

UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA



PROYECTO DE GRADO

**“SISTEMA DE INFORMACIÓN Y SEGUIMIENTO ACADÉMICO
COLEGIO NACIONAL MIXTO BICENTENARIO LIBERTADOR
SIMÓN BOLÍVAR”**

PARA OPTAR AL TÍTULO DE LICENCIATURA EN INFORMÁTICA
MENCION: INGENIERÍA DE SISTEMAS INFORMÁTICOS

POSTULANTE: EDWIN LUIS CALLISAYA CONDORI
TUTOR METODOLÓGICO: M.Sc. ROSA FLORES MORALES
ASESOR: M.Sc. CARLOS MULLISACA CHOQUE

LA PAZ – BOLIVIA

2018



**UNIVERSIDAD MAYOR DE SAN ANDRÉS
FACULTAD DE CIENCIAS PURAS Y NATURALES
CARRERA DE INFORMÁTICA**



LA CARRERA DE INFORMÁTICA DE LA FACULTAD DE CIENCIAS PURAS Y NATURALES PERTENECIENTE A LA UNIVERSIDAD MAYOR DE SAN ANDRÉS AUTORIZA EL USO DE LA INFORMACIÓN CONTENIDA EN ESTE DOCUMENTO SI LOS PROPÓSITOS SON ESTRICTAMENTE ACADÉMICOS.

LICENCIA DE USO

El usuario está autorizado a:

- a) visualizar el documento mediante el uso de un ordenador o dispositivo móvil.
- b) copiar, almacenar o imprimir si ha de ser de uso exclusivamente personal y privado.
- c) copiar textualmente parte(s) de su contenido mencionando la fuente y/o haciendo la referencia correspondiente respetando normas de redacción e investigación.

El usuario no puede publicar, distribuir o realizar emisión o exhibición alguna de este material, sin la autorización correspondiente.

TODOS LOS DERECHOS RESERVADOS. EL USO NO AUTORIZADO DE LOS CONTENIDOS PUBLICADOS EN ESTE SITIO DERIVARÁ EN EL INICIO DE ACCIONES LEGALES CONTEMPLADOS EN LA LEY DE DERECHOS DE AUTOR.

Dedicatoria

A mi padre Pablo a quien amo mucho, por estar siempre a mi lado, brindándome todo su apoyo constante, por inculcarme valores precisados en la vida.

A mi mami Antonia a la mujer que más amo, quien es mi ejemplo de vida, por ser la mejor mama del mundo, quien me brinda todo el amor, comprensión y vela que nunca me falte nada.

A mi hermano Richard y mi cuñada Yolanda a quienes quiero mucho, por la paciencia y el constante apoyo que recibo de ellos.

A mis hermosos sobrinos Eduardo, Anahí y Grover quienes con su alegría y ternura siempre me supieron sacar una sonrisa.

AGRADECIMIENTOS

Agradecer ante todo a Dios por haberme guiado a lo largo de mi vida, mostrándome el camino que me trajo hasta estas alturas de mi vida.

A mi revisor M.Sc. Carlos Mullisaca Choque, que a pesar de las adversidades siempre me fue apoyando y guiando hacia adelante, transmitiéndome sus conocimientos para poder concluir el presente trabajo.

A mi tutora M.Sc. Rosa Flores Morales, por todo su tiempo y paciencia en el seguimiento del presente trabajo, por su desinteresado apoyo y enseñanza en las materias en las cuales tuve el placer de tenerla como docente, dejándome entender que podemos mejorar la calidad de vida en el mundo con nuestros conocimientos.

Agradecer a mis amigos, por haberme hecho pasar momentos inolvidables en todo el transcurso de mi formación académica, agradecer a todos aquellos que estuvieron apoyándome.

Por ultimo agradecer a la Universidad Mayor de San Andrés, la Carrera de Informática, por haberme acogido a lo largo del transcurso de mi vida universitaria.

RESUMEN

El Colegio Nacional Mixto Bicentenario Libertador Simón Bolívar se encuentra ubicado en la ciudad de El Alto, entre sus objetivos se tiene la educación de niños y jóvenes, además de mantener actualizada su modalidad de enseñanza. En el manejo de la información es donde se pudo notar ciertos problemas como ser lentitud en los trámites, demora en inscripciones, debido a que todas estas tareas se las realiza en forma manual, a su vez esto conlleva a reclamos por parte de los padres de familia como de los mismos estudiantes ya que no pueden tener información de manera rápida y segura.

Tomando en cuenta lo mencionado anteriormente, el presente proyecto tiene como objetivo principal la implementación del “Sistema de Información y Seguimiento Académico del Colegio Nacional Mixto Bicentenario Libertador Simón Bolívar” a fin de facilitar el manejo de datos que existe en la unidad educativa, brindar información hacia los estudiantes acerca de las diferentes actividades como también la del seguimiento académico, organización horarios entre otros, de manera que facilita información tanto para estudiantes como padres de familia.

Para el análisis y desarrollo del Sistema Académico mencionado se utilizó la Metodología de Diseño de Hipertextos Orientada a Objetos (OOHDM) con apoyo de los diagramas Lenguaje de Modelado Unificado (UML) como ser diagramas de casos de uso, diagrama de secuencias, diagrama de clases entre otros, con esta metodología es que se generó una serie de documentos que permiten la implementación del sistema.

Con la implementación de este Sistema se logró coadyuvar en el manejo de información tanto de los estudiantes como de los profesores y personal administrativo, lo que permite tener información en forma inmediata asegurando la privacidad de los mismos.

Palabras Clave: Sistema Académico, Sistema Responsivo, Información Académica, Seguimiento Académico, plataforma educativa.

ÍNDICE

CAPITULO 1. MARCO REFERENCIAL

| | Pág. |
|--|------|
| 1.1. Introducción..... | 1 |
| 1.2. Antecedentes..... | 2 |
| 1.2.1. Antecedentes de la Institución..... | 2 |
| 1.2.1. Antecedentes del Proyecto..... | 4 |
| 1.3.Planteamiento del Problema..... | 5 |
| 1.3.1. Análisis del Problema..... | 6 |
| 1.3.2. Problema Principal..... | 6 |
| 1.4.Objetivos..... | 6 |
| 1.4.1. Objetivo General..... | 6 |
| 1.4.2. Objetivos Específicos..... | 7 |
| 1.5. Justificación..... | 7 |
| 1.5.1. Justificación Técnica..... | 7 |
| 1.5.2. Justificación Social..... | 8 |
| 1.5.1. Justificación Económica..... | 8 |
| 1.6. Alcances y Limites..... | 8 |
| 1.6.1. Alcances..... | 8 |
| 1.6.1.1. Alcance Geográfico..... | 8 |
| 1.6.1.1. Alcance Temporal..... | 9 |
| 1.6.1.1. Alcance Temático..... | 9 |
| 1.6.2. Limites..... | 10 |
| 1.7. Aportes..... | 10 |
| 1.8. Metodología..... | 10 |

CAPITULO 2. MARCO TEÓRICO

| | |
|--|----|
| 2.1. Tecnología Web..... | 11 |
| 2.1.1. Sistemas de Información Basadas en la Web..... | 11 |
| 2.1.1.1. Sitio Web..... | 13 |
| 2.1.1.2. Página Web..... | 13 |
| 2.2. Metodología de diseño de Hipertextos Orientado a Objetos (OOHDM)..... | 14 |
| 2.2.1. Obtención de Requerimientos..... | 15 |
| 2.2.2. Diseño Conceptual..... | 16 |
| 2.2.3. Diseño Navegacional..... | 17 |
| 2.2.4. Diseño de Interfaz Abstracta..... | 19 |
| 2.2.5. Implementación..... | 19 |
| 2.3. Diseño Arquitectónico..... | 20 |
| 2.3.1. Modelo Vista Controlador (MVC)..... | 21 |
| 2.3.2. Arquitectura Rest..... | 22 |
| 2.4. Stack Pevn..... | 22 |
| 2.5. Calidad de Software..... | 24 |
| 2.5.1. Calidad de Software con ISO-9126..... | 24 |
| 2.5.2 Estimación del Tamaño por Puntos Función..... | 26 |

CAPITULO 3. MARCO APLICATIVO

| | |
|--|----|
| 3.1. Análisis de la Institución..... | 29 |
| 3.1.1. Organización del Modelo Ambiental..... | 29 |
| 3.1.2. Descripción roles en el Sistema..... | 30 |
| 3.1.3. Diagramas de Contexto del Sistema..... | 31 |
| 3.2. Metodología de Diseño de Hipertextos Orientado a Objetos (OOHDM)..... | 32 |
| 3.2.1. Obtención de Requerimientos..... | 32 |
| 3.2.1.1. Requerimientos Funcionales..... | 34 |

| | |
|---|----|
| 3.2.1.2. Requerimientos no Funcionales..... | 35 |
| 3.2.1.3. Diagramas de Casos de Uso..... | 35 |
| 3.2.1.3.1. Caso de Uso Sistema Académico..... | 36 |
| 3.2.1.3.2. Caso de Uso Estudiante..... | 36 |
| 3.2.1.3.3. Caso de uso Profesor..... | 38 |
| 3.2.1.3.4. Caso de Uso Administración..... | 39 |
| 3.2.2. Modelo Conceptual..... | 41 |
| 3.2.2.1. Diseño Orientado a Objetos..... | 42 |
| 3.2.2.2. Diseño de la Base de Datos..... | 42 |
| 3.2.3. Diseño Navegacional..... | 44 |
| 3.2.3.1. Diseño Navegacional Principal..... | 44 |
| 3.2.3.2. Diseño Navegacional Administrador..... | 45 |
| 3.2.3.3. Diseño Navegacional Estudiante..... | 46 |
| 3.2.3.4. Diseño Navegacional Profesor..... | 47 |
| 3.2.3.5. Modelo de Clases Navegacionales..... | 48 |
| 3.2.4. Diseño de Interfaz Abstracta..... | 49 |
| 3.2.4.1. Interfaz Abstracta Inicio..... | 50 |
| 3.2.4.2. Interfaz Abstracta Inicio de Sesión..... | 51 |
| 3.2.4.3. Interfaz Abstracta Administrador..... | 52 |
| 3.2.4.4. Interfaz Abstracta Estudiante..... | 54 |
| 3.2.4.5. Interfaz Abstracta Profesor..... | 55 |
| 3.2.5. Implementación..... | 56 |
| 3.2.5.1. Implementación Pagina Inicial..... | 58 |
| 3.2.5.2. Implementación Inicio de Sesión..-..... | 59 |
| 3.2.5.3. Implementación Administrador..... | 60 |
| 3.2.5.4. Implementación Profesor..... | 61 |
| 3.2.5.5. Implementación Estudiante – Tutor..... | 61 |

CAPITULO 4. CALIDAD Y SEGURIDAD

| | |
|---|----|
| 4.1. Calidad del Software..... | 63 |
| 4.2. Calidad ISO/IEC 9126..... | 63 |
| 4.2.1. Funcionalidad..... | 63 |
| 4.2.2. Confiabilidad..... | 66 |
| 4.2.3. Usabilidad..... | 69 |
| 4.2.4. Capacidad de Mantenimiento..... | 70 |
| 4.2.5. Portabilidad..... | 72 |
| 4.2.6. Eficiencia..... | 73 |
| 4.3. Seguridad..... | 74 |
| 4.3.1. Seguridad de la Base de Datos..... | 74 |
| 4.3.1.1. Encriptación de Datos..... | 74 |
| 4.3.1.2. Autenticación de Usuarios..... | 75 |
| 4.3.1.3. Manejo de Sesiones..... | 75 |

CAPITULO 5. CONCLUSIONES Y RECOMENDACIONES

| | |
|----------------------------------|----|
| 5.1. Conclusiones..... | 76 |
| 5.2. Recomendaciones..... | 77 |
| Bibliografía..... | 78 |
| ANEXO A – Manual de Usuario..... | 80 |

ÍNDICE DE FIGURAS

| | Pág. |
|--|------|
| Figura 1 Caso de Uso..... | 15 |
| Figura 2 Sintaxis Diagrama de Clases..... | 16 |
| Figura 3 Descripción Esquema Navegacional..... | 18 |
| Figura 4 Diseño Interfaz Abstracta..... | 19 |
| Figura 5 Proceso Calculo del PF..... | 28 |
| Figura 6 Diagrama de Contexto del Sistema..... | 31 |
| Figura 7 Diagrama de Secuencia Proceso de Inscripción Actual..... | 33 |
| Figura 8 Caso de Uso Seguimiento Académico..... | 36 |
| Figura 9 Caso de Uso Estudiante Tutor..... | 37 |
| Figura 10 Caso de Uso Profesor Sistema Académico..... | 38 |
| Figura 11 Caso de Uso Administrador Sistema Académico..... | 39 |
| Figura 12 Diagrama de Clases del Sistema Académico..... | 42 |
| Figura 13 Diagrama Relacional del Sistema Académico..... | 43 |
| Figura 14 Diagrama Entidad Relación Sistema Académico..... | 44 |
| Figura 15 Diseño Navegacional Inicio de Sesión..... | 45 |
| Figura 16 Diseño Navegacional Modulo Administrador..... | 46 |
| Figura 17 Diseño Navegacional Estudiante..... | 47 |
| Figura 18 Diseño Navegacional Profesor..... | 48 |
| Figura 19 Modelo de Clases Navegacionales..... | 48 |
| Figura 20 Diseño Interfaz Abstracta Página de Inicio..... | 50 |
| Figura 21 Diseño de Interfaz Abstracta Pagina Inicio de Sesión..... | 51 |
| Figura 22 Diseño de Interfaz Abstracta Página de Inicio Administrador..... | 53 |
| Figura 23 Diseño de Interfaz Abstracta Pagina Inicio Estudiante..... | 54 |
| Figura 24 Diseño de Interfaz Abstracta Página de Inicio Docente..... | 55 |
| Figura 25 Implementación Pagina Inicial..... | 58 |
| Figura 26 Implementación Inicio de Sesión..... | 59 |

| | |
|---|----|
| Figura 27 Implementación Pagina Inicio Administrador..... | 60 |
| Figura 28 Implementación Pagina Inicio Profesor..... | 61 |
| Figura 29 Implementación Pagina Inicio Estudiante | 62 |
| Figura 30 Diagrama de Transferencia..... | 68 |

ÍNDICE DE TABLAS

| | Pág. |
|---|------|
| Tabla 1 Requerimientos Funcionales..... | 34 |
| Tabla 2 Descripción Caso de Uso Estudiante Tutor..... | 37 |
| Tabla 3 Descripción Caso de Uso Docente..... | 38 |
| Tabla 4 Descripción Caso de Uso Administrador..... | 40 |
| Tabla 5 Descripción Caso de Uso Inicio de Sesión..... | 41 |
| Tabla 6 Calculo del Punto Función..... | 64 |
| Tabla 7 Rangos de Calificación Punto Función..... | 65 |
| Tabla 8 Calculo de confiabilidad..... | 68 |
| Tabla 9 Calculo de Usabilidad..... | 70 |
| Tabla 10 Calculo de la Eficiencia..... | 71 |

CAPÍTULO I

1. MARCO REFERENCIAL

1.1. INTRODUCCIÓN

Un aspecto muy importante en cualquier ciudad o área rural es la educación ya que de ello dependerá la calidad de vida de sus habitantes, así también el desarrollo de la misma.

