página maestra..... | 16 |
| Ilustración 18 - Reutilización de página maestra..... | 17 |
| Ilustración 19 - Referencia con el servicio..... | 19 |
| Ilustración 20 - Código cargar los datos..... | 20 |

1. Introducción

La finalidad de todo manual técnico es la de proporcionar al lector la lógica con la que se ha desarrollado una aplicación, la cual se sabe que es propia de cada programador; por lo que se considera necesario ser documentada.

Aclarando que este manual no pretende ser un curso de aprendizaje de cada una de las herramientas empleadas para el desarrollo del sitio, sino documentar su aplicación en el desarrollo del sitio. Para un mayor detalle acerca de cada una de las herramientas utilizadas, y su forma de operación y aplicación, se recomienda consultar los manuales respectivos de cada una de ellos.

2. Objetivo:

Proporcionar una guía para el lector, del desarrollo de la interfaz y programación del Sistema de Gestión Académica del colegio ABA de la ciudad de Juigalpa Chontales.

3. Requerimientos técnicos

3.1 Requerimientos mínimos de hardware

- Procesador : 1.8 GHz
- Memoria RAM : 2 Gigabytes (GB)
- Disco Duro : 500 GB
- Impresora

3.2 Requerimientos mínimos de software

- Conexión a internet desde cualquier navegador
- Cualquier sistema operativo Windows desde XP

4. Herramientas utilizadas para el desarrollo

4.1 ASP

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia, etc.

4.2 SQL Server

SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) de Microsoft que está diseñado para el entorno empresarial. SQL Server se ejecuta en T-SQL (Transact - SQL), un conjunto de extensiones de programación de Sybase y Microsoft que añaden varias características a SQL estándar, incluyendo control de transacciones, excepción y manejo de errores, procesamiento fila, así como variables declaradas.

- Soporte de transacciones.
- Soporta procedimientos almacenados.
- Incluye también un entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.

5. Contenido:

El contenido de este manual se divide en base a los requerimientos de este proyecto de software.

5.1 Diseño de la aplicación

El Diseño de páginas e interfaces de usuario, se ha hecho utilizando las siguientes herramientas

- Software Utilizado: VisualStudio.Net 2012 y SQL Server 2012
- Lenguajes de diseño: HTML5 y CSS3
- Lenguaje de programación: C# ASP.NET

Ejemplo: página principal

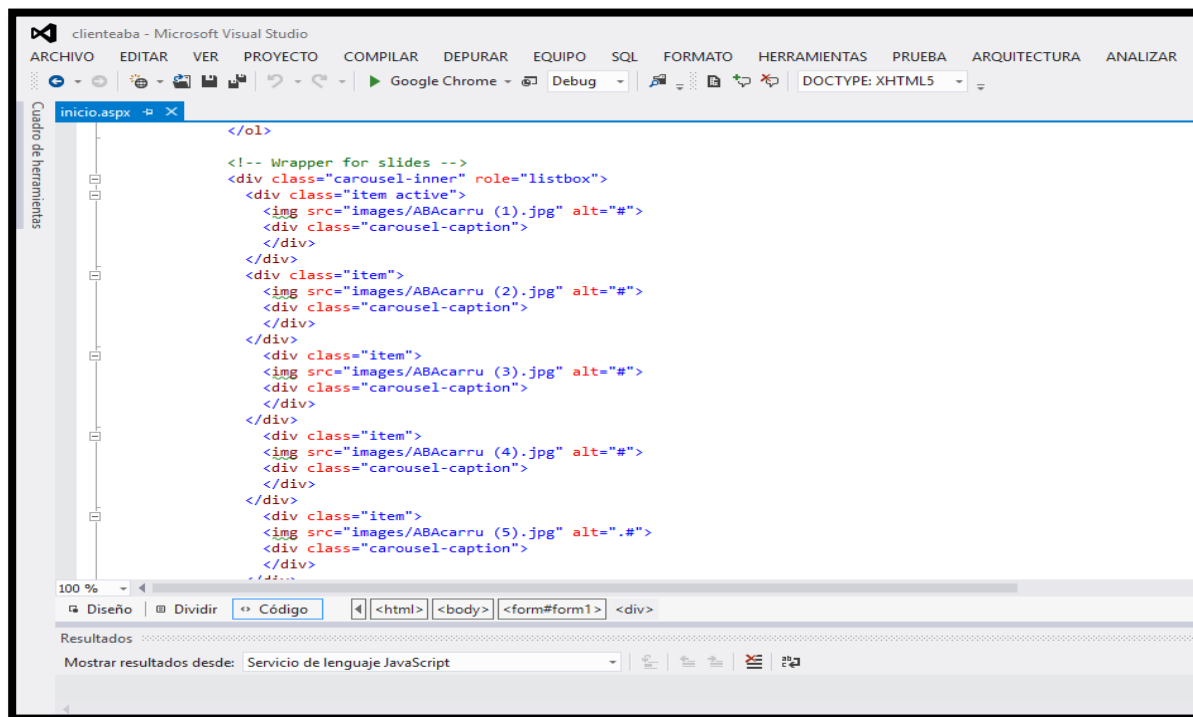


Ilustración 1 - Programación de página principal

Páginas web

Una página web se define como un documento electrónico el cual contiene información textual, visual y/o sonora que se encuentra alojado en un servidor y puede ser accesible mediante el uso de navegadores. Una página web forma parte de una colección de otras

páginas webs dando lugar al denominado sitio web el cual se encuentra identificado bajo el nombre de un dominio.

Ejemplo: interfaz de usuario

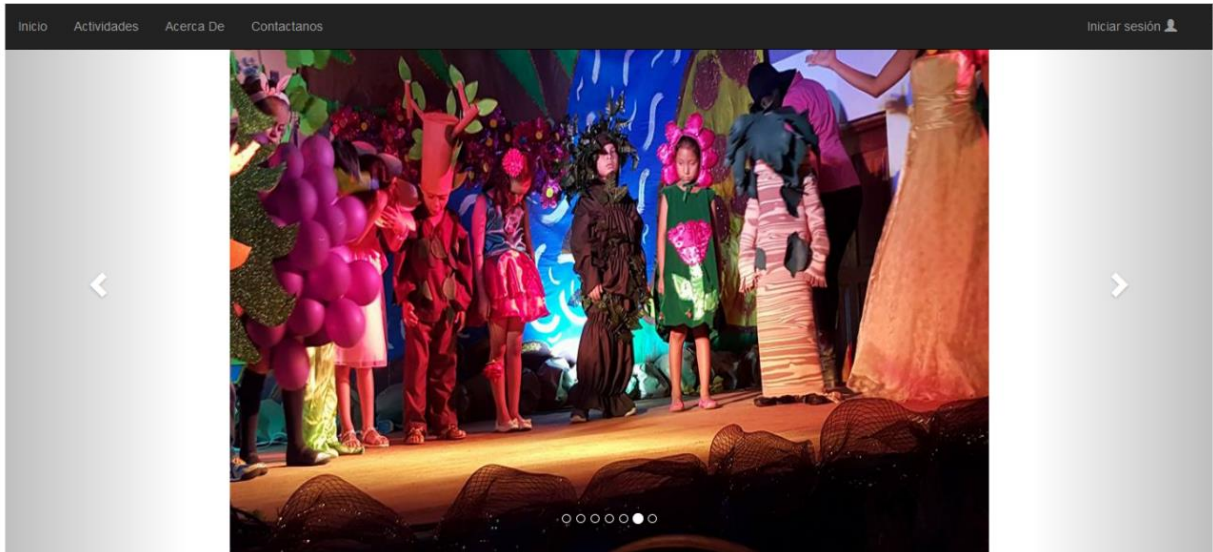


Ilustración 2 - Interfaz pantalla de inicio

Utilizando los lenguajes HTML y CSS con apoyo de dos Frameworks Bootstrap y Materialize basados en el mismo CSS han sido creados diseños agradables e interactivos.



Ilustración 3 - Interfaz de menú Actividades

Ejemplo de implementación de diseños:

```
<body>
  <form id="form1" runat="server">
    <div>
      <nav style="position: fixed; top:0; width:100%" class="navbar navbar-inverse navbar-static-top marginBottom-0" role="navigation">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbar-collapse-1">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
        </div>
        <div class="collapse navbar-collapse" id="navbar-collapse-1">
          <ul class="nav navbar-nav navbar-left">
            <li class="dropdown"><a href="#form1" >Inicio</a>
            </li>
            <li><a href="#pnActividades" >Actividades</a>
            </li>
            <li><a href="#pnAcerca" >Acerca De</a>
            </li>
            <li><a href="#pnContactanos" >Contactanos</a>
            </li>
          </ul>
          <ul class="nav navbar-nav navbar-right">
            <li style="margin-right:40px;" class="dropdown"><a href="cesion.aspx" >Iniciar sesión <span class="glyphicon glyphicon-user" aria-hidden="true"></span></a>
          </ul>
        </div><!-- /.navbar-collapse -->
      </nav>
    </div>
  </form>
</body>
```

Ilustración 4 - Programación del diseño

Parte del código HTML utilizando clases de Materialize, se puede apreciar la estructura principal de documento, y dentro de las diferentes etiquetas, los contenidos de clases que controlan el diseño de la web.

Este apartado de la página de inicio es específicamente el menú superior, este diseño es igualmente basado en el Framework Materialize.

Para comprender el funcionamiento de este framework puede visitar: <http://materializecss.com/>



Ilustración 5 - Interfaz de usuario administrador

Pantalla del administrador del sistema basada en framework Bootstrap 3.

```

<meta http-equiv="content-type" content="text/html; charset=utf-8"/>
<title>Pagina Principal</title>
<link href="menu.css" rel="stylesheet" />
<!-- Latest compiled and minified CSS -->
<link href="css/bootstrap.min.css" rel="stylesheet" />
<!-- Latest compiled and minified JavaScript -->
</head>
<body style="background:#d0c7c7;">
<form id="form1" runat="server">
<div>
<nav class="navbar navbar-inverse navbar-static-top marginBottom-0" role="navigation">
<div class="navbar-header">
<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#navbar-collapse-1">
<span class="sr-only">Toggle navigation</span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
<span class="icon-bar"></span>
</button>
</div>
<div class="collapse navbar-collapse" id="navbar-collapse-1">
<ul class="nav navbar-nav">
<li id="menuMat" runat="server" class="dropdown"><a href="#" class="dropdown-toggle" data-toggle="dropdown">Matricula<b class="caret" ></b></a>
<ul class="dropdown-menu">
<li><a href="tipoMatricula.aspx">Nueva</a></li>
<li><a href="busquedaEstudiantesMat.aspx">Buscar</a></li>
<li><a href="">Lista Estudiante Grado</a></li>
</ul>
</li>
<li id="menuReport" runat="server" class="dropdown"><a href="#" class="dropdown-toggle" data-toggle="dropdown">Socios<b class="caret" ></b></a>
<ul class="dropdown-menu">
<li><a href="crearSocio.aspx">Nuevo</a></li>
</ul>
</li>
<li id="menuCaja" runat="server" class="dropdown"><a href="#" class="dropdown-toggle" data-toggle="dropdown">Caja<b class="caret" ></b></a>
<ul class="dropdown-menu">

```

Ilustración 6 - Programación de las interfaz usando bootstrap

El presente código muestra cómo se utiliza el framework dentro de la aplicación, es muy importante recalcar que los dos frameworks utilizados no son implementados en conjunto en cada página, esto para evitar errores en las clases similares.

Para conocer más sobre este framework puede visitar: <http://getbootstrap.com>

5.1.1 Programación

Como es mencionado anteriormente, la web está controlada mediante códigos de lenguaje ASP los cuales permiten que las páginas sean dinámicas y permite el control de los datos del sitio.