La información que se produce en cualquier institución educativa con todas las actividades que realiza cotidianamente llega a ser de gran importancia, de esa información valiosa es que dependerá la toma de decisiones y con ello llegar a las metas que se planteadas.

Actualmente muchas Unidades Educativas hacen uso de computadoras como herramientas de ayuda y apoyo en las actividades que se tienen diariamente, con ayuda de paquetes como Microsoft Excel y otros programas de ofimática que apoyan al proceso educativo, estos programas no ayudan a la automatización de los procesos que se realizan en la institución educativa.

El constante crecimiento de las necesidades de aplicaciones computacionales en las instituciones educativas, hace necesario el desarrollo de sistemas de información oportunos, confiables y seguros que permitan a estas instituciones estar al tanto de la tecnología. Es así, que en el presente trabajo describe un sistema de información de seguimiento académico y administrativo para el colegio nacional mixto bicentenario libertador “SIMÓN BOLÍVAR” que facilita la automatización de los procesos que se realizan en la institución,

llegando a ser una herramienta muy útil para la institución educativa la cual permite que la información se guarde de forma segura y esté disponible para todos aquellos usuarios que requieran del mismo.

Los procesos e información considerados para el desarrollo del sistema fueron: seguimiento académico, en menor grado el manejo administrativo, sin dejar de lado lo que es el manejo de datos personales de todo estudiante como el de personal de la institución educativa.

1.2. ANTECEDENTES

1.2.1. ANTECEDENTES DE LA INSTITUCIÓN

El Colegio Nacional mixto Bicentenario del Libertador Simón Bolívar realiza sus actividades en la zona Franz Tamayo de la ciudad de El Alto. Esta institución fue creada y fundada para brindar los servicios de educación al país.

Dicha unidad educativa comenzó sus actividades con una infraestructura muy pequeña, así también el número de estudiantes era pequeño, es gracias a la ayuda de organizaciones como ser: Misión Alianza de Noruega conjuntamente con la Honorable Alcaldía Municipal de El Alto es que se llegó a mejorar la infraestructura. En la actualidad cuenta con una infraestructura considerablemente grande además cuenta con una gran cantidad de estudiantes, aproximadamente 800 personas.

En la actualidad la unidad educativa realiza sus actividades en el turno tarde con los cursos de 1° grado a 6° grado de secundaria, cada uno de los grados cuenta con 4 paralelos teniendo como base un total de 24 cursos en el establecimiento.

Las funciones que cumple cada uno de los administrativos se exponen detalladamente a continuación:

Dirección.

- Inspeccionar, verificar y direccionar la eficiencia del personal que se encuentra en su cargo.
- Organizar al personal de la mejor forma para evitar contratiempos.
- Realizar informes de cada uno de los profesores y personal que se encuentran en su cargo.

Profesores:

- Realizar la planificación del plan de estudio respecto a la materia en la cual es especialista.
- Cumpliendo como consejero (asesor) de un curso del cual tendrá que realizar el llenado de las respectivas libretas.

Secretarios:

- Realizar las tareas necesarias para el llenado del libro de inscripción formularios de Registro Único del Estudiante (RUDE).
- Manipular de manera adecuada los documentos que existen en la unidad educativa.

Regencia:

- Velar de manera rigurosa la disciplina de los estudiantes así como el respeto, la asistencia al establecimiento, la puntualidad, etc.

Junta Escolar:

- Verificar que las actividades que se realizan en la unidad educativa sea en bien de los estudiantes, que ninguna actividad vaya en contra de la integridad educativa como físico de los estudiantes.

Asesores:

- Consejero de un curso en particular el cual se lo designa la dirección.

Conserje:

- Mantener la unidad educativa limpia así también tiene la responsabilidad de quien entra o sale del establecimiento.

Actualmente la unidad educativa realiza sus actividades de manera manual, apoyado con paquetes Microsoft office es por ello la gran importancia que se realice un sistema académico automatizado.

1.2.2. ANTECEDENTES DEL PROYECTO

En muchas universidades así como en la Universidad Mayor de San Andrés (U.M.S.A.) cuentan con proyectos que tratan de sistemas académicos relacionados con colegios particulares como fiscales de los cuales podemos citar y resaltar algunos.

- Proyecto educativo, académico y de gestión Colegio Nacional Mejillones, del año 2010, Es un proyecto que fue realizado en el lenguaje PHP mysql, maneja todos los hechos que se realiza en el establecimiento así como las inscripciones, registro de notas, registro del personal que se encuentra en actividad en el establecimiento; este proyecto fue desarrollado por Maria Jose Bargas Mercader como proyecto de grado.
- Software DILOGROS de nacionalidad Colombiano, se encarga del manejo educativo, es uno de los primeros que implemento la matriculación, listas para los profesores, permite realizar seguimiento en las calificaciones, fue desarrollado por la gobernación de Colombia el año 2005.

- Proyecto “Sistema de información académica I.S.E.C.A.” fue realizado por Virginia C. Mendoza el año 2009 el cual contiene un modelo relacional para poder tener en línea cualquier movimiento que realice algún miembro del establecimiento.
- Sistema de gestión Académica para la Unidad Educativa Piloto Invertida (UEPI) fue desarrollado por Edgar Vircochea, el año 2006, es un proyecto de grado el cual fue desarrollado utilizando la metodología RUP y UML, este sistema realiza la evaluación socioeconómica para los estudiantes que postulan a la Unidad Educativa.
- Gestión de Información Académica y administrativa del colegio Particular Luz a las naciones este proyecto fue desarrollado con la metodología OOHDM y Web Site QEM desarrollado por Guido Cutipa, dicho sistema realiza un control del ingreso monetario por concepto de pensiones de los estudiantes el año 2006.
- Sistema de Seguimiento Académico de Pre grado y Post Grado de la carrera de Arqueología realizado por el Lic. Franz E. Madani el año 2010 que realiza el seguimiento académico de la carrera de Arqueología así como las inscripciones registro de notas así como las publicación de las mismas.

1.3. PLANTEAMIENTO DEL PROBLEMA

Todas las unidades educativas existentes tienen como fin formar estudiantes, para ello la institución maneja grandes cantidades de información de cada estudiante, así también de profesores y otro personal de la unidad educativa, al no contar con ningún sistema computarizado, todo este manejo de información se hace de manera manual lo que conlleva a que los procesos sean lentos e ineficientes. Este sistema manual lleva a los siguientes problemas:

- La entrega de notas se realiza cada bimestre, el padre o tutor solo cuenta con un día para acceder a las notas del estudiante, pasada esa fecha debe esperar hasta el próximo bimestre para conocer las notas del estudiante.
- El registro de las notas se realiza de manera manual, para obtener el promedio se realizan cálculos manuales que conlleva la inversión de mucho tiempo de trabajo.
- El archivo está compuesto por libros que contienen el historial académico de los estudiantes, para consultar información histórica deben invertir bastante tiempo de búsqueda en los libros que componen este archivo.
- La realización de los horarios y la publicación del mismo es manual. siempre llega a demorar, razón por la que los estudiantes tienen que esperar unas semanas a tener un horario bien establecido y para acceder al horario la única manera es apersonarse a la dirección donde se publica el horario.

1.3.1. PROBLEMA PRINCIPAL

Lentitud e ineficiencia en el manejo de información en la gestión académica del colegio Nacional Mixto Bicentenario Libertador Simón Bolívar.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Diseñar e implementar un sitio web amigable para el colegio Nacional Mixto Bicentenario del Libertador Simón Bolívar el cual automatice las tareas manuales que se observaron y con ello reducir tiempo y recursos.

1.4.2. OBJETIVOS ESPECÍFICOS

- Creación de sitio Web en la cual se publiquen y actualicen constantemente las notas de cada uno de los estudiantes en los cuales cada padre o tutor podrá acceder en cualquier instante con una cuenta otorgada por la unidad educativa.
- Creación de una base de datos en la que se almacenan los datos de cada estudiante y con ello poder tener un historial con la opción de consultar cualquier dato de forma instantánea.
- Contar con un servicio SMTP para envío de notificaciones a los correos de los padres de familia sobre cualquier falta echa por su hijo o hija.
- Un módulo para la realización y creación de los horarios de cada paralelo, el cual se registre en la base de datos y se publique en el mismo sitio Web.

1.5. JUSTIFICACIÓN

1.5.1. JUSTIFICACIÓN TÉCNICA

Actualmente la tecnología llevo a revolucionar en gran magnitud, llegando a ser indispensable para cada institución tanto pública como privada. Una Institución Educativa no puede dejar de lado el uso de las herramientas tecnológicas para automatizar las tareas que se realizan cotidianamente en la institución.

La unidad educativa con apoyo de proyectos tanto de la gobernación como de la Alcaldía cuenta con equipos de computación que son capaces de soportar el Sistema de Información y Seguimiento Académico.

1.5.2. JUSTIFICACIÓN SOCIAL

La población de cada ciudad constantemente va aumentando, del mismo modo la demanda por servicios de educación llegan a ser más altos, es por ello que las instituciones educativas deben adoptar algún método para poder brindar sus servicios a numerosas de personas que por ley cuentan con derecho a la educación, la tecnología ayuda en los procesos que se realizan en una unidad educativa automatizando y mejorando la seguridad de este modo se evitan reclamos y molestias por parte de la sociedad

1.5.3. JUSTIFICACIÓN ECONÓMICA

El trabajo manual conlleva a esfuerzo y tiempo, el presente sistema de información automatiza muchos procesos realizándolo más rápido y seguro ahorrando tanto mano de obra como recursos.

1.6. ALCANCES Y LIMITES

1.6.1. ALCANCES

1.6.1.1. ALCANCE GEOGRÁFICO

- El presente trabajo posee un alcance geográfico que contempla El Colegio Nacional Mixto Bicentenario del Libertador Simón Bolívar Ubicado en la Ciudad de El Alto Zona Franz Tamayo.

1.6.1.2. ALCANCE TEMPORAL

- El alcance temporal estará establecido y planificado en el cronograma de actividades del desarrollo del proyecto de grado, en el cual se expondrá el tiempo requerido en cumplir con cada uno de los objetivos.

1.6.1.3. ALCANCE TEMÁTICO

El proyecto planteado contempla los siguientes alcances temáticos:

- Creación de sitio Web amigable en la cual se publican las notas de cada estudiante así como las noticias y eventos que existan en la misma institución.
- Para la seguridad de los usuarios, se les asignara una cuenta única que le permitirá ingresar al sistema con un tipo de usuario el cual podrá ser: administrador, padre de familia, estudiante y profesor caso que hubiese extravió de contraseña u otro inconveniente podrá recuperarlo apersonándose a dirección del colegio.
- En cuanto a los estudiantes se realizara un seguimiento académico que abarcara la inscripción de los estudiantes nuevos como antiguos, control de asistencia y puntualidad de cada estudiante, por parte de los administradores del sistema tendrán el acceso al registro de notas.
- Creación de reportes de manera rápida y sencilla con toda la información requerida ya sea el historial de un estudiante en particular u otros datos.

1.6.2. LIMITES

Con el Proyecto expuesto se logró automatizar tareas que se dan en la Institución Educativa entre las cuales podemos mencionar: inscripción de estudiantes tanto nuevos como antiguos, registro y control tanto de docentes como del personal administrativo, creación y publicación de horarios, registro de calificaciones de cada estudiante, la información de actividades que se lleva en la Unidad Educativa.

1.7. APORTES

El sistema expuesto anteriormente será principalmente el aporte esencial puesto que ayudara en las tareas tanto académicas como administrativas que se realizan cotidianamente en El Colegio Nacional Mixto Bicentenario del Libertador Simón Bolívar, llegando a simplificar muchos procesos molestos y tardíos, pero principalmente aumentando la calidad y prestigio del Instituto Educativo.

1.8. METODOLOGÍA

Para desarrollar el sistema mencionado anteriormente, se hizo uso de la Metodología de Diseño de Hipertextos Orientada a Objetos (OOHDM) este modelo es útil para construir grandes aplicaciones de hipermedia, como ser: sitios web y sistemas de información y comunicación.

2. MARCO TEÓRICO

MARCO CONCEPTUAL

2.1. TECNOLOGÍA WEB

La tecnología web en la actualidad ha llegado a convertirse en una parte muy importante y fundamental en el mundo de la tecnología de la información, es capaz de interactuar con todas las tecnologías de inter conectividad de ordenadores y servidores que permite a los usuarios el intercambio de todo tipo de datos de información como ser textos, imágenes, sonidos los cuales con el tiempo van creciendo es por ello que se hace muy complicado el mantenimiento como también el manejo de datos de un sistema, siendo de este modo la principal característica para el surgimiento de metodologías de diseño web las cuales nos ayudan a seguir ciertos procedimientos que nos ayudan a ordenar todas las actividades.

2.1.1. SISTEMAS DE INFORMACIÓN BASADA EN LA WEB

Según Schmuller (Schmuller, 2001) los sistemas de información con el tiempo han ido evolucionando, emigrando constantemente en distintos entornos como ser la web, con la tecnología *World Wide Web* un sistema de información es capaz de emplear recomendaciones técnicas y tecnológicas muy robustas que a la vez ofrecen un gran rendimiento y fiabilidad, siendo al mismo tiempo muy sencillo la navegación para el usuario final. La manipulación y el diseño del mismo se los realiza con herramientas ampliamente conocidos, como son los navegadores o los métodos de búsqueda y navegación.

Para un estudio más amplio acerca de los sistemas de información basados en la web tocaremos las siguientes características fundamentales de este tipo de sistemas.

- **Arquitectura.** El hecho de que la información se mueve a través de internet y en ella misma se implementa sus servicios, estos servicios de sistemas informáticos cuentan con una arquitectura común, al contrario de lo que sucede con otras soluciones creadas específicamente para trabajar en sistemas aislados. Esto favorece que se puedan retroalimentar unas aplicaciones de otras, sin que necesariamente pertenezcan a la misma empresa o, incluso, sin que hayan sido diseñadas para realizar las mismas funciones. Además se facilita la migración de datos de una aplicación a otra, así como la interconexión de la información procedente de diferentes fuentes, pero que empleen la misma arquitectura (Schmuller, 2001).
- **Evolución y desarrollo.** Al estar dentro de un entorno tan dinámico y cambiante, la mayoría de estos sistemas tienen la necesidad de estar en constante evolución, presentando continuas versiones que actualizan las funciones y posibilidades de las anteriores (Schmuller, 2001).
- **El usuario por encima del sistema.** El usuario del entorno Web se ha acostumbrado a recibir la información de una manera concreta y a navegar empleando una serie de parámetros basadas en su experiencia. Es lógico pensar que un sistema de información que emplee este entorno crezca también de esas experiencias para trasladarlas a su propuesta de aplicación Web. Primero porque el usuario se las exigirá, y segundo porque no hacerlo sería desaprovechar años de experiencia en el desarrollo de aplicaciones (Schmuller, 2001).

- Escalabilidad. Es conocida como la habilidad que tiene un sistema para reaccionar a las necesidades cambiantes tanto del usuario como del responsable del servicio y la información que este proporciona, la capacidad para adaptarse creciendo de manera continuada y fluida. La capacidad que tienen los sistemas de información en la Web es que pueden crecer tanto como quiera, puesto que el contexto en el que se mueven, la propia Web, se basa en el principio de ser cada vez un sistema más grande (Schmuller, 2001).

2.1.1.1. SITIO WEB

Un sitio web es un espacio documental organizado que en todos los casos está dedicado a un tema en particular o un propósito común, un sitio web puede contener múltiples enlaces o referencias a cualquier otro sitio web ya sea del mismo entorno o un entorno totalmente diferente, es por ello que la navegación de un sitio web hacia otra puede llevar hacia un ciclo interminable, para el usuario final es casi borrosa la distinción de dichos sitios web.

Según Smuller (Schmuller, 2001) un sitio web es un conjunto de archivos electrónicos y páginas web referentes al tema principal el cual se toma, incluye una página inicial de bienvenida, generalmente denominada con `index` con un nombre definido por el desarrollador.

Un sitio web dinámico es uno que puede tener cambios frecuentes en la información. Cuando el servidor web recibe una petición para una determinada página de un sitio web, la página se genera automáticamente por el software, como respuesta directa a la petición de la página.

2.1.1.2. PAGINA WEB

Todas las páginas web se encuentran almacenadas en un servidor web el cual puede ser local (en la propia maquina) como también puede estar almacenada remotamente, se puede

tener acceso a estas páginas, utilizando el protocolo de transferencia de hipertextos como ser (HTTP) (Schmuller, 2001).

Según Schmuller (Schmuller, 2001) una página web también conocida como página electrónica o ciberpagina es un documento o información electrónica es capaz de contener texto, sonido, video, programas, enlaces, imágenes y muchas más cosas, todas estas páginas se adaptan a la llamada Word Wide Web el cual puede llegar a ser accedida mediante un navegador web. Esta información que maneja el navegador puede estar en formato HTML o XHTML la cual está a su vez puede tener enlaces de hipertextos el cual puede proporcionar acceso a otras páginas web, en algunos casos también contienen a su vez hojas de estilo en cascada, scripts, imágenes y otros elementos.

2.2. METODOLOGÍA DE DISEÑO DE HIPERTEXTOS ORIENTADO A OBJETOS (OOHDM)

En los todos estos años se han creado muchos métodos para poder desarrollar sitios web, entre ellas está el modelo OOHDM Metodología de diseño de Hipertextos Orientado a Objetos, este es un método para el diseño de aplicaciones hipermedia especialmente para la web, se podría decir por otra parte que es una evolución de la metodología conocida como Metodología de Diseño de Hipertextos HDM con la característica que ahora este método será orientada a objetos además que en la actualidad este método conjuntamente con otros se a convertido en una de las metodologías más utilizadas en muchas aplicaciones hipermedia como ser sitios web.

Según Roger Pressman (Pressman, 2010) la metodología OOHDM es una metodología de diseño de hipermedia, que utiliza el enfoque orientado a objetos extendiéndolo e integrándolo con técnicas de representación gráfica de relaciones entre objetos y de

contextos navegacionales que son ricos en representación estructural y semántica este método contiene un proceso que conllevan hacia 5 fases en el cual se combinan notaciones UML como otras propias de la metodología las cuales se describen a continuación.

2.2.1. OBTENCIÓN DE REQUERIMIENTOS

Para comenzar con este método el primer paso es la obtención de requerimientos para ellos primeramente se debe identificar los actores que intervendrán en el sistema seguidamente se debe identificar cada uno de las funciones que puede realizar un actor, para plasmar de forma gráfica toda la identificación de requerimiento de datos se hará uso de los Diagramas de casos de uso como se observa un ejemplo en la figura 1, seguidamente se podrá realizar un diagrama de iteraciones por cada actor Estos diagramas proveen una representación concisa utilizando una metáfora gráfica del flujo de información entre el usuario y la aplicación durante la ejecución de cualquier tarea (Pressman, 2010).

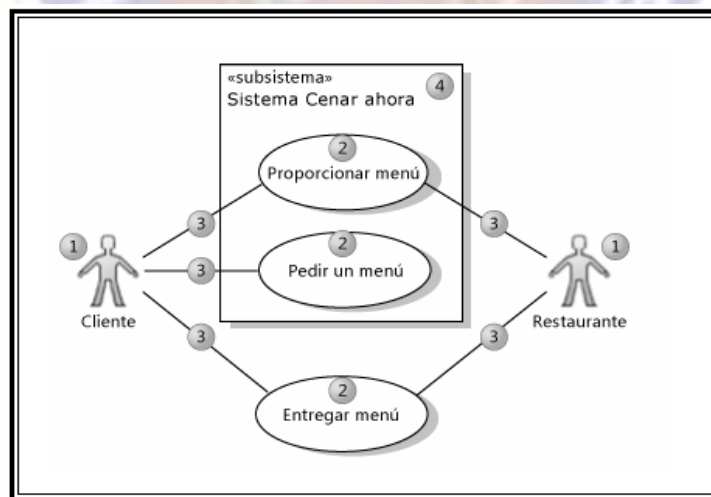


Figura 1. Caso de uso
Fuente: (Lapiente, 2016)

2.2.2. DISEÑO CONCEPTUAL

El diseño conceptual siendo la segunda fase de la Metodología de Diseño de Hipermedia Orientada a Objetos es la fase en la cual se desarrolla un modelo del sistema orientado a objetos el cual representa el sistema con finalidad de poder plasmar de forma gráfica toda la funcionalidad del sistema.

Según Roger Pressman (Pressman, 2010) el diseño conceptual del OOHDM genera una representación de los subsistemas o clases con relaciones que define las aplicaciones para la webapp, para el desarrollo de esta fase se puede hacer uso de UML para la creación de diagramas de clase apropiados, agregaciones y representaciones compuestas de las clases, también un diagrama de colaboración puede ser muy útil u otra clase de información que describa el dominio de la aplicación, así mismo en esta segunda fase se obtienen diferentes tipos de resultados los cuales se plantean a continuación.

- Productos: Diagrama de Clases en la figura dos observamos un ejemplo sobre el diseño de diagramas de clases, División en subsistemas y relaciones.
- Herramientas: Técnicas de modelado O.O, patrones de diseño.
- Mecanismos: Clasificación, agregación, generalización y especialización.
- Objetivo de diseño: Modelo semántico de la aplicación.

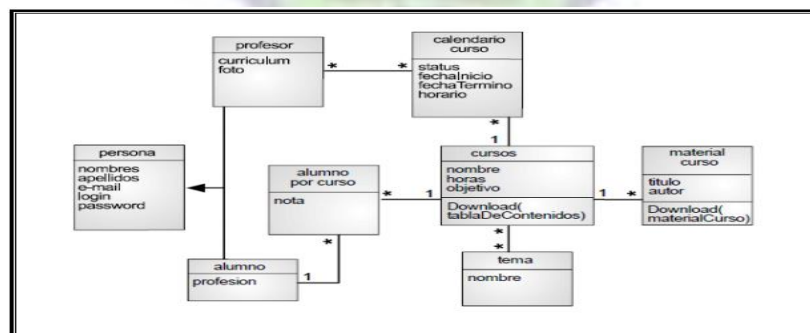


Figura 2. Sintaxis diagrama de clases

Fuente: (Sanchez, 2017)

2.2.3. DISEÑO NAVEGACIONAL

El Diseño Navegacional como la tercera fase de la Metodología de Diseño de Hipermedia Orientada a Objetos es la etapa en la cual se realiza una estructura de navegación la cual contendrá la aplicación todo a través de un hiperdocumento en la cual se realiza por medio de modelos navegacionales que a su vez representaran las vistas que se pueden obtener en el diseño conceptual las cuales se definió en la fase anterior.

Según Roger Pressman (Pressman, 2010) el diseño de navegación identifica un conjunto de objetos que se derivan de las clases definidas en el diseño conceptual, Para incluir estos se define una serie de clases conocidos como clases de navegación o nodos, en esta tercera fase también se hará uso de UML con las herramientas de tablas de estado y diagramas de secuencia apropiados estos diagramas son muy necesarios para poder comprender los requerimientos de la navegación y conforme a ello para el desarrollo del diseño se utilizan patrones de navegación el modelo OOHDMM por inclusión hace uso de lo que es nodos, vínculos, anclas y estructuras de acceso todas estas son más elaboradas también incluyen mecanismos tales como un menú de la webapp, puede ser también conocido como mapa del sitio o recorrido guiado, una vez definidas las clases de navegación existirá un contexto con la descripción de la estructura de navegación local la restricción impuesta al acceso de los objetos de contenido y los métodos requeridos para tener el acceso a los objetos de contenido, en ello existen algunos esquemas los cuales se mencionan a continuación.

- Esquema de las clases navegacionales. Estas deben mostrar las posibles vistas de un hiperdocumento. Las vistas se representan a su vez con unos tipos predefinidos como nodos o enlaces, además de otras clases que expresan estructuras o formas alternativas de acceso a nodos, como índices o recorridos guiados.
- Esquema de contexto navegacional. Estas permiten la estructuración de hiperespacio de navegación entre sub espacios, en ellos se mostrara la información

que se requiere mostrar al usuario, y los enlaces que estarán disponibles cuando se acceda a un objeto o nodo de un contexto determinado.

En la figura 3 se observa un ejemplo de un esquema navegacional con las características primordiales que se debe obtener en un esquema navegacional.

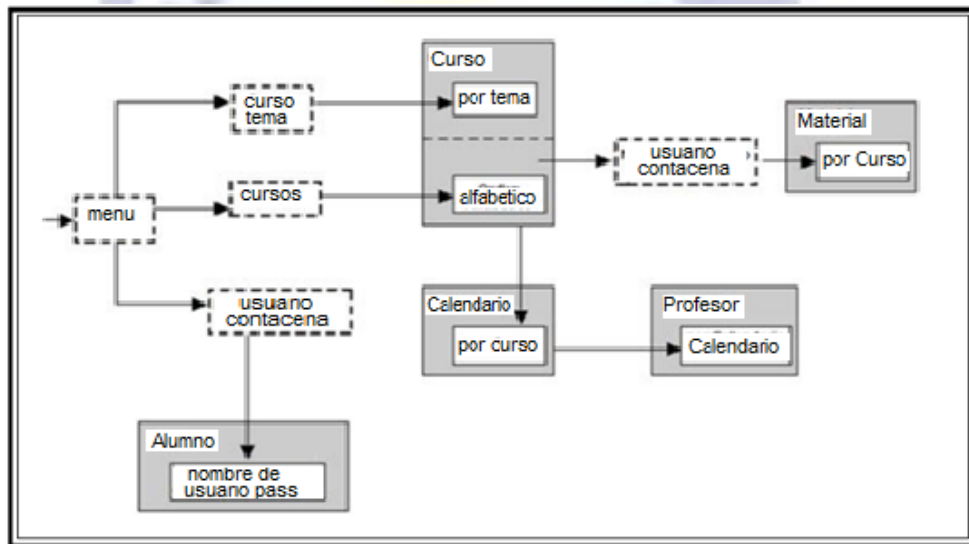


Figura 3. Descripción esquema navegacional
Fuente: (Colombia M. d., 2011)

2.2.4. DISEÑO DE INTERFAZ ABSTRACTA

Esta fase está dedicada a la especialización de una interfaz abstracta. De esta manera, se puede especificar la forma en la que aparecen los contextos navegacionales los cuales se definieron en la fase anterior así mismo en esta fase se incluyen el modo en la cual la interfaz puede activar otras interfaces.

Según Roger Pressman (Pressman, 2010) la acción e diseño abstracto de la interfaz especifica de los objetos de la interfaz ve cuando ocurre una interacción con la webapp,

para realizar el diseño abstracto se emplea un modelo formal de objetos como pueden ser nombre descripción o al mismo tiempo pueden llegar a ser métodos definidos como ser obtenernombre() entre otros de esa manera se obtendrá una manera visual de un esquema en el cual se plasma la selección que un cliente puede acceder, también puede ser definida como la vista de datos abstractos para representar la relación entre objetos de interfaz y de navegación así como las características de comportamiento de los objetos de interfaz, también se menciona el modelo VDA la cual define una plantilla estática que representara la metáfora de la interfaz e incluye una representación de los objetos de navegación dentro de la interfaz y la especificación de los objetos, la Figura 4 muestra gráficamente la elaboración del diseño conceptual.

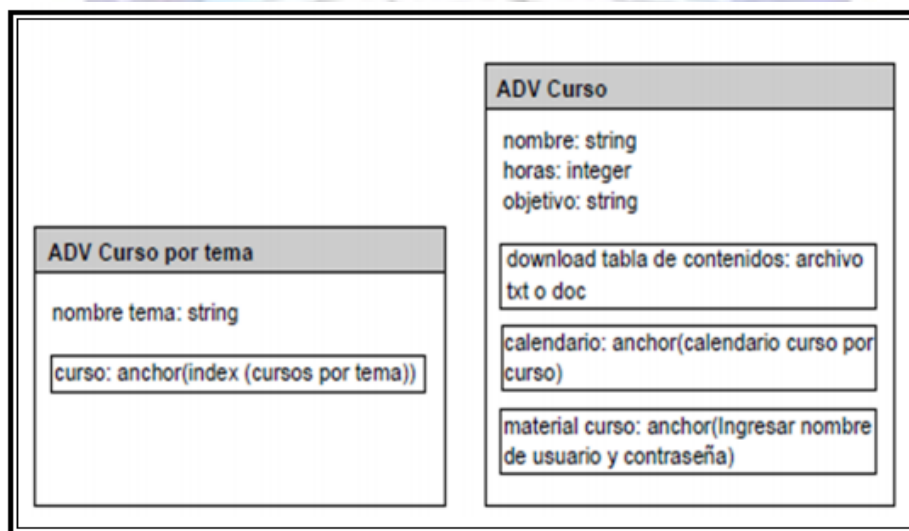


Figura 4. Diseño interfaz abstracta

Fuente: (Lapuente, 2016)

2.2.5. IMPLEMENTACIÓN

La implementación consiste en mapear los objetos del Modelo de Navegación e Interfaz a una aplicación final y puede requerir elaborar arquitecturas (por ejemplo, Cliente-Servidor) en las cuales las aplicaciones son clientes son Browsers Web para compartir el servidor de base de datos que contiene los objetos del modelo conceptual. Un número de aplicaciones

basadas en sistemas de información como también aplicaciones Web han sido desarrollados utilizando OOADM como metodología y empleando una diversidad de tecnologías tal como Java (J2EE), ASPX (.NET), Lua (CGILua) y Ruby (Ruby on Rails), Nodejs, Angular para llevar a cabo la implementación.