Ejemplo de códigos ASP

```
<%@ Page Title="" Language="C#" MasterPageFile="~/masterPage.Master" AutoEventWireup="true" CodeBehind="boletinEstudiante.aspx.cs"
<asp:Content ID="Content1" ContentPlaceHolderID="Contenido1" runat="server">
    <div class="container container-fluid">
        <div class="row">
            <div class="col-md-1">
                </div>
            <div class="col-md-10">
                <div class="panel panel-primary">
                    <div class="panel-body">
                        <asp:Label runat="Server" ID="lbNombreEst" Text="Nombres del estudiante"></asp:Label>
                        <br />
                        <br />
                        <div class="row text-center">
                            <asp:Table ID="tblcalificaciones" CssClass="table" runat="server">
                                <asp:TableHeaderRow>
                                    <asp:TableHeaderCell>Asinatura</asp:TableHeaderCell>
                                    <asp:TableHeaderCell >1er C.</asp:TableHeaderCell>
                                    <asp:TableHeaderCell >2do C.</asp:TableHeaderCell>
                                    <asp:TableHeaderCell >1er S.</asp:TableHeaderCell>
                                    <asp:TableHeaderCell >3er C.</asp:TableHeaderCell>
                                    <asp:TableHeaderCell >4to C.</asp:TableHeaderCell>
                                    <asp:TableHeaderCell >2do S.</asp:TableHeaderCell>
                                    <asp:TableHeaderCell >NF</asp:TableHeaderCell>
                                </asp:TableHeaderRow>
                                </asp:Table>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</asp:Content>
```

Ilustración 7 - Código ASP

Este código se encarga de conectar la aplicación con la base de datos utilizada, este debe ser incluido en cada página en la que se desee controlar de alguna forma datos, Para utilizar este archivo de conexión, es necesario incluir en cada página el siguiente código:


```
<%@ Page Title="" Language="C#" MasterPageFile="~/masterPage.Master" AutoEventWireup="true" CodeBehind="sinMatricula.aspx.cs" Inherits="clienteaba.sinMatricula" %>
<asp:Content ID="Content1" ContentPlaceHolderID="Contenido1" runat="server">
    <div class="alert alert-danger text-center">
        <strong>Atención!</strong> No se encontraron registros para la Matricula deseada.
    </div>
</asp:Content>
```

Ilustración 8 - Código para la conexión

Esto permite que la conexión con los datos este siempre disponible.

Es necesario recalcar que los datos del sitio web están alojados en el sistema gestor de bases de datos SQL Server, un ejemplo de cómo es posible obtener información de esta BD es:

```
[WebMethod]
public estudiante cargarDatosEstudiante(int codestudiante)
{
    try
    {
        bd = new dataconexDataContext();
        var databd = bd.cargarDatosEstudiante(codestudiante);
        estudiante e = new estudiante();

        foreach(var row in databd)
        {
            e.cod_est = row.cod_est;
            e.cod_tutor = row.cod_tutor;
            e.nombre1 = row.nombre1;
            e.nombre2 = row.nombre2;
            e.apellido1 = row.apellido1;
            e.apellido2 = row.apellido2;
            e.ultimogrado = row.ultimogrado;
            e.fecha_nac = row.fecha_nac;
            e.lugar_nac = row.lugar_nac;
            e.sexo = row.sexo;
            e.direccion = row.direccion;
            e.telefono = row.telefono;
            e.celular = row.celular;
        }

        return e;
    }
    catch
    {
        return null;
    }
}
```

Ilustración 9 - Código para cargar los datos

En esta se obtiene los datos de los estudiantes.

Para instruirse en el uso de este lenguaje se recomienda visitar: <https://www.asp.net/>

5.2 Desarrollo de la Aplicación

La aplicación presentada ha sido desarrollada utilizando HTML y CSS para su diseño, ASP para programación y SQL Server para control de los datos internos de la aplicación.

5.2.1 Programación del lado del Servidor

Como su nombre lo indica el procesamiento del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente, las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente, el cliente solamente recibe una página con el código HTML resultante de la ejecución en el servidor, como la página resultante contiene únicamente código HTML, es compatible con casi todos los navegadores. Existen diferentes tecnologías que se utilizan para llevarlo a cabo, entre las que mencionaremos los CGI's, ASP .NET, PHP.

ASP .NET: al hacer la petición, el servidor se percata que la página requerida es ASP.NET (ASPX); la página, que es tratada como un programa, primeramente es procesada por un analizador de código (Parser), que verifica que el programa está bien escrito y que no posee errores de sintaxis o de referencias perdidas (uso de variables, procedimientos, o miembros inexistentes), si el programa no tiene errores de sintaxis o de referencias perdidas, el programa es puesto a disposición del compilador de páginas de .NET Framework; el compilador se encargará de generar un ensamblado de extensión DLL, de nombre único, que define una clase capaz de generar el código HTML que ha de ser retornado al navegador del cliente, este ensamblado es almacenado en un área denominada caché de ensamblados (Assembly Caché), en donde se almacenan todos los ensamblados generados por el compilador de páginas, ya que está generado el ensamblado capaz de producir el contenido web que ha de ser retornado al cliente, un elemento denominado HTTP Runtime se encarga de solicitar la ejecución de la clase generada, creando la instancia del objeto generador, produciendo entonces el contenido web, el contenido web generado por la clase es almacenado en un área denominada caché de salida (Output Caché); en ese espacio se almacenan los contenidos HTML, los scripts de ejecución del

lado del cliente, e incluso los datos generados como parte de la salida, de ahí el contenido es enviado al cliente solicitante, si posteriormente se solicita la misma página, se buscará primeramente en Output Caché; si el contenido web se encuentra todavía ahí, no se realizará procesamiento alguno, por lo cual la respuesta es extremadamente rápida.

5.2.2 Servicio:

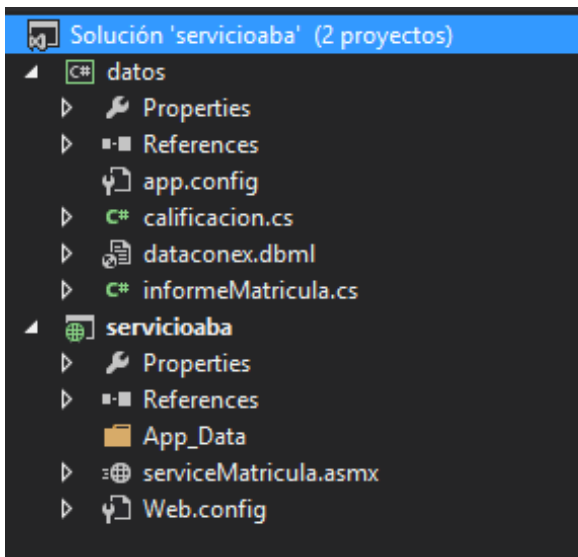


Ilustración 10 - Servicios del proyecto

La estructura del servidor, contiene dos proyectos, “datos” la capa de conexión a la base de datos y “servicioaba” la capa de comunicación con el cliente, la capa “datos” posee 2 tipos de elementos, las clases “.cs” que representan a los objetos o propiedades que no existen en la base de datos, los objetos que si existen en la base de datos son obtenidos y manipulados mediante el otro elemento presente en esta capa “dataconex.dbml”, esta unidad posee una conexión a la base de datos mediante la tecnología LinQ, la cual permite conectar, ver y utilizar los objetos que estén presentes en la base de datos, normalmente tablas, procedimientos almacenados y funciones, por otra parte, la capa de “servicioaba” presenta un solo elemento, siendo “serviceMatricula.asmx” el servicio que proporciona información específica según la solicitud que un cliente realice.

```

namespace datos
{
    5referencias
    public class calificaciones
    {
        string nombre; //Variable privada tipo texto
        int note, cort; //Variables privadas tipo numericas

        1referencia
        public string Nombre //Propiedad, Devuelve o asigna un data a la variable "nombre"
        {
            get { return nombre; }
            set { nombre = value; }
        }

        1referencia
        public int Corte //Propiedad, Devuelve o asigna un dato a la variable "cort"
        {
            get { return cort; }
            set { cort = value; }
        }

        1referencia
        public int Note //Propiedad, Devuelve o asigna un dato a la variable "note"
        {
            get { return note; }
            set { note = value; }
        }
    }
}

```

Ilustración 11 – Captura de datos

Capa de datos: Ejemplo del uso de propiedades para captura de datos de la base de datos, estos permiten crear tipos de datos personalizados para crear uno o varios objetos de tipo calificación la cual contendrá en este caso 3 tipos de datos.

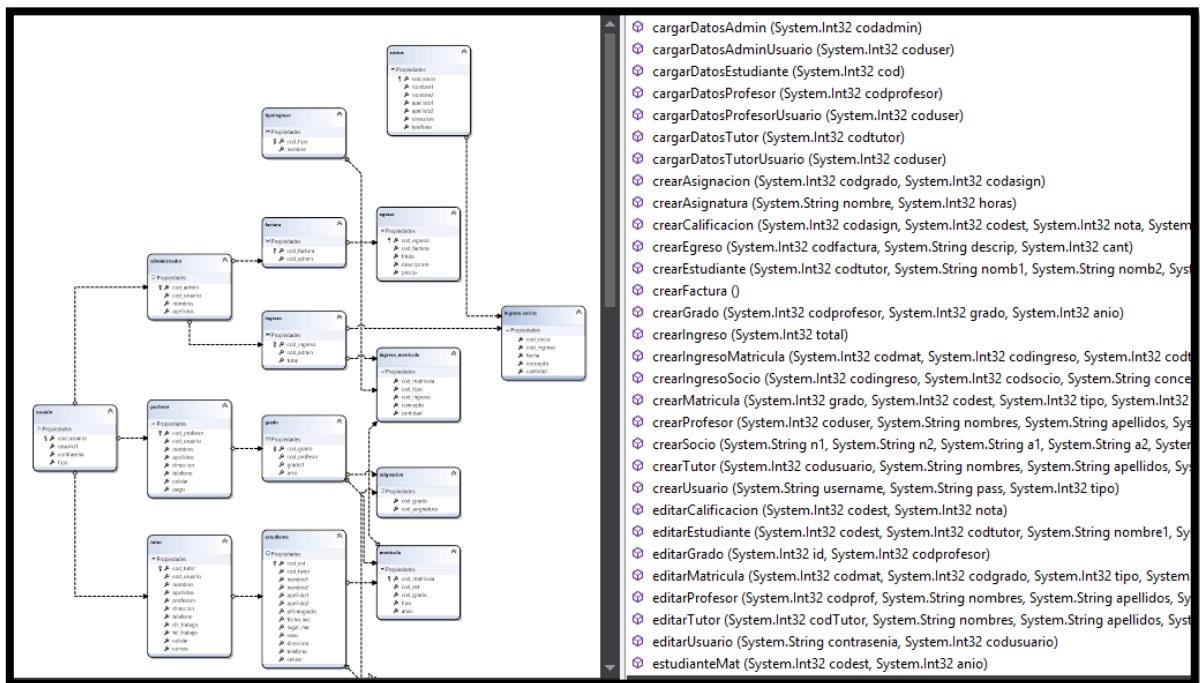


Ilustración 12 - ER Capa de datos

Capa datos: Ejemplo del uso del elemento “dataconex.dbml”, luego de la conexión a la base de datos, esta permite obtener directamente las tablas con sus respectivas conexión, las cuales se puede utilizar en la programación como objetos para darle cuerpo a la información, además podemos ejecutar directamente del código cualquiera de los procedimientos o funciones listadas en esta clase, todos estos elementos deben ser añadidos desde la base de datos con el fin de utilizarlos dentro de la programación del proyecto.

```

    [WebMethod]
    public bool crearAsignatura(string nombre, int horas)
    {
        try
        {
            bd = new dataconexDataConect();
            var data = bd.crearAsignatura(nombre, horas);
            return true;
        }
        catch
        {
            return false;
        }
    }

    [WebMethod]
    public bool crearCalificacion(int cursoId, int contest, int nota, int curso, int asid)
    {
        try
        {
            bd = new dataconexDataConect();
            var data = bd.crearCalificacion(cursoId, contest, nota, curso, asid);
            return true;
        }
        catch
        {
            return false;
        }
    }

    [WebMethod]
    public int crearEstudiante(int cursoId, string c1, string c2, string c3, string c4, int ultimoGrado, DateTime fechaNac, string lugarNac, char sexo, string dir, string tel, string celular)
    {
        try
        {
            bd = new dataconexDataConect();
            var data = bd.crearEstudiante(cursoId, c1, c2, c3, c4, ultimoGrado, fechaNac, lugarNac, sexo, dir, tel, celular);
            int ret = 0;
            foreach (var x in data)
            {
                ret = Convert.ToInt32(x.GetValue());
            }
            return ret;
        }
        catch
        {
            return 0;
        }
    }

    [WebMethod]
    public int crearGabo(int cursoId, int grado, int asid)
    {
        try
        {
            bd = new dataconexDataConect();
            var data = bd.crearGabo(cursoId, grado, asid);
            int congrado = 0;
            foreach (var x in data)
            {
                congrado = Convert.ToInt32(x.GetValue());
            }
        }
    }

```

Ilustración 13 – Estructura de la clase “Servicios asmx”

Vista reducida de la estructura que posee la clase “Servicios asmx” de la capa de comunicación, en esta se obtienen y envían al cliente toda la información necesaria, dependiendo de la solicitud que el mismo realice.