2.3. DISEÑO ARQUITECTÓNICO

Una Arquitectura de Software, también conocido como Arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan un marco definido y claro para interactuar con el código fuente del software.

Como mencionamos anteriormente en el desarrollo del proyecto se harán uso de ciertas tecnologías como lo que es angular como nodejs los cuales manejan la arquitectura de modelo vista controlador (MVC) esa será la arquitectura que se maneja en este proyecto

2.3.1. MODELO VISTA CONTROLADOR (MVC)

Según Ángel Arias y Alicia Durango (Universidad de Alicante, 2017) Antiguamente la arquitectura MVC fue desarrollado para aplicaciones de escritorio, ha sido ampliamente adaptado como arquitectura para diseñar e implementar aplicaciones web en los principales lenguajes de programación. Se han desarrollado multitud de frameworks, comerciales y no comerciales, que implementan este patrón de desarrollo; la arquitectura (MVC) básicamente se divide en dos partes que son el cliente servidor los cuales se integran en las aplicaciones web ya que de esa manera es menos robusta y más rápida a continuación se describen los conceptos de cada elemento de esta arquitectura.

- El Modelo. Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan

descrito en las especificaciones de la aplicación. Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

- El Controlador. Responde a eventos que normalmente los hace el usuario e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información como podría ser una edición de documento u otro. También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo' por diferentes registros que la base de datos demuestra, por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.
- La Vista. Presenta el 'modelo' la información necesaria del negocio en un formato adecuado para interactuar con el usuario de una manera sencilla, por tanto requiere de dicho 'modelo' la información que debe representar como salida.

2.3.2. ARQUITECTURA REST

Según (Wikipedia Enciclopedia, 2018) la arquitectura REST (Representational State Transfer) o en español representación de estado de transferencia es un estilo de diseños de arquitecturas de software para sistemas hipermedias distribuidos tales como la Web. El término REST apareció en el año 2000 fue creada por Roy Fielding, quien es uno de los principales autores de la especificación de HTTP.

En realidad, REST se refiere estrictamente a una colección de principios para el diseño de arquitecturas en red. Estos principios resumen como los recursos son definidos y diseccionados. Es posible diseñar un sistema software de gran tamaño de acuerdo con la arquitectura propuesta por Fielding sin utilizar HTTP o sin interactuar con la Web. Así

como también es posible diseñar una simple interfaz XML+HTTP que no sigue los principios REST, y en cambio seguir algún otro modelo.

Si pensamos un poco en este éxito, nos daremos cuenta que la Web ha sido la única aplicación distribuida que ha conseguido ser escalable al tamaño de Internet, el concepto central de la Web es un espacio de URIs unificado. Las URIs permiten la densa red de enlaces que permiten a la Web que sea tan utilizada. Por tanto, ellos consiguen tejer una mega-aplicación.

Los principales objetivos de este estilo de arquitectura son:

- Escalabilidad de la interacción con los componentes. La Web ha crecido exponencialmente sin degradar su rendimiento. Una prueba de ellos es la variedad de clientes que pueden acceder a través de la Web: estaciones de trabajo, sistemas industriales, dispositivos móviles.
- Generalidad de interfaces. Gracias al protocolo HTTP, cualquier cliente puede interactuar con cualquier servidor HTTP sin ninguna configuración especial.

Hipermedia como un mecanismo del estado de la aplicación. El estado actual de una aplicación Web debería ser capturado en uno o más documentos de hipertexto, residiendo tanto en el cliente como en el servidor. El servidor conoce sobre el estado de sus recursos, aunque no intenta seguirle la pista a las sesiones individuales de los clientes. Esta es la misión del navegador, el sabe cómo navegar de recurso a recurso, recogiendo información que el necesita o cambiar el estado que el necesita cambiar.

2.4. STACK PEVN

El STACK PEVN es una combinación de cuatro tecnologías de programación, en lenguaje JavaScript, las cuales son Nodejs, Postgresql, Vuejs, Express y Npm tecnologías las cuales se definen a continuación:

- Nodejs: **es un entorno JavaScript del lado del servidor, basado en eventos,** nodejs ejecuta JavaScript utilizando el motor V8, fue desarrollado por la una de las empresas más grandes Google para uso de su navegador Chrome. **Aprovechando el motor V8 permite a Nodejs proporcionar un entorno de ejecución del lado del servidor que compila y ejecuta JavaScript a velocidades increíbles.** El aumento de velocidad es importante debido a que V8 compila JavaScript en código de máquina nativo, en lugar de interpretarlo o ejecutarlo. Nodejs es de código abierto, y se ejecuta tanto en Windows como en Linux (Google, 2018).
- PostgreSQL: Es un Sistema de gestión de bases de datos relacional con un complemento que nos permite trabajar con una base de datos no relacional también es orientado a objetos y lo mas sobresaliente es que es opensource de código abierto, publicado bajo la licencia PostgreSQL, como muchos otros proyectos es de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyados por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group, 1997).
- ExpressJS: Express es un framework web transigente, escrito en JavaScript y alojado dentro del entorno de ejecución NodeJS. El modulo explica algunos de los beneficios clave de este framework, como configurar tu entorno de desarrollo y como realizar tareas comunes en desarrollo y publicación web (Express - Infraestructura de aplicaciones web Node.js, 2018).
- VueJs: Es un framework para aplicaciones web de TypeScript de código abierto, mantenido por grupo de desarrolladores sin lucro, que se utiliza para crear y

mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC) el cual se hace uso en este proyecto, con ello es que es un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles (Grupo desarrollo open sour, 2017).

- JavaScript: Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente dinámico. Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio es también significativo (Brendan Eich, 2018).
- NPM A lo largo del desarrollo de una aplicación con Nodejs, nos veremos obligados a utilizar módulos de terceros debido a que el propósito principal de éste es utilizar JavaScript del lado del servidor y delegar otras funcionalidades en su principio de modularidad. La instalación de nuevos módulos con Nodejs se realiza de manera muy sencilla con la herramienta npm (Node Package Manager). Esta herramienta es la encargada de añadir los nuevos módulos que necesitemos (open source, 2017).

Un beneficio muy grande de usar este stack es que al usar JavaScript en ambas partes de la aplicación, los tiempos de desarrollo de la aplicación mejoran notablemente, ya que los desarrolladores pueden trabajar tanto en el back-end como en el front-end sin necesidad de tener que especializarse en una única parte.

Otro detalle importante, es el almacenamiento de la información en formato JSON. Esto nos permite enviar datos en las peticiones con la estructura que va a persistir en los datos.

2.5. CALIDAD DE SOFTWARE

En el proceso de desarrollo de software viene a la mente de cualquier desarrollador de software el tema de calidad como algo muy importante, la calidad es un tema fundamental en cualquier proyecto ya que gracias a este proceso que se aplica se puede crear un producto útil que proporciona un valor medible a los usuarios finales así como a los creadores del producto, la calidad de software con el tiempo fue creciendo y adoptándose en todos los proyectos grandes como pequeños con este crecimiento es que se crean diversos métodos de evaluación de calidad de software es por ello que se crea un estándar de todos los métodos existentes el estándar más conocido es la ISO-9126 (Pressman, 2010).

2.5.1. CALIDAD DE SOFTWARE CON ISO-9126

Según Migani (Universidad de Alicante, 2017) la norma ISO-9126 llega a ser un estándar en el proceso de evaluación de software en los años 1991, la primera presentación de dicha norma se la presenta primeramente como un esquema para la evaluación de calidad de software, esta normativa como cualquier otra normativa cuenta con sus características las cuales a su vez cada característica se divide en sub características, la normativa 9126 es representada como un modelo detallado para la evaluación de cualquier sistema informático, a continuación se describen las características propias de este estándar.

- **Funcionalidad.** Refiere en la capacidad que tiene el software para proveer los servicios necesarios para cumplir con los requisitos funcionales que se determinaron. Este a su vez se sub caracteriza por ser idóneo, nos demuestra si el software desempeñara las tareas por las cuales fue desarrollado, también es importante mencionar que es una normativa con exactitud la cual evalúa el resultado final que obtiene el software y se obtiene consistencia de lo que se obtiene del producto final, nos da la ventaja de ser interoperable lo cual consiste en revisar si el

sistema puede interactuar con otro sistema independiente y es seguro cuando se verifica si el sistema puede impedir el acceso a personal no autorizado.

- **Fiabilidad.** La capacidad del software de mantener las prestaciones requeridas del sistema, durante un tiempo establecido y bajo un conjunto de condiciones definidas, una ventaja de la normativa es que muy completo ya que verifica las fallas del sistema y gracias a ella es que se pueden llegar a eliminar muchas fallas que presente durante la prueba, otra ventana y no menos importante la recuperabilidad que es la capacidad de verificar si el software puede resumir el funcionamiento y restaurar datos perdidos luego de haber ocurrido algún fallo ocasional.
- **Usabilidad.** El usuario final es una de las partes más importantes a la hora de desarrollar un software es por ello que una característica de la normativa 9126 es la usabilidad el cual refiere al esfuerzo requerido por el usuario a la hora de usar el producto, también hace referencia a la dificultad que pueda existir al aprender el uso del sistema, la comprensión nos determina que tal sencillo puede llegar a ser para el usuario la comprensión del funcionamiento del sistema.
- **Eficiente.** Relación entre las presentaciones del software y los requisitos necesarios para su utilización, el comportamiento en el tiempo verifica la rapidez en que responde el sistema, comportamiento de recurso es el que determina si el sistema utiliza los recursos de manera eficiente.
- **Mantenibilidad.** Se lo denominaría como el esfuerzo necesario para adaptarse a las nuevas especificaciones y requisitos del software, cabe mencionar que también evalúa la estabilidad es la que verifica si el sistema puede mantener su funcionamiento a pesar de realizar cambios, la factibilidad de análisis determina si la estructura de desarrollo es funcional con el objetivo de diagnosticar fácilmente las

fallas, la factibilidad de cambio verificara si el sistema puede ser fácilmente modificado.

- Portabilidad. La capacidad del software de ser transferido de un entorno a otro, capacidad de instalación verifica si el software se puede instalar fácilmente, reemplazo es la capacidad de que el software planteado puede reemplazar a otro con características similares.

2.5.2. ESTIMACIÓN DEL TAMAÑO POR PUNTOS FUNCIÓN

Según Migani (Universidad de Alicante, 2017) los puntos de función PF, son una métrica para establecer el tamaño y complejidad de los sistemas informáticos basada en la calidad de funcionalidad requerida y entregada a los usuarios, miden el tamaño lógico o funcional de las aplicaciones software basado en los requisitos funcionales del usuario. A partir de su aparición en 1979 han surgido una serie de diversas propuestas para calcular el tamaño funcional de un proyecto software. Los métodos para la estimación de los PF deben ajustarse a la norma ISO/IEC 14143, cualquiera de los métodos que se ajuste a este estándar puede ser usado para la estimación del tamaño funcional. FP Lite es un método de estimación paramétrico derivado del método de análisis de punto función FPA definido por el Grupo Internacional de Usuarios de Puntos Función IFPUG. Fue creado por la empresa David Consulting dicho método cuenta con elementos imprescindibles a la hora de realizar los cálculos de la métrica dichos elementos funcionales son:

- Datos-ILF. Ficheros Lógicos Internos, es un grupo de datos relacionados lógicamente o información de control, identificable por el usuario referenciado por la aplicación y almacenada en los límites de la aplicación. El objetivo principal es almacenar datos referenciados mediante uno o más procesos elementales,

típicamente se corresponden con las entidades que aparecen en un diagrama Entidad-Relación.

- Interfaces-ELF. Ficheros Lógicos Externos, es un grupo de datos relacionados lógicamente o información de control, identificable por el usuario pero mantenido dentro de los límites de otra aplicación, cuyo objetivo principal es almacenar datos dentro de los límites de otra aplicación.
- Entradas-EI. Entradas Externas, proceso elemental que procesa datos o información de control que proviene de fuera de los límites de la aplicación los datos pueden venir desde otra aplicación de una pantalla de ingreso de datos, el objetivo fundamental es mantener uno o más ILF.
- Salidas-EO: Salidas Externas, proceso elemental que envía datos o información de control hacia fuera de los límites de la aplicación el objetivo fundamental es presentar información al usuario mediante un proceso lógico diferente o adicional al de la recuperación de datos.
- Consultas-EQ: Consultas Externas, proceso elemental que envía datos o información de control fuera de los límites de la aplicación cuyo principal objetivo es presentar información al usuario a través de la recuperación de datos. A causa de los problemas para la evaluación de la complejidad de los elementos funcionales se propuso el método PF Lite, en el que se considera que todos los elementos funcionales tiene una complejidad media para de esta manera simplificar en tres pasos la metodología FPA.

A causa de los problemas para la evaluación de la complejidad de los elementos funcionales se propuso el método PF Lite, en el que se considera que todos los elementos funcionales tiene una complejidad media para de esta manera simplificar en tres pasos la metodología FPA, pasos que se describen en la Figura 5.

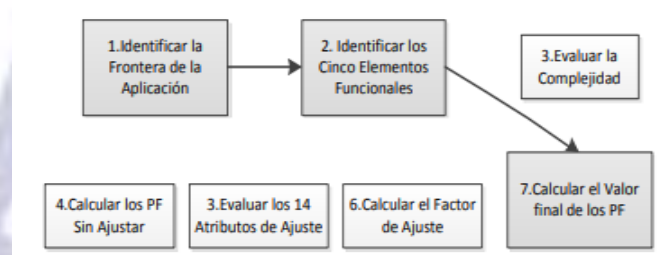


Figura 5 Proceso para el cálculo del PF

Fuente: (Universidad de Alicante, 2017)

El valor final de los puntos función para proyectos en desarrollo está dado por la siguiente formula:

$$PF = \sum_{K=1}^n Ei * Ci \quad E(1)$$

Siendo:

Ei cada elemento funcional (EI,EQ,EO,ELF,ELF).

Ci la complejidad media considerada para cada uno de los tipos de elementos.

3. MARCO APLICATIVO

3.1. ANÁLISIS DE LA INSTITUCIÓN

3.1.1. ORGANIZACIÓN DEL MODELO AMBIENTAL

El colegio Nacional Mixto Bicentenario Libertador Simón Bolívar en la actualidad cuenta con un secretario el cual realiza el llenado de formulario de registro de estudiantes así también realiza el desarrollo de informes bimestrales y anuales de dicha Unidad Educativa, mismo que será el principal usuario para el llenado de formulario control de reportes y otras funciones que el sistema académico propuesto facilitara a la Unidad educativa.

- Declaración de propósitos: El Sistema de Información y Seguimiento Académico Colegio Nacional Mixto Bicentenario Libertador Simón Bolívar tiene como propósito ser de ayuda en la Unidad Educativa mencionada, el cual colaborara en ciertas funciones como recopilar, almacenar, procesar y difundir la información tanto de estudiantes como de docentes. Se realizaran reportes los cuales se generan en la base a los datos que exista en el sistema los cuales serán datos resultantes de muchos acontecimientos que se realizaron en la misma educación educativa.
- Dicho sistema no se podrá apoyar con los recursos computacionales existentes en dicha Unidad Educativa ya que no se cuenta con ciertas características necesarias para poder acoplar al sistema propuesto, es por ello que se harán uso de los recursos de una institución externa a la institución educativa la cual cuenta con los recursos necesarios dicha institución será la Digital Ocean.

3.1.2. DESCRIPCIÓN ROLES EN EL SISTEMA

Cada usuario tiene un rol en el sistema estos roles definen las funciones que poseen, dichos roles se los describe a continuación:

- **Director**
 - Informes para la toma de decisiones.
 - Acceso a todas las funciones.
- **Secretario**
 - Llenado de nuevos datos como actualización de datos (inscripciones, llenado de formulario de nuevo usuario, eliminación y actualización de datos).
 - Informes de seguimiento académico
 - Generación de informes.
- **Docentes**
 - Visualización de lista de estudiantes por cada paralelos.
 - Visualización de estadísticas de promedios.
 - llenado de observación de estudiantes.
- **Padres de familia**
 - reporte de las notas bimestrales como anuales de sus hijos.
 - obtención de notificaciones en tiempo real de observaciones hechas hacia sus hijos
- **Estudiantes**
 - reporte de las notas bimestrales como anuales.

- acceder al chat del sistema con la participación de todos los usuarios del colegio.

3.1.3. DIAGRAMA DE CONTEXTO DEL SISTEMA

El Sistema de Información y Seguimiento Académico, se encuentra envuelto en un ambiente compuesto por: docentes, director, secretario, estudiantes y padres de familia, en la figura 6 se visualiza el diagrama de contexto del sistema.

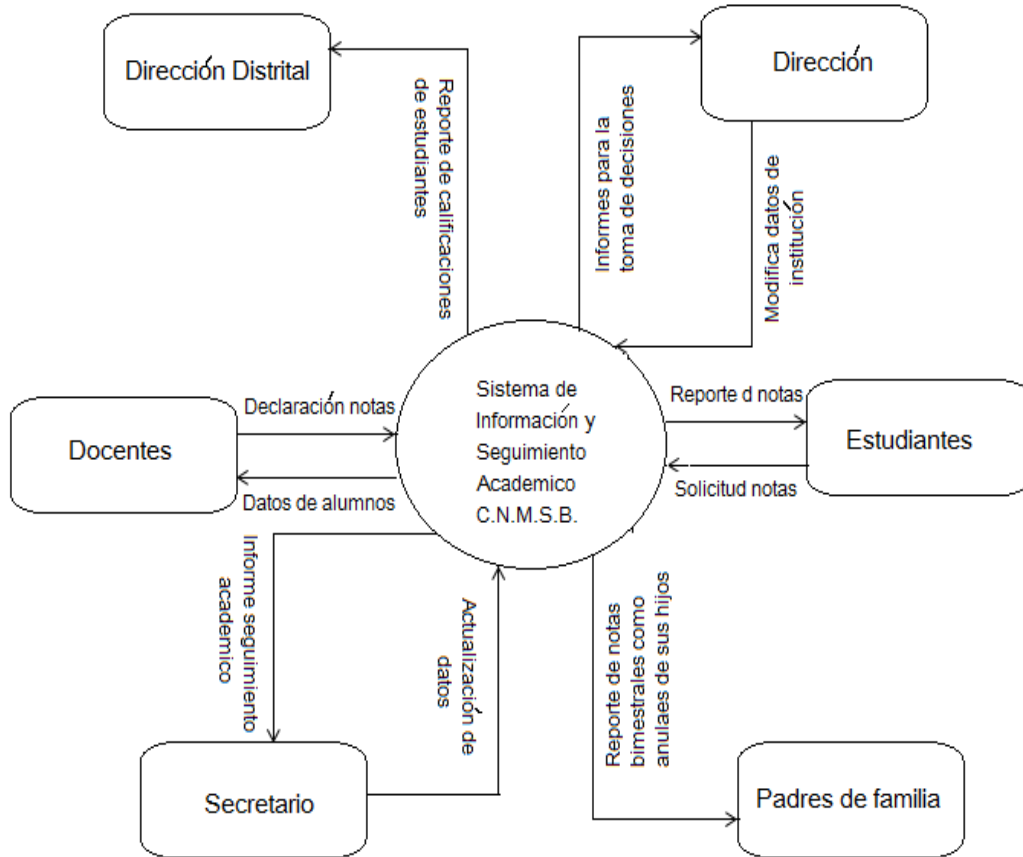


Figura 6. Diagrama de contexto del sistema

3.2. METODOLOGÍA DE DISEÑO DE HIPERTEXTOS ORIENTADO A OBJETOS (OOHDM)

Como se mencionó anteriormente para el desarrollo del proyecto planteado se hará uso de la Metodología de Diseño de Hipertextos Orientada a Objetos la cual a su vez consta de 5 fases las cuales se irán desarrollando una tras otra por el hecho de que estas fases se las realizan en forma ordenada, de esta manera se realizara el estudio necesario para la obtención de requerimientos las cuales demandan cada fase de la metodología a continuación exponemos el desarrollo de las fases de la metodología.

3.2.1. OBTENCIÓN DE REQUERIMIENTOS

De acuerdo con las problemáticas del manejo de la información que mencionamos anteriormente y con ayuda de entrevistas que se realizaron en el establecimiento educativo para ver que necesidades se tiene por los futuros usuarios del sistema, se pudo determinar ciertas deficiencias que existen en el establecimiento educativo como ser ineficiencia y lentitud en el proceso de inscripción, publicación de notas bimestrales, publicación de horarios, etc. Con todos estos requerimientos encontrados se pudo observar que en el establecimiento educativo existen muchos procesos lentos y nada eficientes debido al tratamiento y calculo manual que se realiza en muchos de los procesos actualmente, conllevando al mal desenvolvimiento en las funciones que se realiza en el establecimiento como también en manejo de información.

A continuación realizamos la exposición de los diagramas gráficos necesarios para poder cumplir con los requerimientos que se tiene en la institución.

La figura 7 muestra el comportamiento del proceso de inscripción que en la actualidad se realiza en la unidad educativa, se detalla en un diagrama de secuencias.

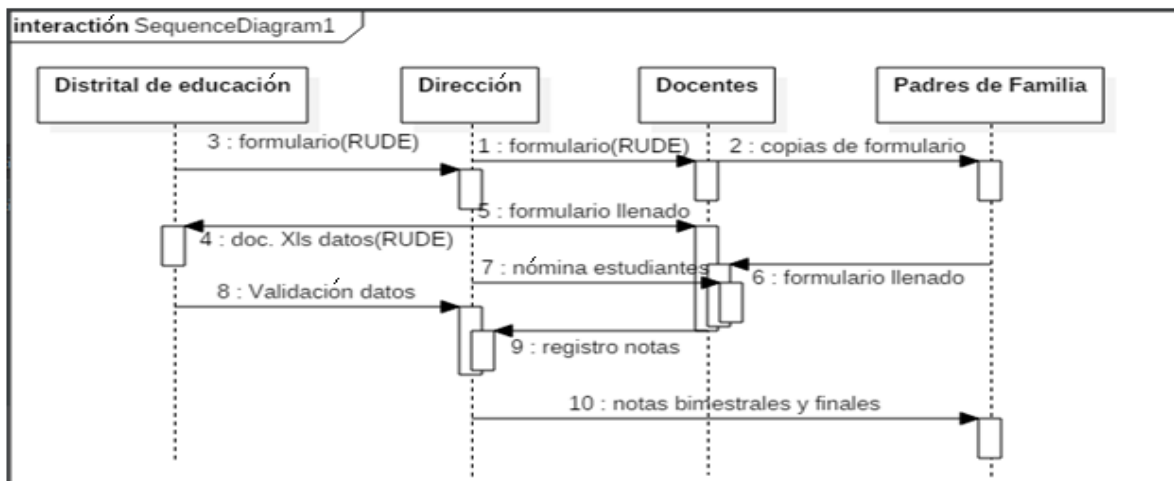


Figura 7. Diagrama de Secuencia Proceso de Inscripción Actual

La información del establecimiento nace a través de formularios de inscripción así como el de actualización, el formulario valido para este proceso es el formulario de registro único de estudiantes (RUDE), el cual es proporcionado por el Ministerio de Educación realizando a su vez el estudio y aprobación necesario del formulario, dicho formulario nos ayuda a recabar la información necesaria de cada estudiante, así como ser unidad actual como unidad anterior si hubiese realizado un cambio de unidad, aspectos sociales, datos de sus familiares cercanos como ser el padre madre, acceso a los servicios básicos, accesibilidad al servicio de transporte, idiomas que domina, si cuenta con seguro médico, etc.

Con el llenado del formulario el siguiente paso es registrarlo en los datos del colegio para luego realizar una transcripción del formulario en una hoja Excel u otro formato luego de ello el siguiente paso es imprimir dicho formulario adjuntar algunos otros documentos del estudiante para luego llevarlos a la distrital de educación correspondiente, el cual verificara los datos de dicho estudiante para luego aprobar y validar la información enviada.

Realizando un pequeño análisis de este procedimiento manual que se realiza en el establecimiento vemos que el tiempo que se otorga a este procedimiento es mucho ya que al llenar los datos en una hoja Excel es hacer dos veces el mismo trabajo llenado a mano y llenado digital en algunos casos son los docentes los que colaboran en este proceso.

Como mencionamos al principio del texto el Colegio Nacional Mixto Bicentenario del Libertador Simón Bolívar no cuenta con un sistema para poder centralizar toda la información acerca de sus estudiantes así como de los docentes que les facilite la integración y manejo de la información.

3.2.1.1.REQUERIMIENTOS FUNCIONALES

Los requerimientos funcionales identificados se observan en la Tabla 1.

Tabla 1. Requerimientos funcionales

| Lista de requerimientos Funcionales | | |
|-------------------------------------|--|---------------------------------|
| Código | Requerimiento | Caso de uso |
| RF1 | El sitio web debe controlar el acceso al sistema a través de un usuario y password. | Caso de uso inicio de sesión |
| RF2 | El administrador debe poder realizar la inscripción. | Caso de uso inscripción |
| RF3 | El sistema debe poder desplegar informes. | Caso de uso informes |
| RF4 | El sistema debe poder registrar notas de materia | Caso de uso notas |
| RF5 | El sistema debe poder registrar y notificar observaciones hecha por mala conducta de estudiantes | Caso de uso observaciones |
| RF6 | El sistema debe registrar nuevos usuarios con el rol específico como ser administrador profesor o estudiante | Caso de uso registro usuarios |
| RF7 | El sistema debe tener la capacidad de organizar un horario | Caso de uso elaboración horario |
| RF8 | El sistema debe facilitar información de los estudiantes hacia los profesores | Caso de uso profesores |
| RF9 | El sistema debe poder desplegar reportes | Caso de uso reportes |

3.2.1.2.REQUERIMIENTOS NO FUNCIONALES

Los requerimientos no funcionales del proyecto son los siguientes:

- El sistema debe tener una interfaz amigable para el usuario y así permitir un fácil uso.
- El sistema debe permitir el acceso desde cualquier sitio con acceso a internet.
- La aplicación web debe ocupar la mayor parte de la pantalla para poder visualizar las diferentes funciones.
- La aplicación web debe tener compatibilidad de visualización con las últimas versiones de los navegadores.

Para tener una idea clara de cómo resolver cada uno de los requerimientos funcionales que se obtuvieron de la institución educativa, serán plasmados en diagramas de casos de uso los cuales nos darán una visualización grafica de cómo será el comportamiento de cada actor (administrador, estudiante, docente, padre de familia) con el sistema.

3.2.1.3. DIAGRAMA DE CASOS DE USO

En los diagramas de casos de uso se visualiza gráficamente el comportamiento de un sistema en desarrollo sin tener que especificar como se implementara dicho comportamiento.

Para una comprensión más clara acerca de los casos de uso que existirán en el sistema se realizaron 4 casos de uso los cuales serán de primer nivel también los conoceremos como los más primordiales.

3.2.1.3.1. CASO DE USO SISTEMA ACADÉMICO

El caso de uso principal del sistema en la cual se puede observar que el usuario administrador llega a tener acceso a todo el sistema sin excepción mientras que tanto usuario tutor como estudiante se encuentran limitados en el acceso al sistema de igual forma podemos observar que actor profesor también se encuentra limitado en funciones estas funciones que se les asigna a los usuarios secundarios son los únicos que necesitan para estar informados acerca de sus intereses.

En la figura 8 se visualiza el caso de uso principal.