El proyecto posee dos tipos de “WebMethods”, los que devuelven un dato concreto y los que devuelven listas de datos, a continuación, la estructura básica de estos es la siguiente:

[WebMethod]
 <Nivel de acceso> <dato de retorno>
 <nombre del método>(<parametros>)

```

{
    Try{
    <código a ejecutar>
    }
    catch{
    <código a ejecutar en caso de error>
    }
}

```

los ejemplos:

```
[WebMethod] //permite que el metodo sea accesible medianta la web
0 referencias
public administrador cargarDatosAdministradorUsuario(int codusuario) //Devuelve un objeto tipo administrador
{
    try // Prueba que las instrucciones de programación para evitar excepciones no controladas
    {
        bd = new dataconexDataContext(); // inicializar la variable bd para la conexión a la base de datos
        administrador a = new administrador(); //Objeto "administrador" toma de la bd las propiedades directamente de la tabla "administrador"
        var data = bd.cargarDatosAdminUsuario(codusuario); //Variable sin formato, se acopla y obtiene los datos recibidos al ejecutar un metodo de la bd

        foreach(var r in data) //Recorre todas las filas obtenidas de la bd y las almacena en una variable(En este caso solo una fila pero se debe recorrer)
        {
            a.cod_admin = r.cod_admin; //Asignar datos a la variable
            a.cod_usuario = r.cod_usuario;
            a.nombres = r.nombres;
            a.apellidos = a.apellidos;
        }

        return a; //Devuelve el objeto tipo "administrador" al cliente para su uso
    }
    catch //En caso de falla al ejecutar las instrucciones se utiliza el código dentro de este
    {
        return null; //Si hay errores de conexión o datos, devuelve un valor nulo
    }
}
```

Ilustración 14 - Código de los WebMethod

Capa de comunicación: Ejemplo de método que devuelve un solo dato, en este, el tipo de dato a enviar como respuesta a una solicitud es “administrador” este tipo de dato en una tabla presente en la base de datos del software y gracias a la tecnología utilizada para la conexión a esta, podemos utilizarlos con solo crear un objeto de tipo “dataconex” e inicializarlo en nuestro método según sea la necesidad, la variable “data” que es tipo “indefinido o sin formato”, se acopla a la información recibida para evitar crear múltiples objetos por cada fila que obtenemos, finalmente el recorrido de la información realizado a través del ciclo “foreach” nos permite conocer cada registro o fila individual obtenida, en este caso aunque es obtenido un solo registro es necesario utilizar el ciclo de recorrido para leer cada registro individual o celda dentro de cada fila.

```

[webMethod]
0 referencias
public List<classInformeMatricula> informeMatricula(int codMatricula)
{
    try
    {
        bd = new dataconexDataContext();
        List<classInformeMatricula> lista = new List<classInformeMatricula>();
        var data = bd.informeMatricula(codMatricula);

        foreach(var r in data)
        {
            classInformeMatricula inf = new classInformeMatricula()
            {
                Nombre1 = r.nombre1,
                Nombre2 = r.nombre2,
                Apellido1 = r.apellido1,
                Apellido2 = r.apellido2,
                Grado = Convert.ToInt32(r.grado),
                Codasig = Convert.ToInt32(r.cod_asignatura),
                Nombreasignatura = r.nombre
            };

            lista.Add(inf);
        }

        return lista;
    }
    catch
    {
        return null;
    }
}

```

Ilustración 15 – Código de capa de comunicación

Capa de comunicación: Ejemplo de método que devuelve un listado de datos de un objeto en específico, las diferencias con un método que devuelve un solo data es que este devuelve un “List<tipo de dato>”, dentro de las instrucciones de programación se crea una variable de este mismo tipo y al hacer el recorrido de los datos se añade a la lista un objeto del tipo definido, en este caso “classInformeMatricula” o “informeMatricula” el cual simplemente es una clase “.cs” almacenada en la capa de conexión a los datos, finalmente, el objeto que retornamos es la variable tipo “Lista” creada.

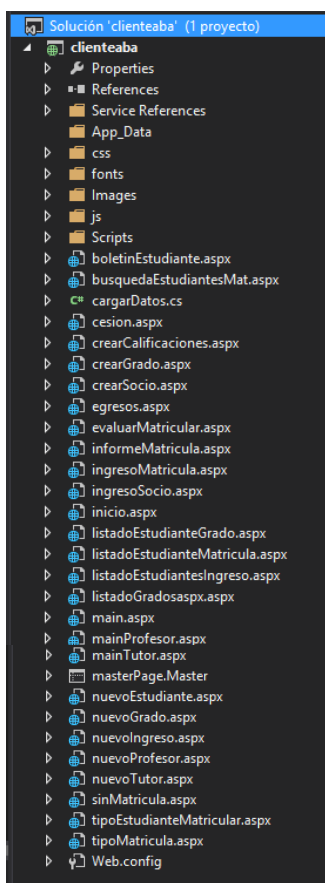
5.2.3 Bases de datos

Una base de datos (cuya abreviatura es *BD*) es una entidad en la cual se pueden almacenar datos de manera estructurada, con la menor redundancia posible. Diferentes programas y diferentes usuarios deben poder utilizar estos datos. Por lo tanto, el concepto de base de datos generalmente está relacionado con el de red ya que se debe poder compartir esta información. De allí el término base. "Sistema de información" es el término general utilizado para la estructura global que incluye todos los mecanismos para compartir datos que se han instalado.

5.2.3.1 Administración de las bases de datos

Rápidamente surgió la necesidad de contar con un sistema de administración para controlar tanto los datos como los usuarios. La administración de bases de datos se realiza con un sistema llamado **DBMS** (Database management system [Sistema de administración de bases de datos]). El DBMS es un conjunto de servicios (aplicaciones de software) para administrar bases de datos.

5.2.4 Cliente:



Estructura del cliente del software esta solución posee un solo proyecto “clienteaba”, siendo este una página web de ASP .NET, posee 3 tipos de elementos, las clases “.cs” utilizadas para obtener, dar formato y cargar elementos, otro elemento son las paginas web de ASP “.aspx” las cuales poseen elementos HTML o AXML(XML de ASP), finalmente se encuentra un solo elemento tipo pagina maestra “.master” la cual es una pagina AXML con contenedores de ASP los cuales cargan las demas páginas dentro de el.

Ilustración 16 - Elemento de la capa de presentación

Uso de “pagina.Master” se da mayormente para crear menus, y cargar las demas paginas dentro de este, en el presente proyecto, este es especificamente el uso que se da.

Uso de “masterPage.Master”:

En ASP .NET lo único necesario para definir una página maestra es el siguiente encabezado:

```
<%@ Master Language="C#" AutoEventWireup="true" CodeBehind="masterPage.master.cs" Inherits="clienteaba.masterPage" %>
```

Seguidamente se utiliza la estructura básica de HTML siendo la única diferencia que al final del elemento “<BODY>” justo antes de la etiqueta “</BODY>” debe de existir uno o más contenedores en los cuales se pueden cargar paginas dependiendo de la navegación del usuario.

Parte del código contenido dentro de “masterPage.Master” ilustra el uso de los contenedores de ASP luego de diseñar un menú de navegación sobre el mismo, esto para cargar bajo el menú principal de la aplicación, cualquier elemento “.aspx” y presentar la información, se deben definir

```
<li id="menuCaja" runat="server" class="dropdown"><a href="#" class="dropdown-toggle" data-toggle="dropdown">Caja <span class="caret" ></b></a>
  <ul class="dropdown-menu">
    <li><a href="listadoEstudiantesIngreso.aspx">Pago Estudiante</a></li>
    <li><a href="ingresoSocio.aspx">Aporte Socio</a></li>
    <li><a href="egresos.aspx">Gasto</a></li>
  </ul>
</li>

</ul>
<ul class="nav navbar-nav navbar-right">
  <li class="dropdown">
    <a href="#" id="nombreUsuario" runat="server" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-haspopup="true" aria-expanded="false">
      <span class="caret"></span></a>
    <ul class="dropdown-menu">
      <li><a href="inicio.aspx">Cerrar Sesión</a></li>
    </ul>
  </li>
</ul>
</div><!-- /.navbar-collapse -->
</nav>
</div>
<h1 style="text-align:center; font-family:fantasy; font-weight:normal;">
Academia Bilingüe Amerisque </h1>
<h1 style="text-align:center; font-family:'Brush Script MT';">
"Learning with us is different"</h1>
<div>
<asp:ContentPlaceHolder ID="Contenido1" runat="server">
</asp:ContentPlaceHolder>
```

Ilustración 17 - Código de la página maestra

Es mandatorio que en una página “.Master” exista al menos un contenedor, este contenedor debe de poseer un ID y la propiedad “runat=<nombre definido para el servidor>” normalmente se utiliza “runat=‘server’” si el nombre del servidor ASP no fue modificado.

Las páginas web de ASP “.aspx” dentro del proyecto serán cargadas dentro del contenedor que se especifique, para cargar una página dentro de un contenedor, es requerido que la cabecera del documento posea el siguiente código:

```
<%@ Page Title="" Language="C#" MasterPageFile="~/masterPage.Master"
AutoEventWireup="true" CodeBehind="mainProfesor.aspx.cs"
Inherits="clienteaba.mainProfesor" %>
```

En donde

- “MasterPageFile=‘<dirección de pagina.Master>’”

Para mostrar la página dentro del contenedor de la página maestra se utiliza la etiqueta “<content></content>”, definiendo los atributos siguientes:

- ID = <ID del contenedor>
- ContentPlaceHolderID = <ID del contenedor de la página maestra>
- runat = <Nombre definido del servidor>

Ejemplo de uso de página ASP “.aspx” dentro de un contenedor tipo “.Master”:

```
<asp:Content ID="Content1" ContentPlaceHolderID="Contenido1" runat="server">
  <div class="container">
    <div class="row">
      <div class="col-md-3"></div>
      <div class="col-md-6">
        <div class="panel panel-primary">
          <div class="panel-heading">
            <h2 class="title"> Listado grados a su cargo </h2>
          </div>
          <div class="panel-body">
            <asp:Table runat="server" CssClass="table table-bordered" ID="listadoEstudiantes">
              <asp:TableHeaderRow>
                <asp:TableHeaderCell CssClass="col-md-2 text-center">Cod</asp:TableHeaderCell>
                <asp:TableHeaderCell CssClass="col-md-6">Grado</asp:TableHeaderCell>
                <asp:TableHeaderCell CssClass="col-md-2 text-center">Año</asp:TableHeaderCell>
                <asp:TableHeaderCell CssClass="col-md-2 text-center">Listar</asp:TableHeaderCell>
              </asp:TableHeaderRow>
            </asp:Table>
          </div>
        </div>
      </div>
    </div>
  </div>
</asp:Content>
```

Ilustración 18 - Reutilización de página maestra

Un factor importante a recalcar y que se observa en esta ilustración es que, cualquier Framework de diseño o código que deba ser cargado para utilizar la página, se debe de implementar en la página “.Master” ya que estos cargan en cada página que contenga.

Las paginas ASP poseen la ventaja de que además de manejar controles puros de HTML y elementos ASP con más características disponibles para programación y la misma flexibilidad y utilidad de los elementos HTML.

Por ejemplo, una tabla de ASP se define de la siguiente forma:

```
<asp:Table runat="server" ID="listadoEstudiantes">
    <asp:TableHeaderRow>
        <asp:TableHeaderCell></asp:TableHeaderCell>
    </asp:TableHeaderRow>
    <asp:TableRow>
        <asp:TableCell></ asp:TableCell >
    <asp:TableRow>
</asp:Table runat="server" ID="listadoEstudiantes">
```

Mientras que una tabla de HTML se define:

```
<table>
    <tr>
        <td></td>
    </tr>
</table>
```

La programación de las páginas de ASP con lenguaje C# se da en el archivo que en el diseñador de la página web definamos, esto en una línea de código vista anteriormente

```
<%@ Page Title="" Language="C#" MasterPageFile="~/masterPage.Master"
AutoEventWireup="true" CodeBehind="mainProfesor.aspx.cs"
Inherits="clienteaba.mainProfesor" %>
```

En el encabezado de cualquier página de ASP se define la propiedad “CodeBehind” el cual indica el nombre del archivo con extensión “.cs” que controlará todo lo respectivo al comportamiento mediante programación de la página en cuestión, de esta manera, podemos proceder a implementar códigos.

Lo primero que necesitamos en nuestro proyecto cliente para interactuar con el servicio es crear una referencia al mismo, nuestro servicio “servicioaba”, esto especificando la dirección del servidor y el servicio que al que se desea acceder, y finalmente debemos obtener una referencia que nos permita conectarnos y obtener información. En nuestro proyecto nuestra referencia se llama “serv”

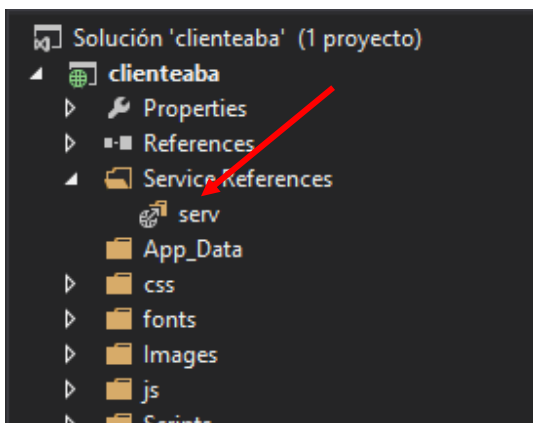


Ilustración 19 - Referencia con el servicio

Para utilizar los métodos dentro de este debemos crear una conexión SOAP al mismo, ejemplo implementado en el proyecto:

```
serv.Service1SoapClient service = new serv.Service1SoapClient();
```

Hecho esto, podemos utilizar todos los [WebMethod] creados en nuestro servicio para obtener y manipular información de la base de datos.

Finalmente, en el proyecto existe una clase única de manejo de información en este caso específicamente para cargar datos, esta clase es llamada “cargarDatos.cs” la cual puede ser referenciada desde cualquiera de las páginas y reutilizar los códigos, esta clase es también un buen ejemplo de cómo obtenemos y manipulamos datos obtenidos del servicio.

```
25 referencias
public class cargarDatos
{
    serv.Service1SoapClient service;

    12 referencias
    public cargarDatos()
    {
        service = new serv.Service1SoapClient();
    }

    1 referencia
    public List<serv.tutor> listadoTutores()
    {
        try
        {
            List<serv.tutor> lista = new List<serv.tutor>();
            var data = service.listarTutores();

            foreach(var r in data)
            {
                serv.tutor t = new serv.tutor()
                {
                    cod_tutor = r.cod_tutor,
                    nombres = r.nombres,
                    apellidos = r.apellidos
                };

                lista.Add(t);
            }

            return lista;
        }
        catch
        {
            return null;
        }
    }
}
```

Ilustración 20 - Código cargar los datos

Manual de Usuario

Contenido

| | |
|--|----|
| 1. Introducción | 1 |
| 2. La pantalla principal e Inicio de Sesión: | 1 |
| 3. Interfaz del usuario Administrador: | 4 |
| 4. Interfaz del usuario Tutor | 11 |
| 5. Interfaz del usuario Docente | 12 |

Índice de Ilustraciones

| | |
|---|----|
| Ilustración 1 - Interfaz página de inicio de la aplicación..... | 1 |
| Ilustración 2 - Interfaz página principal | 2 |
| Ilustración 3 - Interfaz Página Principal | 3 |
| Ilustración 4 - Interfaz Inicio de sesión | 3 |
| Ilustración 5 - Interfaz Usuario Administrador | 4 |
| Ilustración 6 - Menú Socios..... | 4 |
| Ilustración 7 - Menú Caja | 4 |
| Ilustración 8 - Menu Matricula..... | 4 |
| Ilustración 9 - Interfaz Tipo De Matricula..... | 5 |
| Ilustración 10 - Interfaz Listado de Estudiantes | 5 |
| Ilustración 11 - Interfaz Tipo de Matricula | 6 |
| Ilustración 12 - Interfaz Nuevo Estudiante..... | 6 |
| Ilustración 13 - Interfaz Dato del Tutor..... | 7 |
| Ilustración 14 - Interfaz Nuevo Estudiante..... | 7 |
| Ilustración 15 - Interfaz Datos de socio..... | 7 |
| Ilustración 16 - Nuevo Socio | 7 |
| Ilustración 17 - Interfaz Menú de Caja..... | 8 |
| Ilustración 18 - Interfaz Listado de Estudiantes | 8 |
| Ilustración 19 - Interfaz Ingresos..... | 9 |
| Ilustración 20 - Interfaz Tipo de Ingreso Aporte de socio..... | 9 |
| Ilustración 21 - Interfaz Egreso | 10 |
| Ilustración 22 - Interfaz pantalla principal tutor | 11 |
| Ilustración 23 - Interfaz Crear Calificación..... | 12 |
| Ilustración 24 - Interfaz Listado de alumno a cargo..... | 12 |
| Ilustración 25 - Listado de Grado A cargo | 13 |
| Ilustración 26 - Interfaz Crear Calificaciones..... | 13 |
| Ilustración 27 - Interfaz Listas de tutores | 14 |
| Ilustración 28 - Interfaz Modificar datos del tutor | 14 |
| Ilustración 29 - Interfaz Modificar datos del tutor | 14 |
| Ilustración 30 - Listado de estudiante por grado | 14 |

1. Introducción

El presente documento muestra el modo de utilizar el sitio web del colegio ABA, en este se detalla para cada tipo de usuario, suponiendo que poseen conocimientos básicos del uso de las computadoras y el internet, los procesos que puede realizar y como proceder para hacerlo.

2. La pantalla principal e Inicio de Sesión:

La primera pantalla observada al ingresar el sitio es una pantalla informativa la cual posee un menú con el cual navegamos por los diferentes módulos disponibles desde esta.

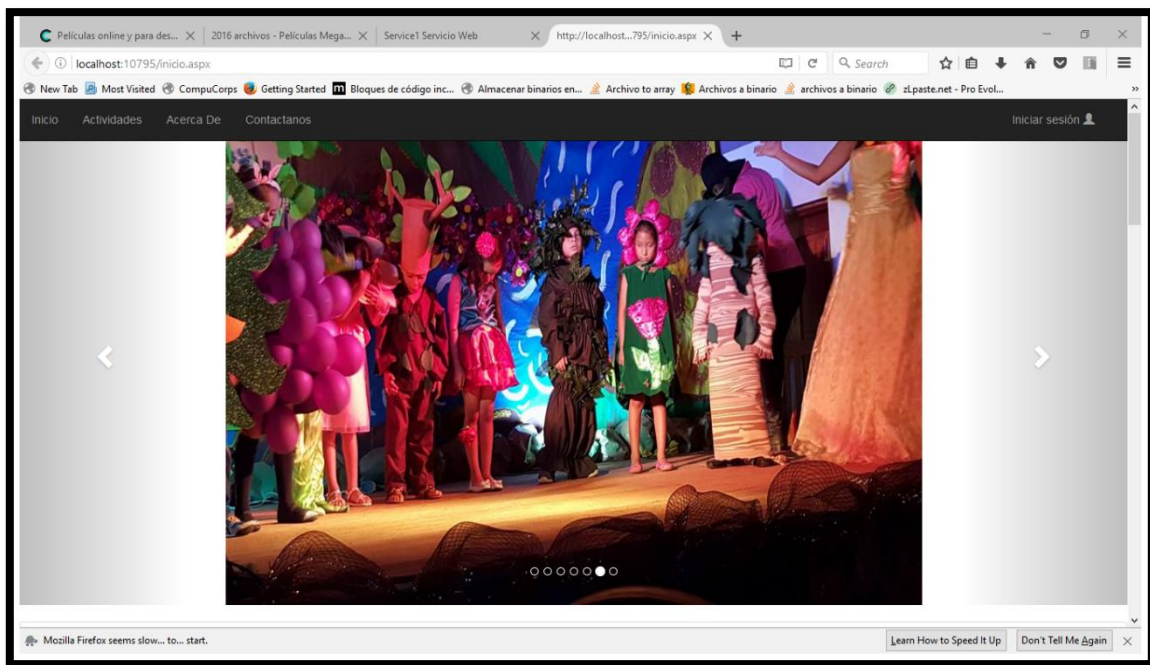


Ilustración 1 - Interfaz página de inicio de la aplicación

En el menú de la pantalla inicial navegamos por la página de dos maneras, seleccionando en el menú a donde ir o utilizando el ratón para navegar de arriba abajo, este caso presenta el desplazamiento hasta el contenido de “Actividades”



Ilustración 2 - Interfaz página principal

Esta pantalla muestra hasta donde nos llevara el menú de “Contáctanos”

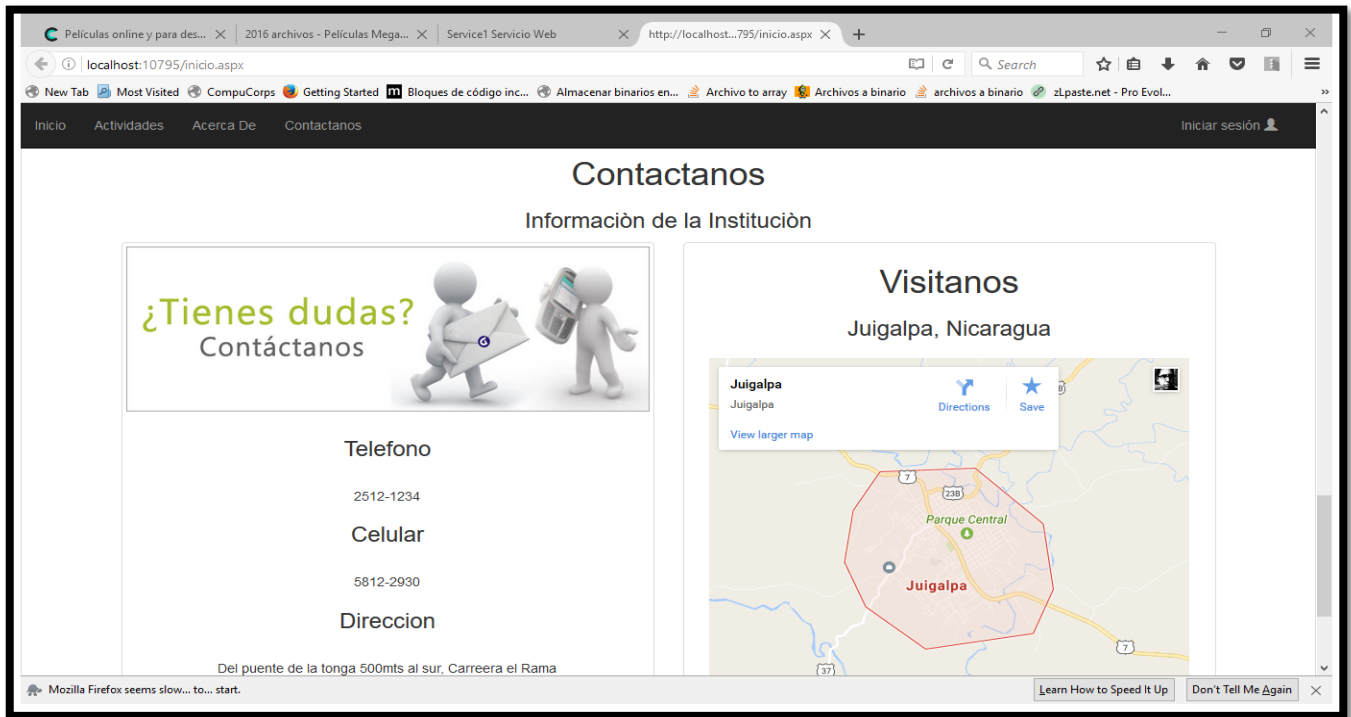


Ilustración 3 - Interfaz Página Principal

Al seleccionar “Iniciar sesión” en el menú de la página principal, nos direccionara a la página de Inicio de Sesión, es aquí donde cada usuario ingresa sus credenciales de inicio y accede al sistema.

Ilustración 4 - Interfaz Inicio de sesión

3. Interfaz del usuario Administrador:

La presente es la pantalla principal mostrada a un usuario tipo administrador, se observan todos los menus disponibles, en donde se diferencian los tres modulos principales del sistema que son Matricula, Notas y Caja.