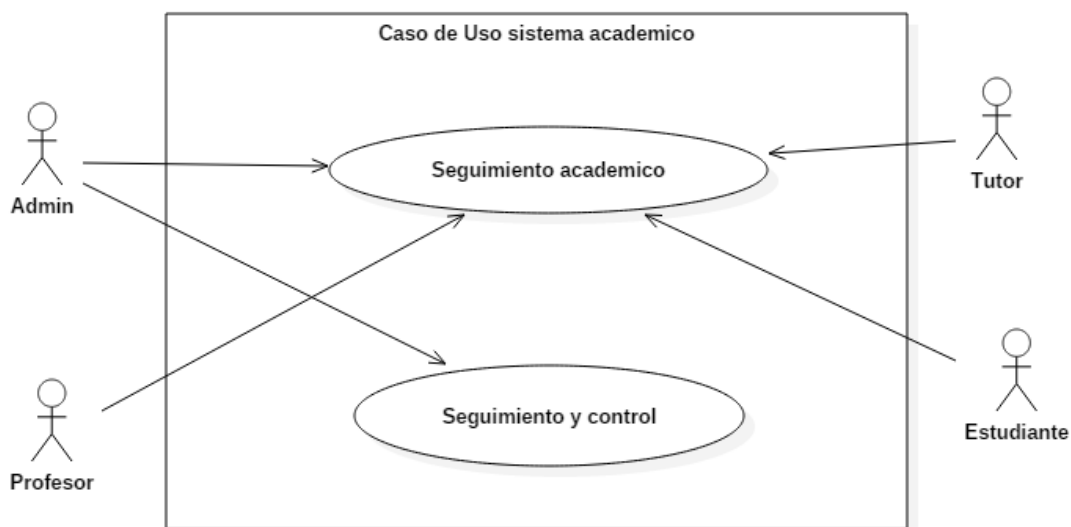


Figura 8. Caso de Uso del sistema de seguimiento académico

3.2.1.3.2. CASO DE USO ESTUDIANTE

El usuario estudiante como el tutor son los mayores beneficiarios con el sistema académico planteado ya que en la actualidad no se tiene acceso a los datos que requieren como las notas bimestrales o las notas generales.

En la figura 9 como también en la Tabla 2 se observa todas las funciones las cuales tiene tanto el estudiante como el tutor ya que ellos tendrán acceso a la misma información con un pequeño variación el tutor recibirá información si su hijo fuera observado por alguna falla que cometiera.

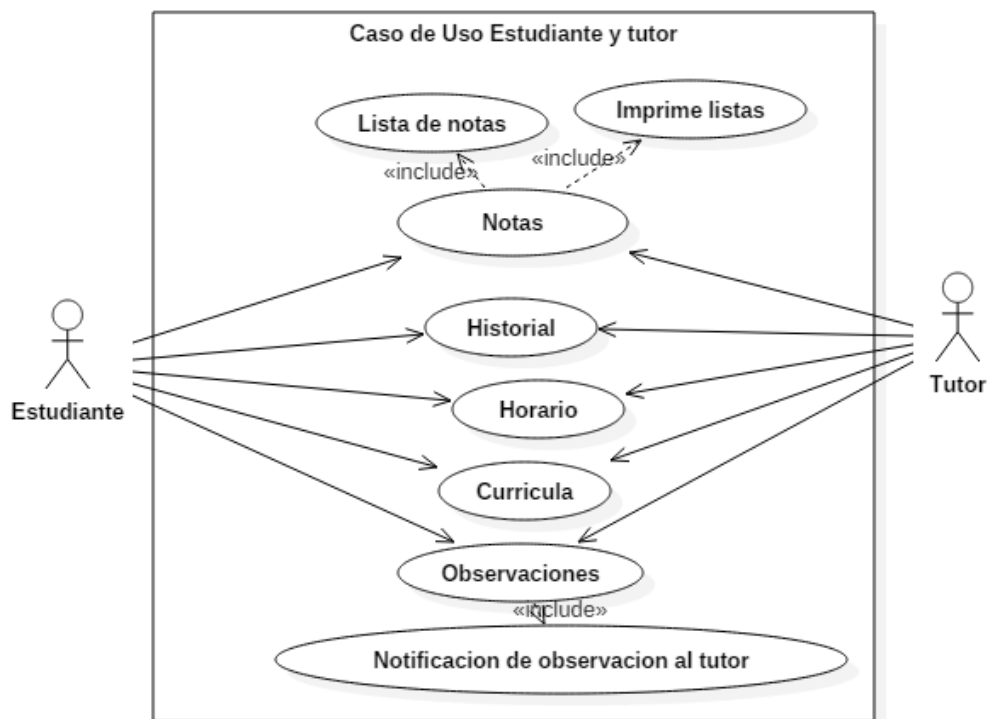


Figura 9. Caso de uso Estudiante Tutor

Tabla 2 Descripción Caso de uso estudiante – tutor

| DOCUMENTO DE DESCRIPCIÓN DE CASOS DE USO ESTUDIANTE - TUTOR | | | | | | | | | |
|--|---|--|-----------------|---|--|--|-----------------------------|--|--------------------------|
| Nombres: | Informe del estudiante | | | | | | | | |
| Actores: | Padres de Familia | | | | | | | | |
| Descripción | Describe el rendimiento individual de los estudiantes. | | | | | | | | |
| Flujo Principal | <table border="1"> <thead> <tr> <th>Eventos Actor</th> <th>Eventos Sistema</th> </tr> </thead> <tbody> <tr> <td>1. - Activa la función para la generación de reportes.</td> <td>1. - Muestra en pantalla las calificaciones del alumno.</td> </tr> <tr> <td>2. - El usuario escoge si quiere imprimir el reporte.</td> <td>2. - El sistema imprime.</td> </tr> <tr> <td>3. - El usuario elige si seguir o terminar.</td> <td>3. - Terminar sesión.</td> </tr> </tbody> </table> | Eventos Actor | Eventos Sistema | 1. - Activa la función para la generación de reportes. | 1. - Muestra en pantalla las calificaciones del alumno. | 2. - El usuario escoge si quiere imprimir el reporte. | 2. - El sistema imprime. | 3. - El usuario elige si seguir o terminar. | 3. - Terminar sesión. |
| | Eventos Actor | Eventos Sistema | | | | | | | |
| | 1. - Activa la función para la generación de reportes. | 1. - Muestra en pantalla las calificaciones del alumno. | | | | | | | |
| | 2. - El usuario escoge si quiere imprimir el reporte. | 2. - El sistema imprime. | | | | | | | |
| 3. - El usuario elige si seguir o terminar. | 3. - Terminar sesión. | | | | | | | | |
| Precondición | Se tiene las calificaciones actualizadas de los estudiantes. | | | | | | | | |
| Post Condición | Se tiene impreso el informe correspondiente. | | | | | | | | |

3.2.1.3.3. CASO DE USO PROFESOR

En la Figura 10 como en la Tabla 3 se observa todos los usos que el profesor puede darle al sistema académico tanto como recabar lista de estudiantes como también información de sus horarios con la opción de observar alguna falla que cometiere algún estudiante.

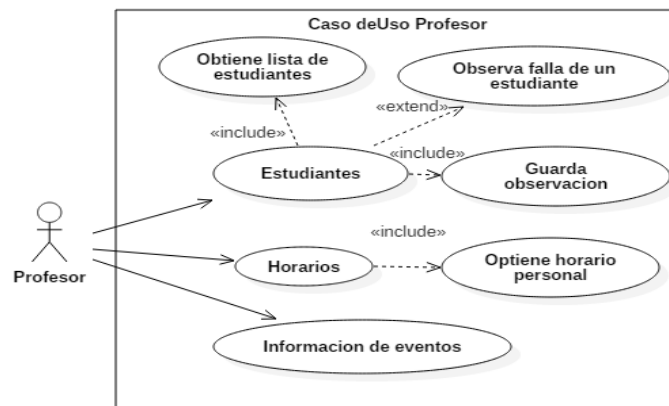


Figura 10. Caso de Uso Profesor Sistema Académico

Tabla 3 Descripción caso de uso docente

| DOCUMENTO DE DESCRIPCIÓN DE CASOS DE USO DOCENTE | | |
|---|---|---|
| Nombres: | Información del rendimiento por cursos en centralizados. | |
| Actores: | Docente | |
| Descripción | Describe el reporte de estadísticas de rendimiento y calificaciones de los estudiantes en centralizadores por cursos. | |
| Flujo Principal | Eventos Actor | Eventos Sistema |
| | 1. - Activa la función para dar el reporte de rendimiento de los estudiantes y estadísticas por cursos. | 1. Muestra en pantalla los datos de calificaciones y las estadísticas. |
| | 2. - Realiza las observaciones correspondientes. | 2. -Revisa la información de acuerdo al registro de alumnos. |
| | 3. - Activa la impresión de reportes. | 3. - Muestra en pantalla confirmación de impresión de reporte. |
| Precondición | Adquirir los datos actualizados del módulo de seguimiento académico. | |
| Post Condición | Se tiene impreso los informes correspondientes. | |
| Presunción | La base de datos de proyectos está disponible. | |

3.2.1.3.4. CASO DE USO ADMINISTRACIÓN

En cualquier sistema de información el usuario administrador siempre llega a ser el encargado de realizar muchas tareas, llevando toda la responsabilidad de que cualquier información que adicione o cambie tenga la veracidad absoluta ya que esta información será la que todos los usuarios de más bajo nivel observaran.

En la figura 11 como en la Tabla 4 se plasma todos los casos de uso el cual da una visión grafica de todos los privilegios que el usuario administrador tendrá en el sistema, poseerá acceso a la información de los usuarios ya sean estos estudiantes como profesores o administradores tendrá la potestad de poder adicionar o cambiar datos también contendrá acceso a realizar inscripciones al establecimiento educativo, podrá crear horarios, la función más importante de sus funciones adicionar notas de cada estudiante el cual será la información más valiosa de la institución.

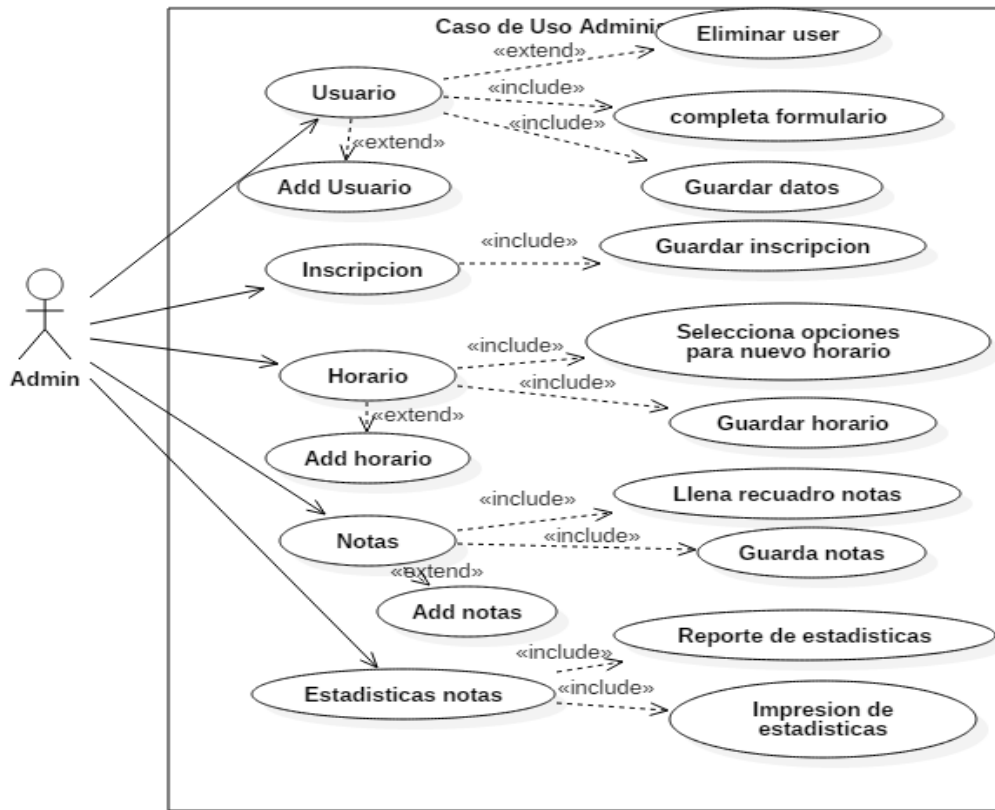


Figura 11. Caso de Uso Administrador sistema académico

Un usuario Administrador a diferencia de los otros dos usuarios los cuales son usuario profesor, usuario estudiante, cargara con una gran responsabilidad ya que si en algún registro llegase a errar en la manipulación de la información los usuarios finales se verán perjudicados, esto puede llevar a confusiones y molestia, es por ello que en cada función que realice el administrador siempre tendrá la opción de editar los datos, con ello podrá corregir cualquier falla que se haya cometido en el proceso de inserción de datos al sistema.

Tabla 4 Descripción caso de uso administrador

| DOCUMENTO DE DESCRIPCIÓN DE CASOS DE USO ADMINISTRADOR | | |
|---|---|---|
| Nombres: | Control rendimiento de estudiantes generación informe mensual | |
| Actores: | Administrador | |
| Descripción | Directo, secretario podrán realizar diversas funciones | |
| Flujo Principal | Eventos Actor | Eventos Sistema |
| | 1. - Activa módulo de seguimiento académico. - Activa módulo de control del personal. | 1. - Muestra en pantalla el menú de opciones de seguimiento académico, el control de personal. |
| | 2. - Reporte de alumnos efectivos abandonos y promedios de rendimiento. | 2. - Confirma el registro de archivos. |
| | 3. - Imprime los reportes | 3. - Muestra en pantalla la confirmación de impresión. |
| Precondición | - El reporte se confirma con los registros de archivos. | |
| Post Condición | - El informe paso por aprobación de dirección. | |
| Presunción | - La base de datos de proyectos está disponible. | |

Un aspecto muy importante en cualquier sistema es la seguridad que se puede brindar hacia los usuarios la seguridad que se brinda hacia los usuarios son las credenciales que se les entrega para que puedan entrar al sistema y poder realizar tareas de su interés para ello podemos contemplar un caso de uso de Inicio de sesión al ser el más importante del sistema en la Tabla 5 se observa la descripción del caso de uso de Inicio de sesión.

Gracias a los casos de uso se puede visualizar cada uno de las tareas que podrá realizar cada uno de los actores que intervienen en el sistema como ser el Administrador, Profesor y tanto el estudiante como el tutor, para observar más a detalle los casos de uso de tercer nivel.

Tabla 5 Descripción caso de uso Inicio de Sesión

| Caso de Uso | Inicio de Sesión |
|------------------------------|--|
| Actor | Administrador, profesor, estudiante |
| Precondición | Contar con un usuario |
| Postcondicion | Usuario queda identificado |
| Flujo de Evento Básico | |
| 1 | El operario abre la aplicación web desde un navegador |
| 2 | el sistema muestra un formulario de autenticación |
| 3 | el operario ingresa su usuario y contraseña |
| 4 | El sistema muestra la interfaz principal |
| 5 | El Operario Cierra la sesión |
| 6 | Finaliza el Caso de Uso |
| Flujo de Eventos Alternativo | |
| 3.1 | Si la clave es incorrecta el sistema muestra mensaje de error clave no valida y retorna al formulario de autenticación paso 2. |
| 3.2 | Si el usuario es incorrecto el sistema muestra mensaje de error usuario no registrado y retorna al formulario de autenticación paso 2. |

3.2.2. MODELO CONCEPTUAL

En esta fase de la metodología (OOHDM) se detalla de manera gráfica las entidades (clases) que intervendrán en el sistema así mismo también se detalla la relación que existe entre cada uno de ellos para esta parte se hará uso de los diagramas de clases de igual manera expondremos el modelo para la base de datos el cual es el modelo Entidad Relación.

la segunda forma normal, la tercera forma normal, con toda esta optimización llego a ser un modelo muy completo para poder cumplir con todos los objetivos que se planteó anteriormente.