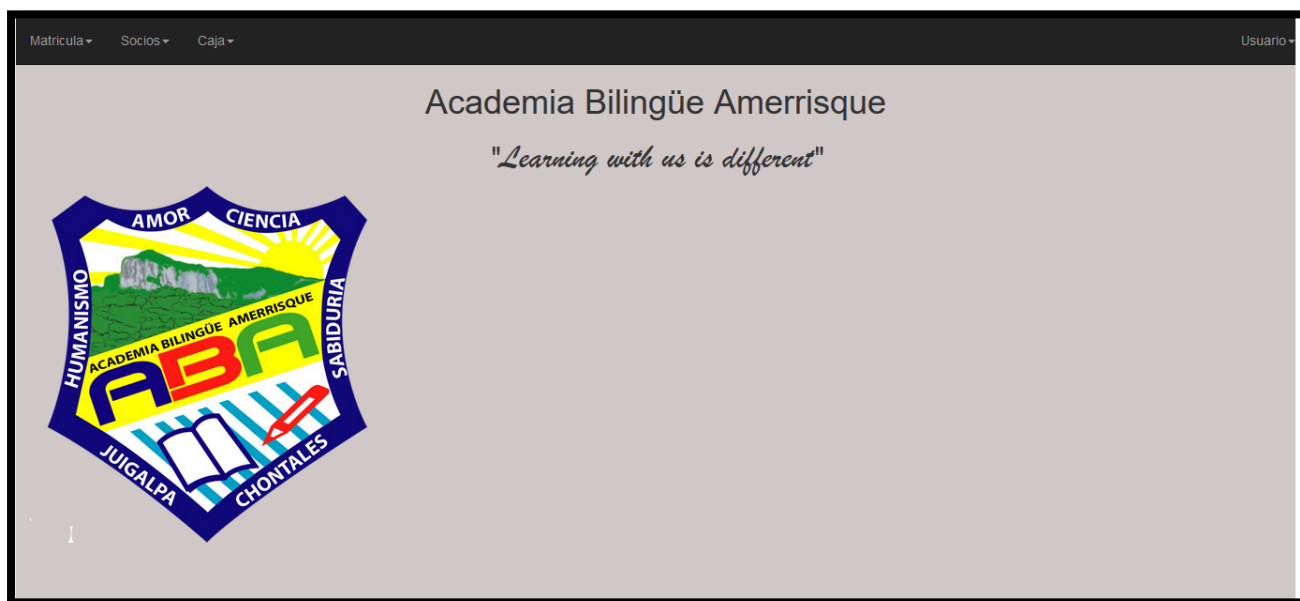


Ilustración 5 - Interfaz Usuario Administrador

Se destaca que este menú que muestra todas las opciones solo podrá ser visualizado por el usuario Administrador ya que es él quien posee los privilegios y está autorizado para controlar algunas transacciones, por ejemplo del módulo de caja.

A continuación vemos las opciones que se encuentran en los menús:

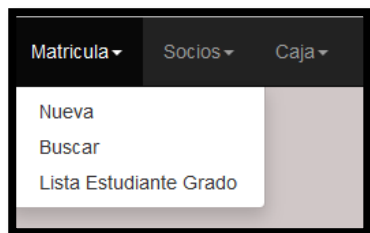


Ilustración 8 - Menu Matricula

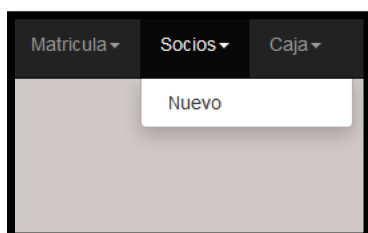


Ilustración 6 - Menú Socios

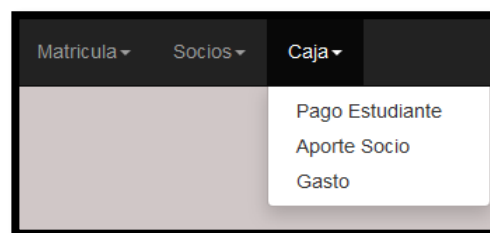


Ilustración 7 - Menú Caja

En el menú <Matricula> <Nueva> se muestra una lista de las opciones de una nueva matricula:

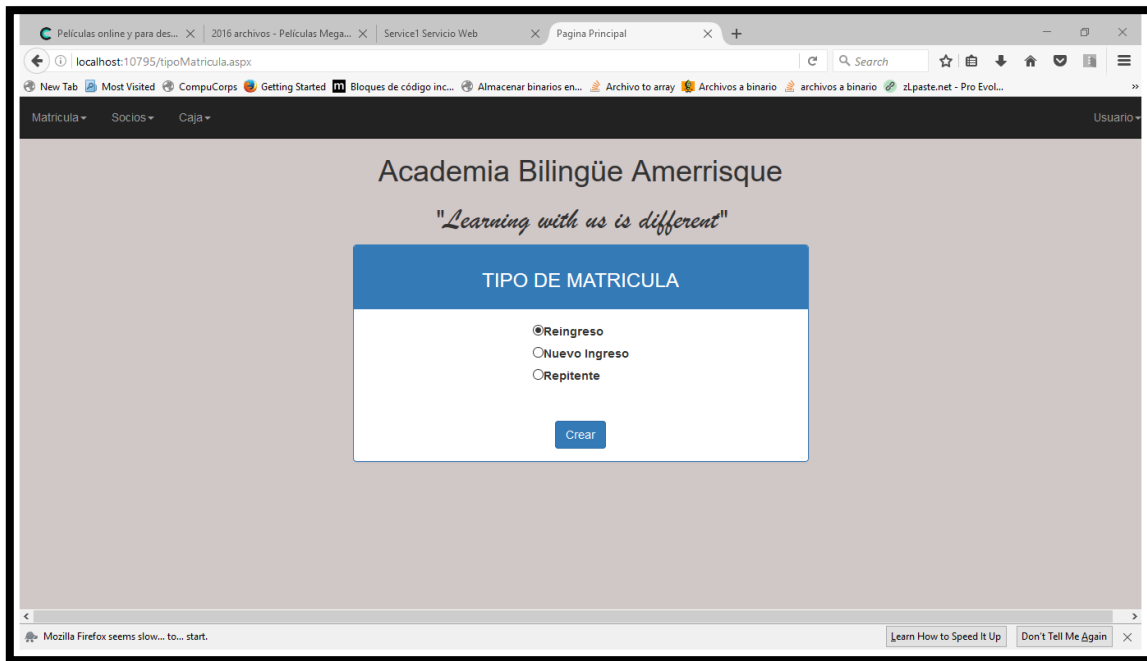


Ilustración 9 - Interfaz Tipo De Matricula

Al seleccionar “Reingreso” se muestra una lista de todos los estudiantes ya registrados en el sistema a los cuales el usuario podrá matricularlos automaticamente:

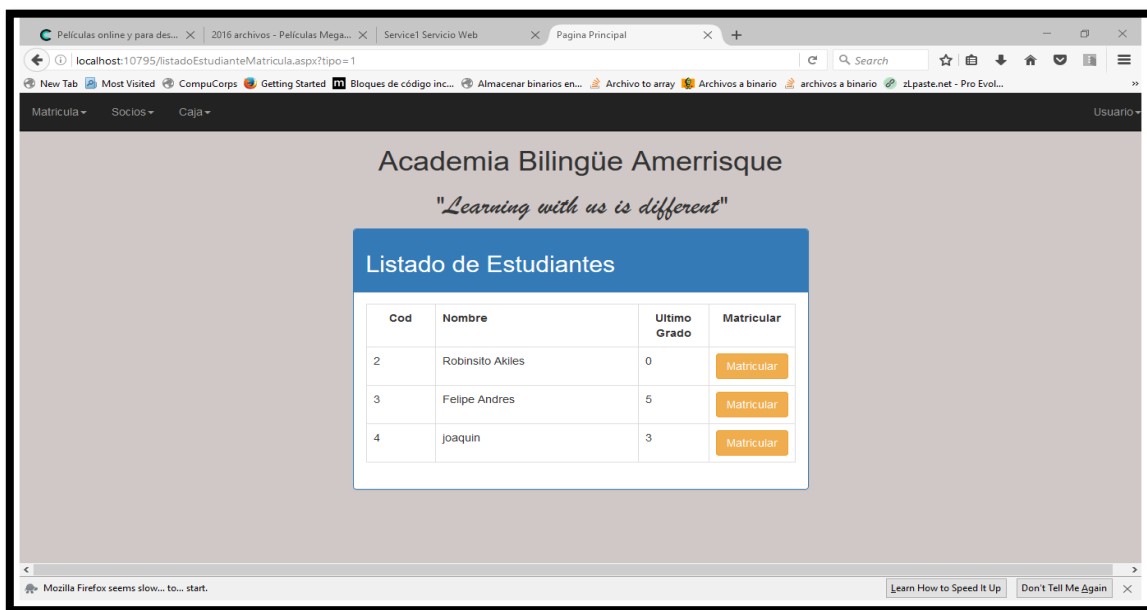


Ilustración 10 - Interfaz Listado de Estudiantes

Al seleccionar la opción “Nuevo Ingreso” el sistema muestra las opciones que siguen:

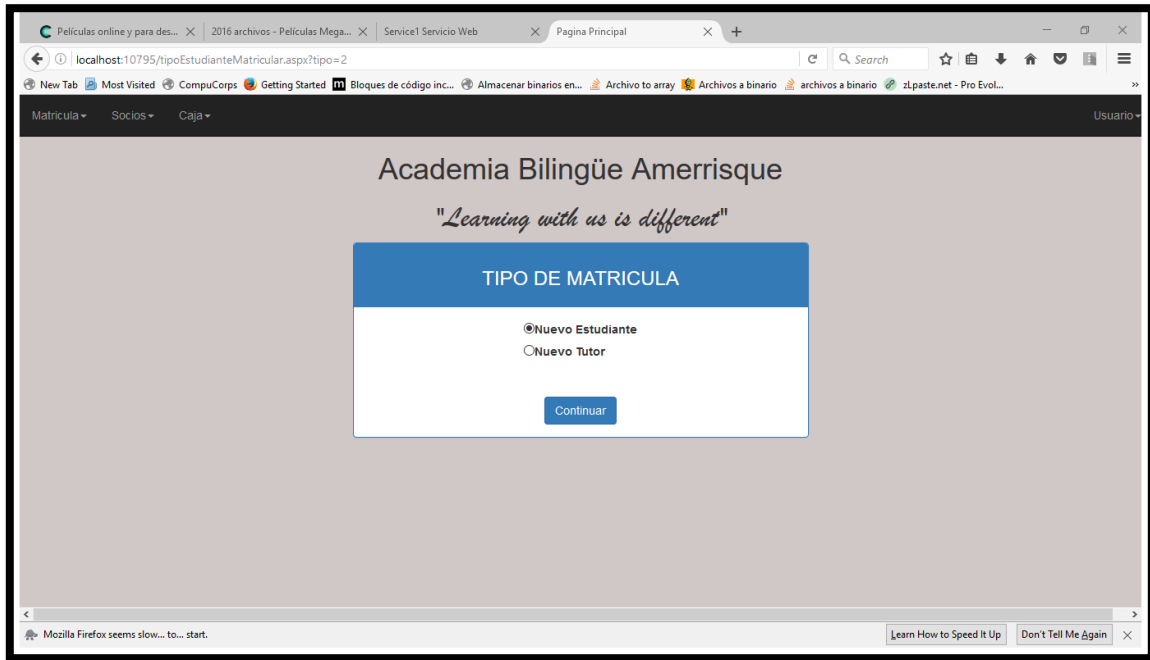


Ilustración 11 - Interfaz Tipo de Matricula

Esto es debido a que el Tutor ya se haya registrado en el sistema, en este caso, el nuevo ingreso solo aplicaria para un Nuevo Estudiante:

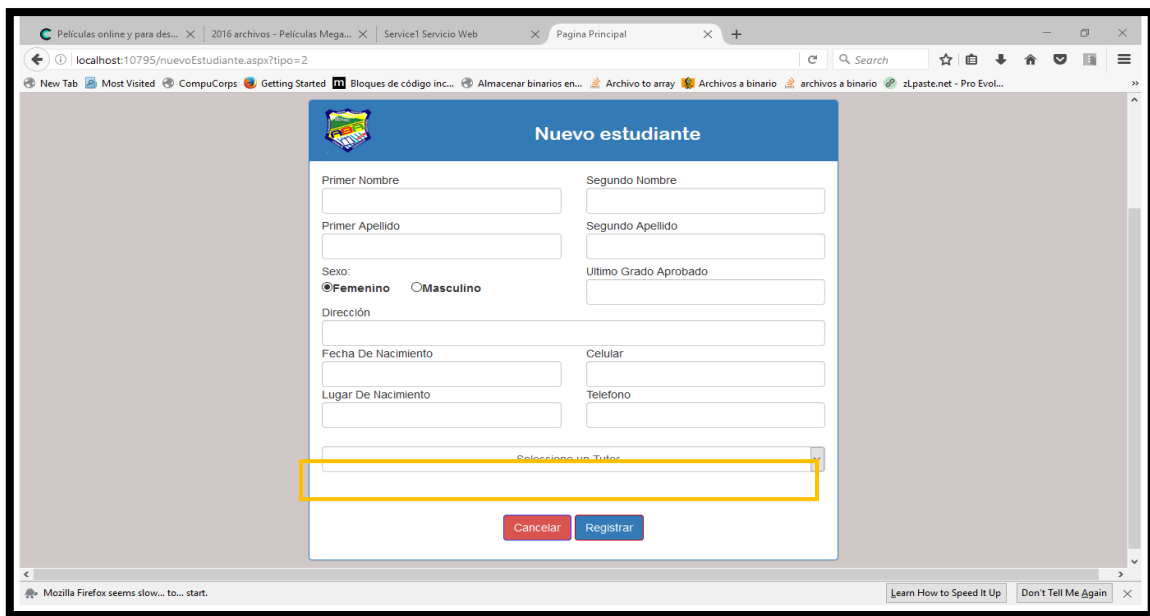


Ilustración 12 - Interfaz Nuevo Estudiante

En este formulario se llena la información general del estudiante y también se le asigna el tutor a cargo listando en un *Listbox* a todos los tutores existentes.

En caso de ser un nuevo Tutor el que será registrado entonces se llena el formulario con los datos del tutor y después el de los datos del estudiante:

Datos del tutor

Nombres

Apellidos

Usuario:

Contraseña: Confirmar Contraseña:

Telefono Celular

Correo Electronico

Dirección de domicilio

Ocupación telefono del trabajo

Dirección del Trabajo:

Ilustración 13 - Interfaz Dato del Tutor

Nuevo estudiante

Primer Nombre Segundo Nombre

Primer Apellido Segundo Apellido

Sexo: Femenino Masculino Ultimo Grado Aprobado

Dirección

Fecha De Nacimiento Celular

Lugar De Nacimiento Telefono

-- Seleccione un Tutor --

Cancelar Registrar

Ilustración 14 - Interfaz Nuevo Estudiante

En el Menu de Socios: Se crea un nuevo socio los cuales haran aportes que entra como ingreso en la contabilidad del colegio.

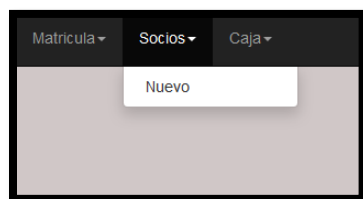


Ilustración 15 - Nuevo Socio

Datos de Socio

Primer Nombre Segundo Nombre

Primer Apellido Segundo Apellido

Direccion Telefono

Usuario

Cancelar Registrar

Ilustración 16 - Interfaz Datos de socio

El módulo de caja que muestra los menús:

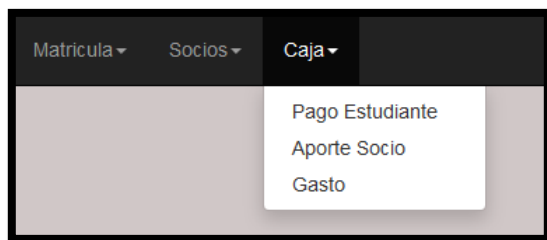


Ilustración 17 - Interfaz Menú de Caja

Para un Pago en caja se listan los estudiantes, el administrador selecciona que estudiante realizará el pago y procede a crea la factura:

| Cod | Nombre | Facturar |
|-----|------------------|----------|
| 2 | Robinsito Akiles | Crear |
| 3 | Felipe Andres | Crear |
| 4 | joaquin | Crear |

Ilustración 18 - Interfaz Listado de Estudiantes

Despues, solamente queda llenar la factura escribiendo el concepto, la cantidad, etc...



The screenshot shows a web interface titled "Ingresos" with a blue header. On the left is a logo of a school building. The main content area contains three input fields: "Concepto" (empty), "Cantidad" (empty), and "Tipo" (a dropdown menu with "Pago Matricula" selected). At the bottom are two buttons: "Cancelar" (red) and "Registrar" (blue).

Ilustración 19 - Interfaz Ingresos


Otro tipo de pago o Ingreso es el “Aporte de Socios” Cuyos datos ya han sido almacenados, por lo tanto en este formulario solamente se registra la cantidad y el concepto del pago o aporte que realizará el socio al colegio:



The screenshot shows a web interface titled "Ingresos" with a blue header. On the left is a logo of a school building. The main content area contains three input fields: "Concepto" (empty), "Cantidad" (empty), and "Tipo" (a dropdown menu with "Luis David Amado Lopez" selected). At the bottom are two buttons: "Cancelar" (red) and "Registrar" (blue).

Ilustración 20 - Interfaz Tipo de Ingreso Aporte de socio

Dentro de el Menú <Caja> en la opción “Egresos” se llena el formulario con los gastos que se registran en el sistema:



Egreso

Descripcion

Precio

Cancelar Registrar

Ilustración 21 - Interfaz Egreso

4. Interfaz del usuario Tutor

Para el usuario Tutor la pantalla principal es la que se muestra a continuación. El usuario puede actualizar sus datos desde el menú “Usuario”. Aquí se le muestra al tutor sus estudiantes actualmente matriculados en el colegio con las opciones que le permite realizar para cada uno.

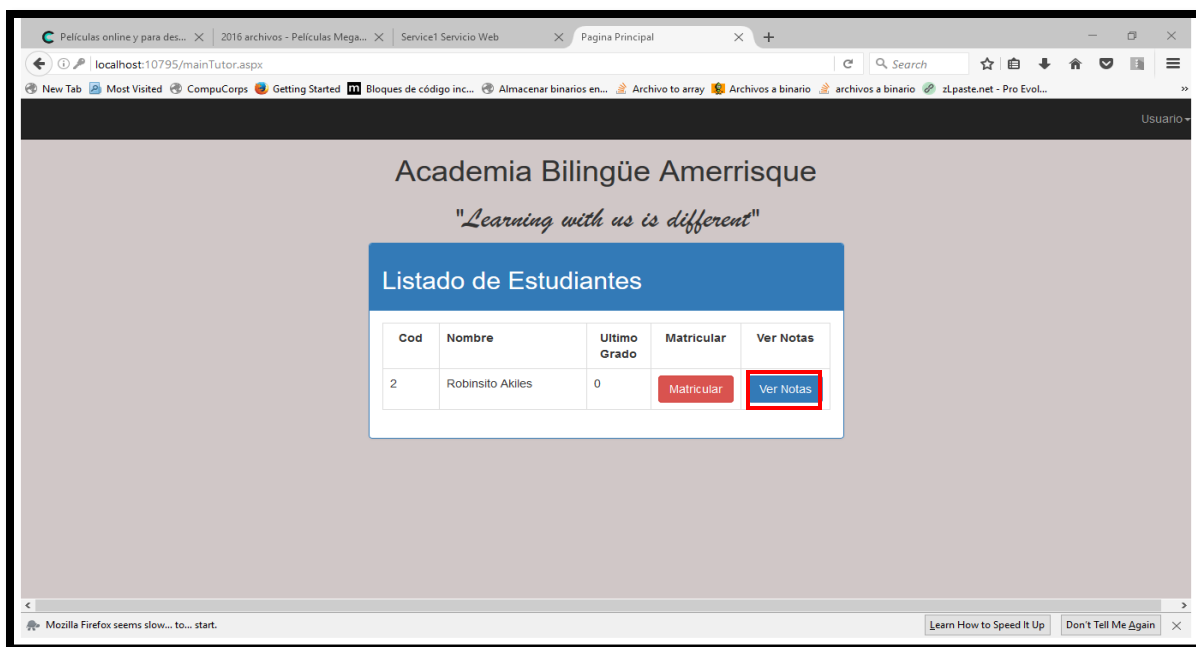


Ilustración 22 - Interfaz pantalla principal tutor

El botón de “Ver Notas” muestra las calificaciones de los estudiantes por corte evaluativo: Estas notas se podrán imprimir posteriormente como un boletín.

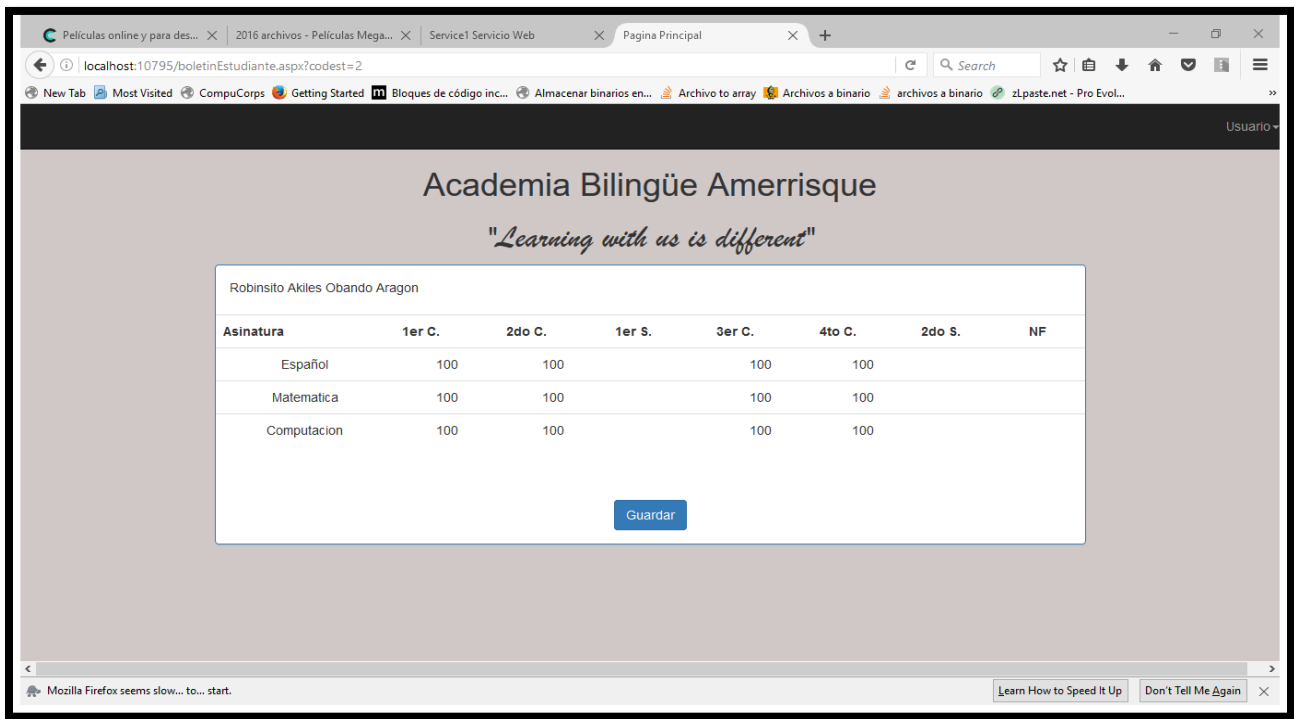


Ilustración 23 - Interfaz Crear Calificación

5. Interfaz del usuario Docente

Para el usuario Docente la pantalla principal muestra una lista de sus grados que tiene a cargo