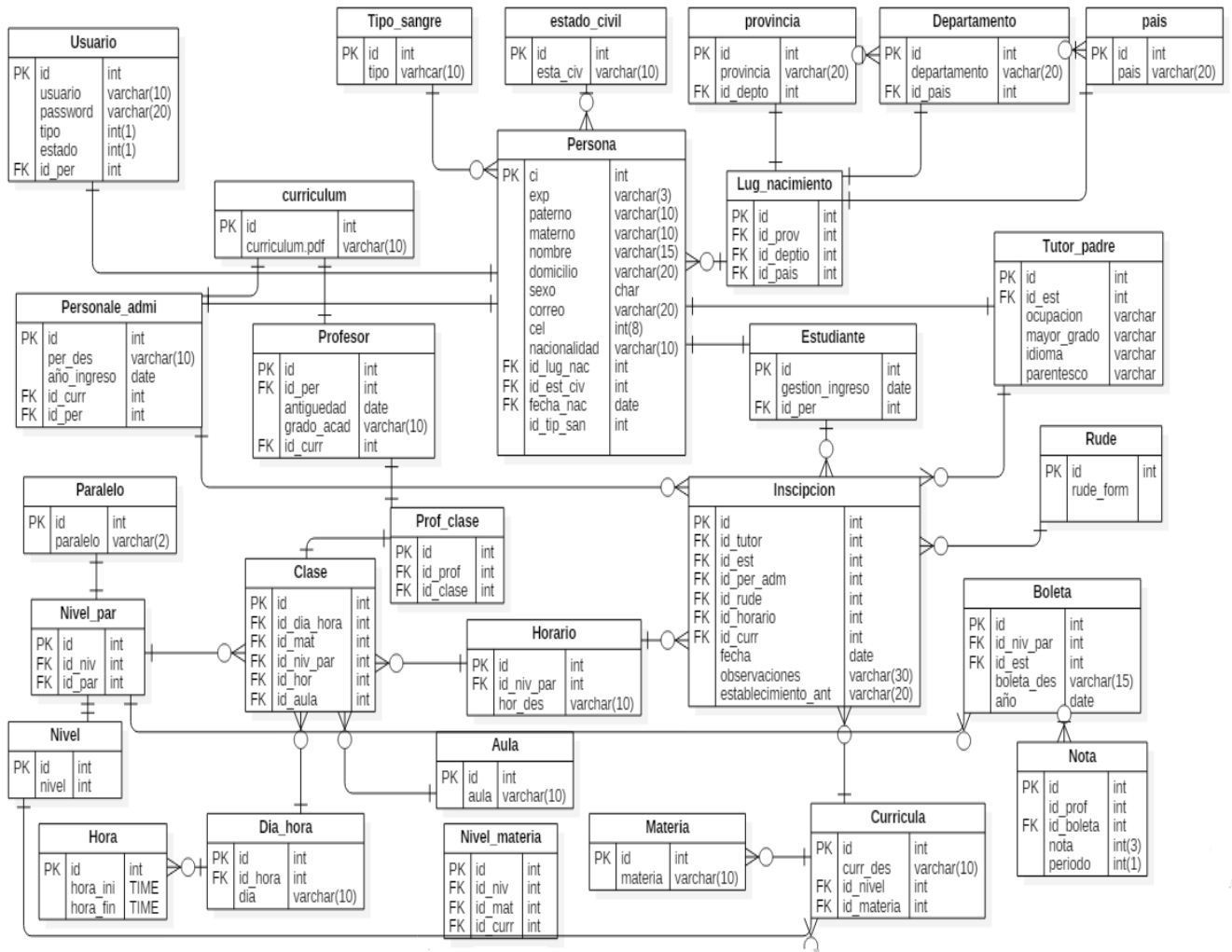


Figura 13. Diagrama Relacional del sistema académico

Así mismo se desarrolló un Diagrama entidad relación del cual se partió para llegar a realizar el modelo relacional, el diagrama entidad relación (ER) comprende todas las entidades, como se encuentran relacionadas entre si tanto como otros aspectos, en ella observamos el diagrama completo de la unidad educativa.

En la Figura 14 se observa el modelo entidad relación con la cual cuenta el sistema académico.

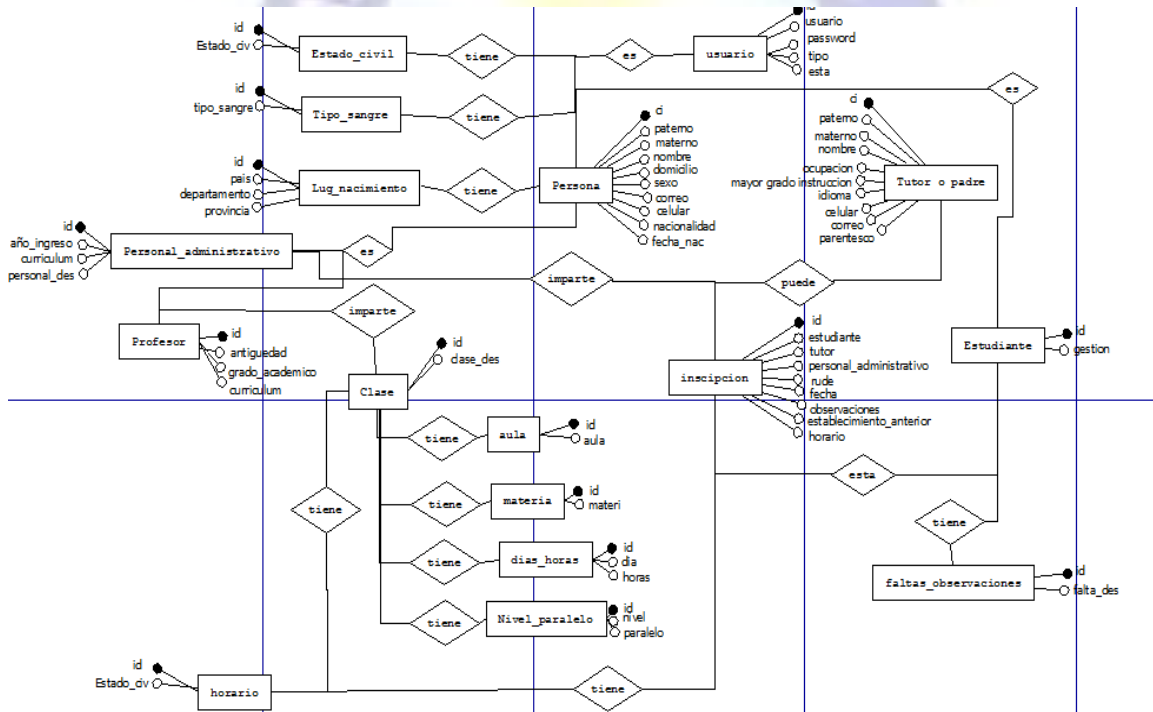


Figura 14 Diagrama entidad relación sistema académico

3.2.3. DISEÑO NAVEGACIONAL

Para el estudio del diseño navegacional se tomó en cuenta las dos herramientas para la elaboración, el diseño de clases navegacionales y el esquema de contextos navegacionales, de los cuales se hará uso solo de los puntos más relevantes del sistema siendo ellos como el inicio y autenticación en el sistema, seguidamente se tendrá todo el diseño navegacional del

administrador, también se plasmara el diseño navegacional del profesor y por ultimo tomaremos el diseño navegacional del estudiante y tutor.

3.2.3.1. DISEÑO NAVEGACIONAL PRINCIPAL

Diseño navegacional Inicial. Todo desarrollador en algún punto conoció el inicio o también conocido como índice de cualquier proyecto, es el arranque de todo con esa página de inicio o índice es posible iniciar una navegación por todo un proyecto por medio de referencias hacia cualquier otra de las paginas las cuales tendremos acceso a continuación en la figura 15 podemos observar el diseño navegacional del index del sistema académico.

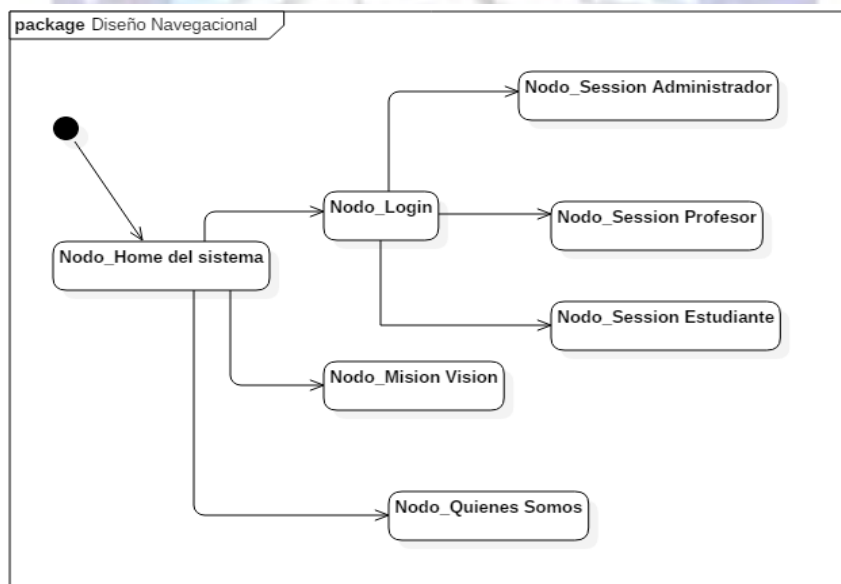


Figura 15 Diseño navegacional inicio del sistema

3.2.3.2. DISEÑO NAVEGACIONAL ADMINISTRADOR

Diseño navegacional modulo-administrador. El administrador como se mencionó anteriormente cuenta con todo los privilegios que el sistema brinda sin excepción, en la

figura 16 se observa como el administrador accede a la información de los usuarios con la facultad de poder cambiar datos, también tiene acceso al nodo de inscripción siendo el encargado de realizar la inscripción de una gestión, del mismo modo tiene el acceso a los horarios teniendo el poder de cambiarlos o crear uno nuevo, como se pudo visualizar cada nodo sea el último retorna hacia su antecesor de ese modo es que al realizar una tarea el sistema nos re direccionara hacia el inicio para poder seguir trabajando.

El usuario Administrador posee una navegación mucho más amplia en comparación de los otros usuarios, conteniendo al mismo tiempo muchos nodos que enlazan los puntos de navegación, todos estos puntos de navegación solo podrán ser accedidos por un usuario administrador, en cada enlace siempre se tendrá un control de verificación del token que el usuario administrador poseerá al momento de la autenticación.

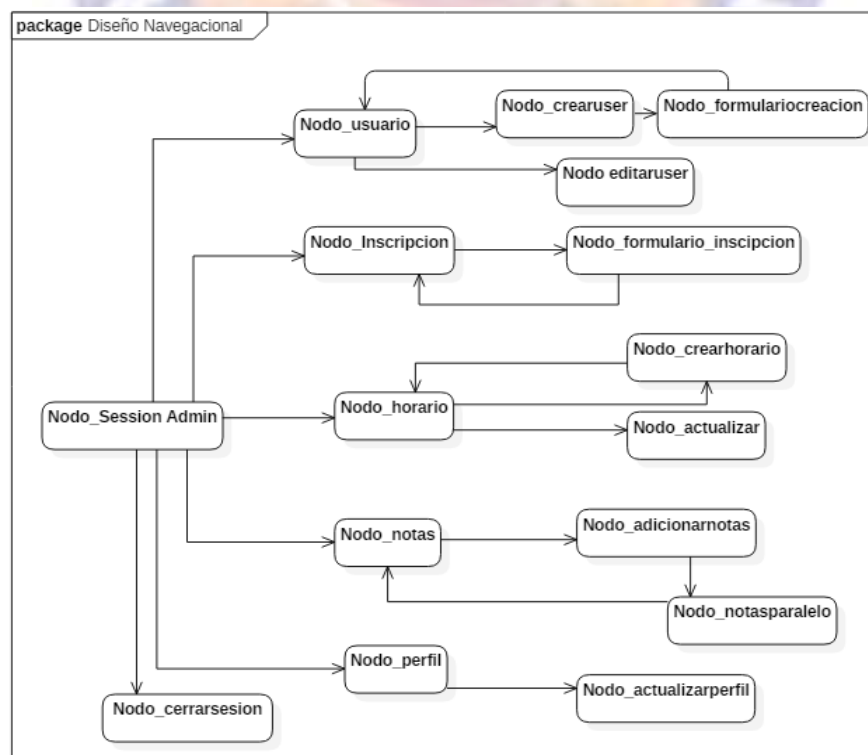


Figura 16. Diseño navegacional modulo-administrador

3.2.3.3. DISEÑO NAVEGACIONAL ESTUDIANTE

Diseño navegacional Estudiante-tutor. Un estudiante así como un tutor llegan a ser el usuario final del sistema ya que toda la información que se brindara en el sistema será de gran interés para estos usuarios tanto para el estudiante como para el tutor, estos usuarios al ser usuarios finales tendrán muchas restricciones acerca de la manipulación del sistema solo tendrán reportes no podrá hacer cambios ni modificaciones a los datos que se tienen en el sistema académico.

En la figura 17 se observa el diseño navegacional tanto del estudiante como del tutor, así mismo se advierte que este usuario tendrá acceso al perfil para ver sus datos personales, podrá acceder al nodo horario para ver su horario antes que inicien sus clases y, el más importante, podrá acceder a la información de sus notas obtenidas hasta el periodo, de igual forma podrá acceder al historial de todas sus notas registradas.