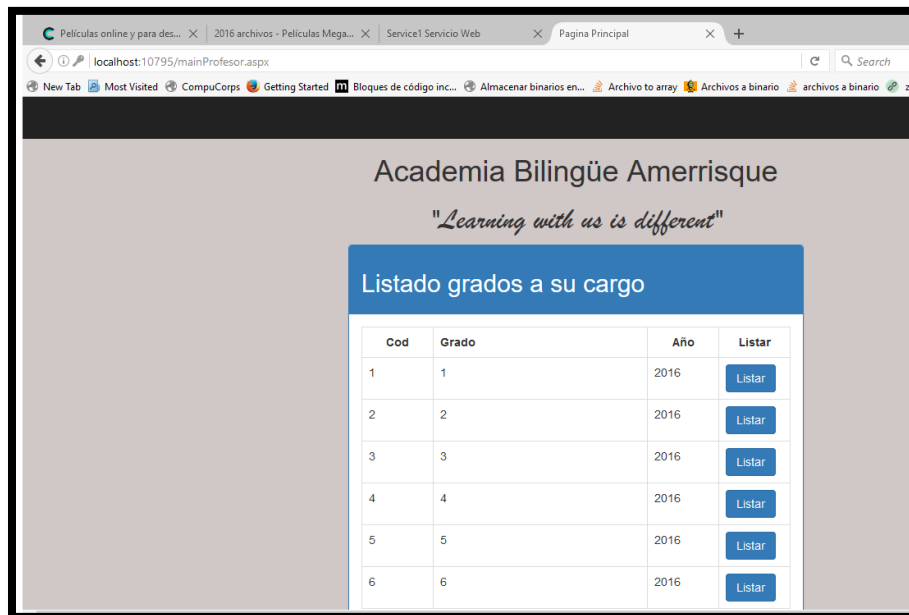


Ilustración 24 - Interfaz Listado de alumno a cargo

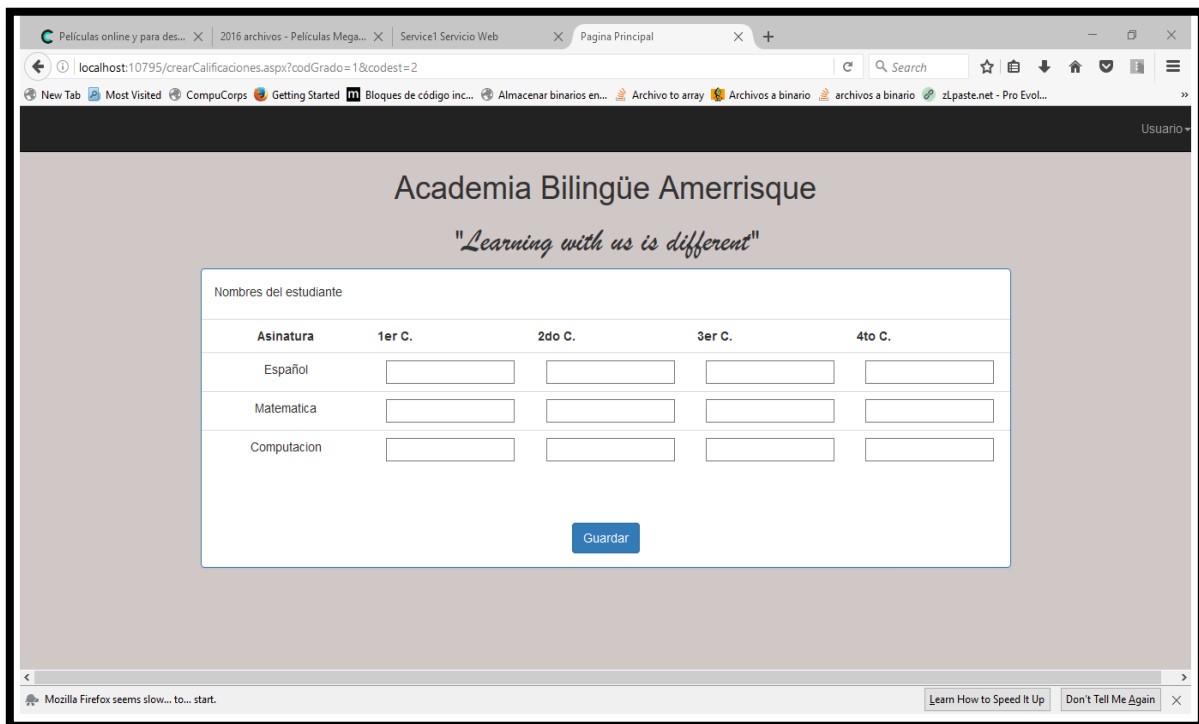
Al listar el grado a cargo se muestra el listado de los estudiantes matriculados en ese grado:



| Cod | Nombre | Evaluar |
|-----|-----------------------------------|--|
| 2 | Robinsito Robinsito Obando Aragon | <input type="button" value="Calificar"/> |

Ilustración 25 - Listado de Grado A cargo

En el botón “Evaluar” el sistema direcciona a la página donde el docente escribe las notas de los estudiantes por corte evaluativo.



Academia Bilingüe Amerrisque
"Learning with us is different"

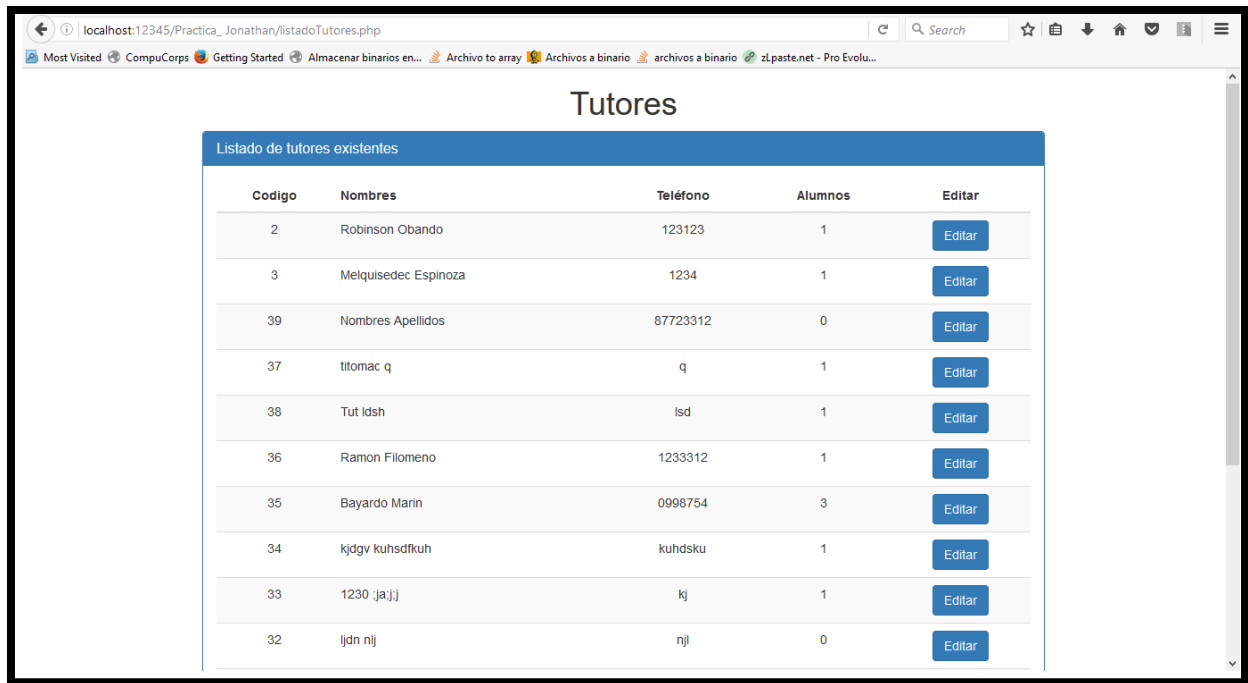
Nombres del estudiante

| Asinatura | 1er C. | 2do C. | 3er C. | 4to C. |
|-------------|----------------------|----------------------|----------------------|----------------------|
| Español | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| Matematica | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |
| Computacion | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

Ilustración 26 - Interfaz Crear Calificaciones

Al guardar la calificación esta se actualiza automáticamente a lo que será mostrado al tutor.

El menú de listado de tutores muestra una lista de los mismos, su teléfono, los alumnos que están asignados al mismo y la posibilidad de editar sus datos.



The screenshot shows a web browser window with the URL localhost:12345/Practica_Jonathan/listadoTutores.php. The page title is 'Tutores'. Below the title is a table titled 'Listado de tutores existentes'. The table has five columns: 'Codigo', 'Nombres', 'Teléfono', 'Alumnos', and 'Editar'. Each row represents a tutor and includes an 'Editar' button.

| Codigo | Nombres | Teléfono | Alumnos | Editar |
|--------|----------------------|----------|---------|--------|
| 2 | Robinson Obando | 123123 | 1 | Editar |
| 3 | Meiquesedec Espinoza | 1234 | 1 | Editar |
| 39 | Nombres Apellidos | 87723312 | 0 | Editar |
| 37 | titomac q | q | 1 | Editar |
| 38 | Tut Idsh | Isd | 1 | Editar |
| 36 | Ramon Filomeno | 1233312 | 1 | Editar |
| 35 | Bayardo Marin | 0998754 | 3 | Editar |
| 34 | kjgfv kuhdsfkuh | kuhdsku | 1 | Editar |
| 33 | 1230 .ja.jj | kj | 1 | Editar |
| 32 | ljdn nij | nij | 0 | Editar |

Ilustración 27 - Interfaz Listas de tutores

Al escoger editar un tutor, nos muestra todos sus datos, los cuales pueden ser modificados a voluntad.

Editando Datos

Nombres:
Melquisedec

Apellidos:
Espinoza

Direccion:
Americable 1/2 c. al sur

Telefono:
1234

Correo:
melqui@gmail.com

Datos de Inicio

Usuario:
mel

Contraseña:
123

Modificar

Ilustración 28 - Interfaz Modificar datos del tutor

Esta es la pantalla de edición de datos del usuario, como puede apreciar es idéntica a la de modificar tutor, con la diferencia que aquí se modifica el usuario que está activo en el momento.

Editando Datos

Nombres:
Jorge

Apellidos:
Cruz

Direccion:
Curacao 1c. al sur

Telefono:
12345678

Correo:
jorcruz2@gmail.com

Datos de Inicio

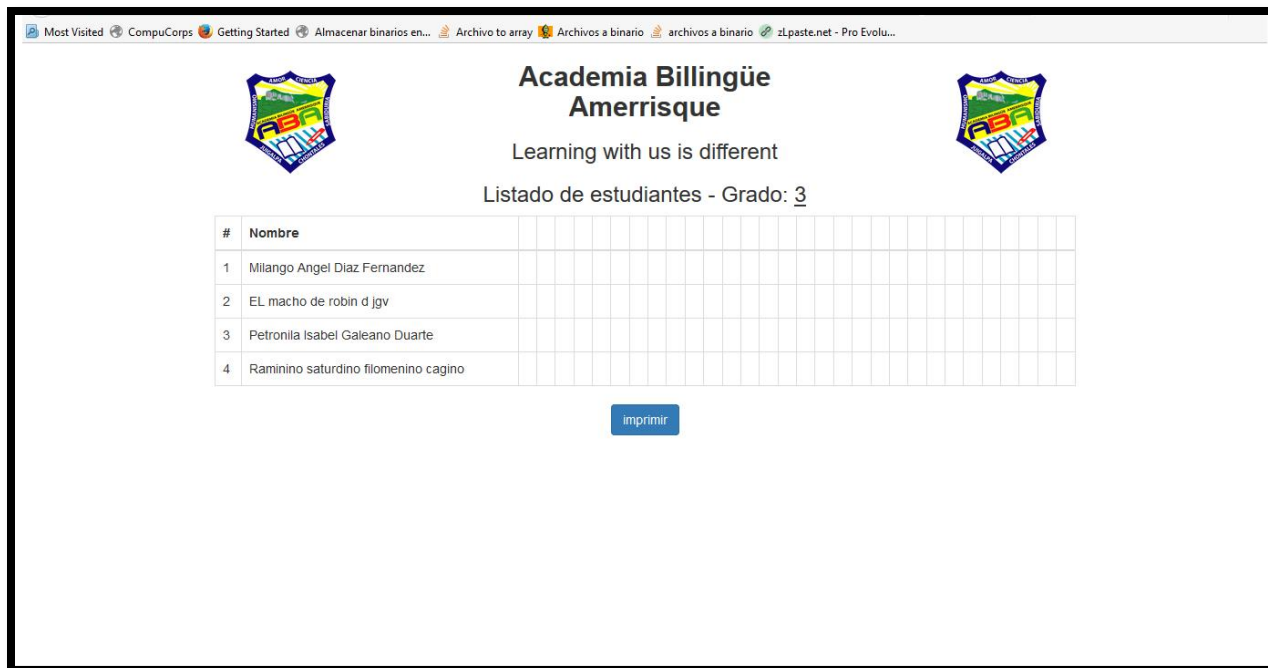
Usuario:
jor

Contraseña:
123

Modificar

Ilustración 29 - Interfaz Modificar datos del tutor

Finalmente si en el menú de “Estudiantes” seleccionamos un grado, nos presentara una esquila que podrá ser impresa para facilitar a los docentes.



The screenshot shows a web browser window with the following content:

- Browser tabs: Most Visited, CompuCorps, Getting Started, Almacenar binarios en..., Archivo to array, Archivos a binario, archivos a binario, zLpaste.net - Pro Evolu...
- Logo: Academia Billingüe Amerrisque (Bilingual Education)
- Text: **Academia Billingüe Amerrisque**
- Slogan: Learning with us is different
- Section: Listado de estudiantes - Grado: 3
- Table with 4 rows and 10 columns:

| # | Nombre | | | | | | | | |
|---|--------------------------------------|--|--|--|--|--|--|--|--|
| 1 | Milango Angel Diaz Fernandez | | | | | | | | |
| 2 | EL macho de robin d jgv | | | | | | | | |
| 3 | Petronila Isabel Galeano Duarte | | | | | | | | |
| 4 | Raminino saturdino filomenino cagino | | | | | | | | |

Below the table is a blue button labeled "imprimir".

Ilustración 30 - Listado de estudiante por grado

Al Cerrar Sesión Somos enviados nuevamente a la pantalla de inicio.

Etapa de Pruebas del software

SIS-I

Planes de Pruebas Funcionales

Versión: 01

Fecha: Enero 2017

Versión del software 2.5

Contenido

| | |
|--|---|
| 1. Hoja de control | 1 |
| 2. Objetivo | 2 |
| 3. Alcance | 2 |
| 4. Trazabilidad de casos de pruebas – requisitos | 3 |
| 5. Definición de los casos de pruebas | 4 |

Índice de tablas

| | |
|---|----|
| Tabla 1 - Hoja de control..... | 1 |
| Tabla 2 - Registro de cambios | 1 |
| Tabla 3 - Control de distribución..... | 1 |
| Tabla 4 - Trazabilidad de caso de prueba | 3 |
| Tabla 5 - Definición de los caso de prueba | 4 |
| Tabla 6 - Datos de usuario erróneos | 5 |
| Tabla 7 - Autenticación de tipos usuarios | 6 |
| Tabla 8 - Registro de tutores..... | 7 |
| Tabla 9 - Registro de estudiantes..... | 8 |
| Tabla 10 - Registro de Docentes..... | 9 |
| Tabla 11 - Actualizar la contraseña | 10 |
| Tabla 12 - Modificar datos personales o de usuarios | 11 |
| Tabla 13 - Matricula | 12 |

1. Hoja de control

| | | | |
|------------------------|---|-------------------------|-------------|
| Organismo | Academia Bilingüe Amerrisque | | |
| Proyecto | SIS-I (Sistema de información web de gestión académica) | | |
| Entregable | Planes de Pruebas Funcionales | | |
| Autor | Joaquín Espinoza, Robinson Obando | | |
| Versión/Edición | 01 | Fecha Versión | Enero, 2017 |
| Aprobado por | | Fecha Aprobación | |

Tabla 1 - Hoja de control

REGISTRO DE CAMBIOS

| Versión doc | Causa del Cambio | Responsable del Cambio | Fecha del Cambio |
|--------------------|-------------------------|-------------------------------|-------------------------|
| 01 | Versión inicial | Joaquín Espinoza | Enero, 2017 |
| 01 | Versión inicial | Robinson Obando | Enero, 2017 |
| | | | |

Tabla 2 - Registro de cambios

CONTROL DE DISTRIBUCIÓN

| Nombre y Apellidos |
|---------------------------|
| Joaquín Espinoza |
| Robinson Obando |
| |
| |
| |

Tabla 3 - Control de distribución

2. Objetivo

El objetivo de este documento es recoger los casos de pruebas que verifican que el sistema satisface los requisitos especificados. Deberá contener la definición de los casos de prueba, la matriz de trazabilidad entre casos de pruebas y requisitos, y la estrategia a seguir en la ejecución de las pruebas.

3. Alcance

Este documento pretende evaluar y contrastar de manera comparativa y objetiva en base a los requerimientos funcionales los principales procesos del sistema, definiendo los prerrequisitos que necesita cada proceso para funcionar correctamente y brindando los resultados que se esperan y los que genera el sistema con cada caso de prueba, de esta manera las responsabilidades a las que va dirigida este documento son a los mismos desarrolladores y a los usuarios del software, las validaciones y registros que se hacen en este formato de pruebas están realizadas según el modelo de MADEJA.

4. Trazabilidad de casos de pruebas – requisitos

En este apartado se muestra una matriz la cual indica la correspondencia entre los casos de pruebas definidos, y los requisitos funcionales de la especificación de requisitos. Las filas representan cada uno de los casos de pruebas definidos, y las columnas los requisitos funcionales. Si un caso de prueba se encarga de verificar un requisito, se señala con una X la casilla correspondiente.

| | RF-001 | RF-002 | RF-003 | RF-004 | RF-006 | RF-007 | RF-010 |
|------|--------|--------|--------|--------|--------|--------|--------|
| CP01 | X | | | | | | |
| CP02 | | X | | | | | |
| CP03 | | X | | | | | |
| CP04 | | | X | | | | |
| CP05 | | | X | | | | |
| CP06 | | | | X | | | |
| CP07 | | | | | X | | |
| CP08 | | | | | | X | |
| CP09 | | | | | | | X |

Tabla 4 - Trazabilidad de caso de prueba

5. Definición de los casos de pruebas

Con el primer caso de uso que es para acceder al sitio web del sistema a través de la cabecera http la cual deberá ser escrita correctamente antes de realizar alguna otra transacción con este sistema.

| Ingreso al sitio web del sistema | CP01 | |
|--|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Este caso de prueba verifica si se puede acceder al sitio desde cualquier dispositivo a través de la cabecera HTTP | | |
| Prerrequisitos <ol style="list-style-type: none">1. Que el sistema se encuentre alojado en un servidor.2. Que la dirección del sitio esté claramente definida. | | |
| Pasos: <ol style="list-style-type: none">1. Escribir la dirección en la cabecera http desde cualquier navegador de internet. | | |
| Resultado esperado: El navegador abre el sitio web del SIS-I | | |
| Resultado obtenido: El navegador abre el sitio web del SIS-I | | |

Tabla 5 - Definición de los casos de prueba

En este caso de prueba se pretende validar que la seguridad del sistema permita que no pase desapercibida una equivocación al momento de querer acceder al sistema y que por haber escrito mal las credenciales el sistema muestre un mensaje de alerta que avise al usuario que no son sus credenciales correctas.

| Datos de usuario erróneos | CP02 | |
|--|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Verifica que los datos de los usuarios estén almacenados en la base de datos para poder acceder a su cuenta en el sistema. | | |
| Prerrequisitos <ol style="list-style-type: none"> 1. Que los usuarios se hayan registrado previamente en el sistema. | | |
| Pasos: <ol style="list-style-type: none"> 1. Dar clic en el botón “Inicio de Sesión” 2. Escribir los datos del usuario. 3. Clic en Botón Aceptar | | |
| Resultado esperado: El sistema muestra un mensaje de datos erróneos. | | |
| Resultado obtenido: El sistema muestra un mensaje de datos erróneos. | | |

Tabla 6 - Datos de usuario erróneos

En este CP el sistema valida las credenciales escritas al momento de iniciar sesión y después de comprobar que son correctas realiza una segunda función que se encarga de reconocer las credenciales y a qué tipo de usuario pertenece, por ejemplo con los datos del Administrador.

En este CP el sistema valida las credenciales escritas al momento de iniciar sesión y después de comprobar que son correctas realiza una segunda función que se encarga de reconocer las credenciales y a qué tipo de usuario pertenece, por ejemplo con los datos del Administrador.

| Autenticación de tipos usuarios | CP03 | |
|---|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Verifica que los datos de los usuarios estén almacenados en la base de datos y a qué tipo de usuario pertenecen, para poder acceder a su cuenta en el sistema. | | |
| Prerrequisitos 2. Que los usuarios se hayan registrado previamente en el sistema. | | |
| Pasos: 4. Dar clic en el botón “Inicio de Sesión” 5. Escribir los datos del usuario. Nombre = “admin”; Contraseña = “123”; 6. Clic en Botón Aceptar | | |
| Resultado esperado: Muestra la pantalla principal del Administrador. | | |
| Resultado obtenido: Muestra la pantalla principal del Administrador. | | |

Tabla 7 - Autenticación de tipos usuarios

En el siguiente caso de prueba se comprueba que se guarde correctamente un usuario de tipo “Tutor” llenando datos correctos en el formulario de Registro el cual tiene la opción de Guardar la información y pasar a otro registro el de “Nuevo estudiante”

| Registro de tutores | CP04 | |
|---|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Comprueba que los datos escritos en los formularios sean almacenados de forma correcta | | |
| Prerrequisitos 1. Que el usuario brinde los datos al administrador | | |
| Pasos: 1. En el menú Matricular elegir la opción de Nueva ingreso 2. Llenar el formulario con la información que solicita. 3. Clic en botón Guardar | | |
| Resultado esperado: El sistema muestra la pantalla de registro de estudiante automáticamente. | | |
| Resultado obtenido: El sistema muestra la pantalla de registro de estudiante automáticamente. | | |

Tabla 8- Registro de tutores

Este caso de prueba hace la comprobación del registro de un Nuevo Estudiante y además verifica que el estudiante se vincule al tutor que se le va asignar.

| Registro de estudiantes | CP05 | |
|---|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Comprueba que los datos escritos en los formularios sean almacenados de forma correcta | | |
| Prerrequisitos 1. Que el usuario brinde los datos al administrador | | |
| Pasos: 1. En el menú Matricular elegir la opción de Nueva ingreso 2. Llenar el formulario con la información que solicita. 3. Clic en botón Guardar | | |
| Resultado esperado: El sistema muestra un mensaje que se ha guardado exitosamente | | |
| Resultado obtenido: El sistema muestra un mensaje que se ha guardado exitosamente | | |

Tabla 9 - Registro de estudiantes

Este caso de prueba verifica que se almacene correctamente el registro de un usuario de tipo “Docente” llenando los datos personales en el formulario de registro y después aplicando la opción que brinda el botón “Guardar” el cual despliega un mensaje que indica que los datos son correctos o no.

| Registro de Docentes | CP06 | |
|--|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Comprueba que los datos escritos en los formularios sean almacenados de forma correcta | | |
| Prerrequisitos 1. Que el usuario brinde los datos al administrador | | |
| Pasos: 1. Opción registro de Nuevo Docente 2. Llenar los datos que pide el formulario. 3. Clic en botón Guardar. | | |
| Resultado esperado: El sistema muestra un mensaje que se ha guardado exitosamente | | |
| Resultado obtenido: El sistema muestra un mensaje que se ha guardado exitosamente | | |

Tabla 10 - Registro de Docentes

En este caso de prueba el sistema permite que los usuarios actualicen sus contraseñas, si se hace de forma correcta se procede a actualizar la información del usuario.

| Actualización de Password | CP07 | |
|---|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Permite actualizar la contraseña de usuario. | | |
| Prerrequisitos 1. El usuario debe acceder al sistema | | |
| Pasos: 1. El usuario accede al sistema 2. En la opción de Cambio de Contraseña del menú con su nombre 3. Escribe la nueva contraseña y confirma 4. Clic en “Cambiar” | | |
| Resultado esperado: El sistema actualiza los datos cerrando la sesión del usuario | | |
| Resultado obtenido: El sistema actualiza los datos cerrando la sesión del usuario | | |

Tabla 11 - Actualizar la contraseña

El siguiente caso de prueba verifica que los datos de los usuarios puedan ser editados y actualizados con precisión de manera que cuando el administrador realice la edición de los datos de cualquier usuario ya sean datos personales o datos de usuario estos se actualicen correctamente en la base de datos del sistema.

| Modificar datos personales o de usuarios | CP08 | |
|--|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Permite actualizar la información personal de los usuarios del sistema y la información de los datos de usuario. | | |
| Prerrequisitos <ol style="list-style-type: none"> 1. Que los usuarios ya estén registrados en el sistema 2. Tener la información correcta con la que será actualizada. | | |
| Pasos: <ol style="list-style-type: none"> 1. El administrador accede al sistema 2. El administrador busca a los usuarios 3. Elige la opción “Modificar” 4. Actualiza la información del usuario que necesite 5. Da clic en Guardar | | |
| Resultado esperado: El sistema actualiza los datos de los usuarios correctamente y sale de la pantalla de Modificar | | |
| Resultado obtenido: El sistema actualiza los datos de los usuarios correctamente y sale de la pantalla de Modificar | | |

Tabla 12 - Modificar datos personales o de usuarios

En este caso de prueba se verifica que el Tutor realice una Matricula correctamente, validando la función de Matricular que aparece en las opciones disponibles para los usuarios Tutores en su cuenta. Este usuario lista a sus estudiantes en una tabla y tiene la opción de matricular a cada uno.

| Matricular | CP09 | |
|---|------------------------|----|
| | ¿Prueba de despliegue? | No |
| Descripción: Verifica que el tutor pueda matricular al estudiante que tiene a su cargo. | | |
| Prerrequisitos <ol style="list-style-type: none"> 1. Que el usuario este registrado en el sistema 2. Que el estudiante no esté matriculado | | |
| Pasos: <ol style="list-style-type: none"> 1. El usuario debe acceder al sistema 2. Seleccionar el estudiante que desea matricular 3. Clic en Matricular | | |
| Resultado esperado: El sistema envía un mensaje de éxito al realizar la matricula | | |
| Resultado obtenido: El sistema envía un mensaje de éxito al realizar la matricula | | |

Tabla 13 - Matricula