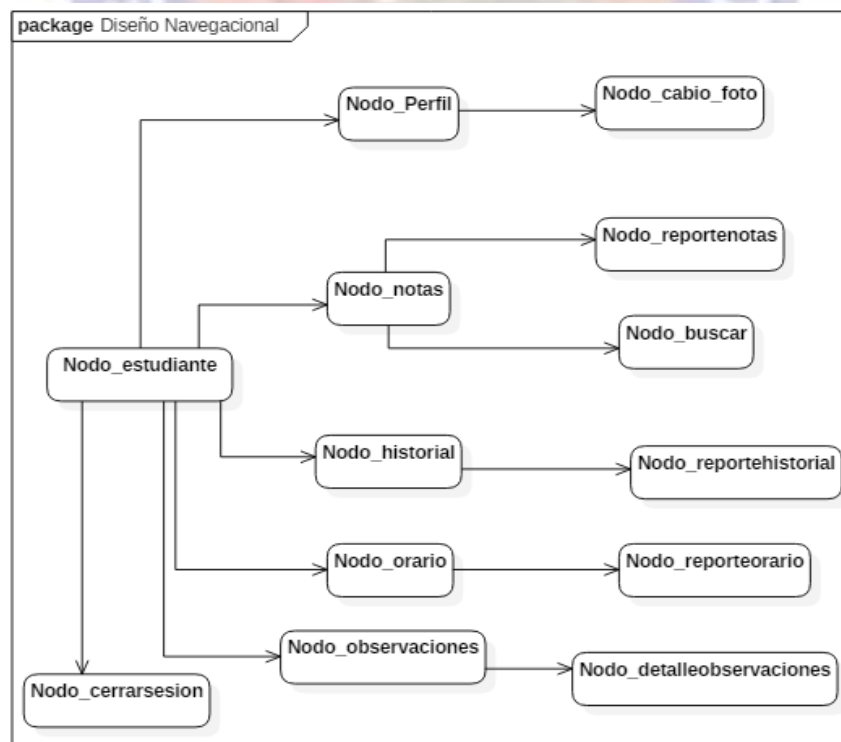


Figura 17 diseño navegacional estudiante

3.2.3.4.DISEÑO NAVEGACIONAL PROFESOR

Diseño navegacional Profesor. Un profesor será el usuario intermedio tendrá la potestad de realizar algunas manipulaciones de datos en el sistema académico así mismo se entiende que el profesor de un establecimiento es el encarado de la educación y disciplina de los estudiantes, también de ellos depende que los estudiantes mejoren en conocimiento pero para eso es necesario tener un poco de control acerca de cada estudiante ese control lo realizara un profesor en comunicación directa con sus tutores.

En la figura 18 se puede observar el diseño navegacional de un profesor como se puede observar un profesor tiene acceso a datos de sus estudiantes para que de ese modo pueda analizar cualquier situación pero el rol más importante, el cual cumplirá un usuario profesor, será el de notificar cualquier observación que se le haga a un estudiante.

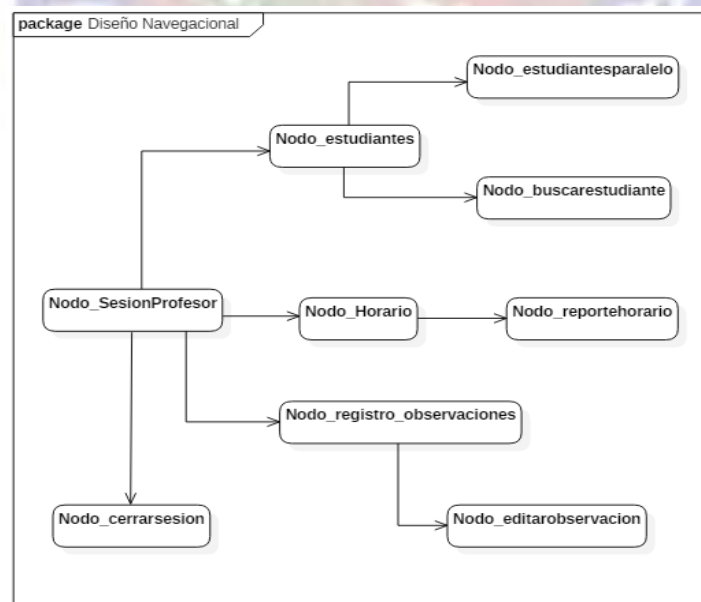


Figura 18 Diseño navegacional profesor

3.2.3.5. MODELO DE CLASES NAVEGACIONALES

Esquema de Clases Navegacionales. Mediante el diagrama Entidad – Relación se creó los nodos y sus enlaces, los cuales son otra alternativa de exponer una clase, contendrá los elementos que tiene la información como una vista en el cual se muestra cada parte del sistema como son: Pagina inicial, pagina menú izquierda, estudiante, etc.

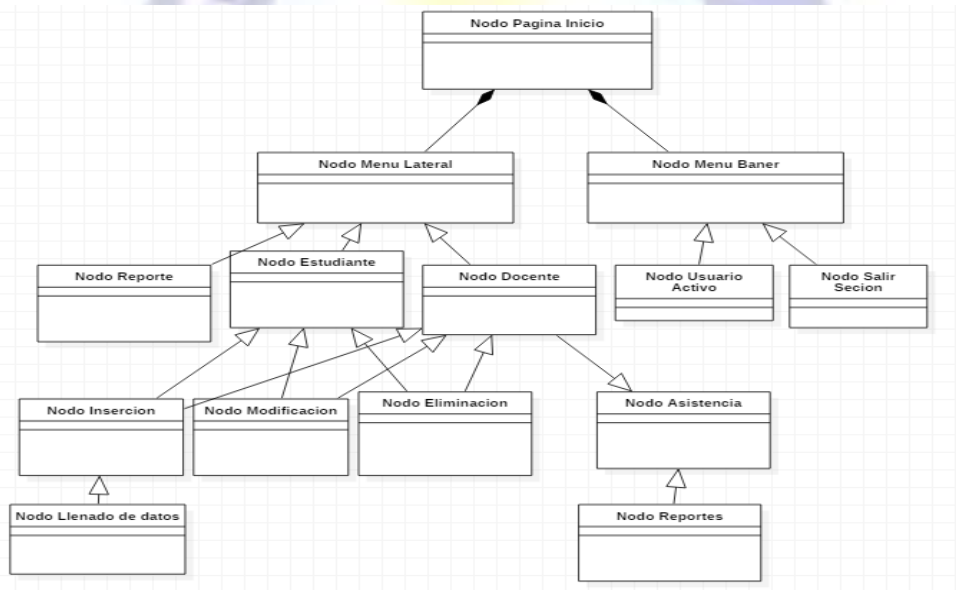


Figura 19 Modelo de Clases Navegacionales

La Figura 19 da una muestra de un modelo de clases navegacionales, mismo que sirve de referencia para entender todo el sistema en conjunto a continuación se describe todas las clases y su forma de navegación.

Nodo Página de inicio. Este nodo página de inicio plasmara el inicio de todo el proyecto mencionado, el inicio de la interfaz, la presentación que tendrá el sitio web.

Nodo Menú Sesión. Es el nodo que se encuentra en la parte derecha del sitio web que dará opción de ver la sesión como también salir de la sesión.

Nodo Menú Lateral. El menú que tendrá todas las opciones que contiene el sistema como ser el de sacar un reporte de algunos datos, visualizar las opciones que se tenga dependiendo del tipo de usuario como ser estudiante o profesor, los cuales se podrá respectivamente visualizar sus datos, notas y sacar reportes de dichos datos.

3.2.4. DISEÑO DE INTERFAZ ABSTRACTA

Una vez terminada la fase del diseño navegacional se pasa al diseño de las interfaces abstractas que contendrá el sistema, se realizaron 5 interfaces abstractas los cuales llegan a ser los más relevantes estos son: interfaz abstracta inicio, interfaz abstracta inicio de sesión, interfaz abstracta administrador, interfaz abstracta profesor, interfaz abstracta estudiante o tutor estas son las interfaces más importantes en el sistema es por ello que se los debe realizar antes de pasar a lo que será la implementación de los mismos.

Los diseños de interfaz abstracta reflejan las características principales los cuales se encuentran en los requerimientos que se obtuvieron en la primera fase de esta metodología gracias a la interfaz abstracta es que se llegó a pasar al último paso que es la implementación que es llevar los datos que se priorizan en estas interfaces hacia un lenguaje de programación para que se muestre en forma apreciable para el usuario final.

Las interfaces abstractas generalmente no tienen mucho valor al llegar al final del desarrollo del sistema, sin embargo son muy útiles para el entendimiento y la administración del sistema, ya sea a largo como a corto plazo. El sistema cuenta con una característica muy importante: es escalable al usar las tecnologías actuales como ser nodejs vuejs cualquier actualización puede ser añadida inmediatamente y sin errores al sistema, del mismo modo la base de datos puede llegar a crecer ya que es un modelo en el que la parte del servidor se puede modificar en la forma más creativa posible.

3.2.4.1.INTERFAZ ABSTRACTA INICIO

Como en todo sitio web la página de inicio es la parte fundamental ya que será lo primero que el usuario visualizara esta interfaz es muy amigable con los datos que realmente requiere un usuario a primera vista como ser: las ultimas noticias, las autoridades actuales, los medios para poder contactarse con el establecimiento, la dirección exacta en la cual se puede ubicar geográficamente la unidad educativa.

También cuenta con un menú el cual permite acceder a: la historia, misión, visión de la unidad educativa los cuales serán la parte informativa del establecimiento y en la parte superior contiene el enlace para poder llegar al formulario de inicio de sesión.

En la Figura 20 se observa la interfaz abstracta del inicio del sistema.



Figura 20. Diseño Interfaz Abstracta Página de Inicio

3.2.4.2. INTERFAZ ABSTRACTA INICIO DE SESIÓN

Un usuario al ser profesor, estudiante o administrador posee un acceso al sistema para realizar algunas funciones las cuales el sistema le guiará para poder acceder a dichas funciones. Estos usuarios tendrán que pasar por el llenado de las credenciales que serán otorgadas por parte de dirección de la unidad educativa ya que ellos serán los encargados de la administración del sistema, será una forma de seguridad con la cual se controlará que solo usuarios autorizados lleguen a ingresar al sistema con un rol específico ningún otro usuario externo podrá acceder sin contar con estas credenciales.

La parte del sistema académico encargada de proporcionar un token único es el servidor también conocido como *backend*, si las credenciales ingresadas son válidas el usuario podrá permanecer en el sistema para realizar las tareas requeridas.

En la figura 21 se observa la interfaz abstracta de inicio de sesión.

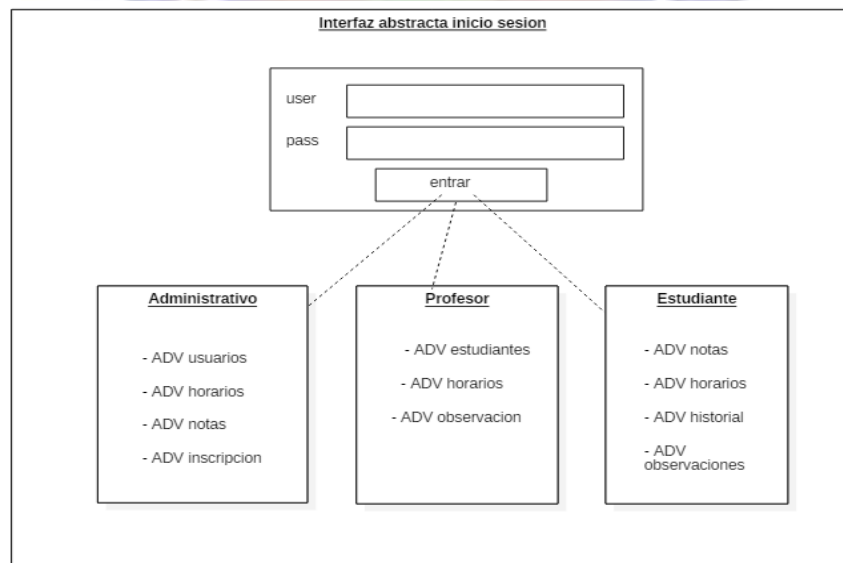


Figura 21. Diseño de Interfaz Abstracta Pagina Inicio de Sesión

3.2.4.3. INTERFAZ ABSTRACTA ADMINISTRADOR

El usuario administrador posee muchas funciones en el sistema como ya mencionamos anteriormente este será el súper usuario, tendrá acceso a cualquier parte del sistema son muchas las tareas que realizara resumimos sus tareas en la interfaz principal.

Primeramente se observa que el administrador posee un menú en la parte izquierda el cual describimos uno por uno a continuación:

Usuario. Este botón o referencia re direcciona hacia una interfaz que permitirá visualizar los usuarios del sistema con opciones a elegir si se quiere visualizar a los usuarios de tipo profesor, estudiante o administrador, así mismo tiene la opción de realizar cambios a un usuario como ser actualización de datos, eliminación de un usuario y también crear un nuevo usuario.

Inscripción. Este re direccionara hacia un formulario para realizar la inscripción de un estudiante, plasmara datos de los estudiantes para seleccionar a cual es al que se requiere inscribir y en que paralelo.

Horario. Este re direccionara hacia la interfaz que observa los horarios existentes actualmente en el sistema para cada uno de los paralelos, así mismo se podrá realizar cambios a estos horarios como ser actualización, eliminación o crear un nuevo horario.

Notas. La parte más importante en el sistema ya que estos llegan a ser datos de vital prioridad son: las notas de los estudiantes estos datos serán las que el estudiante como el tutor requerirán en todo momento este botón re direccionara hacia la interfaz en la cual se visualizara todas las notas de los estudiantes ordenados por paralelos a su vez también se puede observar una estadística de las notas adquiridas por los estudiantes en los últimos

años con ello se realizara una toma de decisiones para el manejo de la administración y temática de enseñanza aplicada por los docentes.

Curricular. La curricula es el plan de estudios que se registrá en el establecimiento educativo primeramente se observa todas las currículas existentes como en todos los casos también podrá editar, eliminar y crear una nueva curricular.

En esta interfaz abstracta se observa las características básicas que se requiere en la interfaz de inicio del administrador ya que aparte de poseer la posibilidad de gestionar usuarios, horarios, notas, curricular también poseerá algunas otras funciones que se describen en el diseño navegacional.

Cerrarsesión. Este llamara una función que se encargara de destruir el token que se asignó desde inicio de sesión, este será una medida para salir del sistema con el re direccionamiento hacia el formulario de inicio de sesión con ello se podrá abandonar el sistema de forma segura.

Interfaz abstracta administrador

| menu | Colegio Nacional Mixto Bic Lib Simon Bolivar | cerrarsesion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------|--|--------------|---------|-------------|--------|------|-------------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| usuario | <input style="width: 80%;" type="text"/> <input style="width: 15%; height: 20px;" type="button" value="buscar"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| usuario | <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">usuario</th> <th style="width: 20%;">nombre</th> <th style="width: 20%;">estado</th> <th style="width: 20%;">tipo</th> <th style="width: 30%;">operaciones</th> </tr> </thead> <tbody> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> <tr><td> </td><td> </td><td> </td><td> </td><td> </td></tr> </tbody> </table> | | usuario | nombre | estado | tipo | operaciones | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| usuario | nombre | estado | tipo | operaciones | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| inscripcion | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| horario | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| notas | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| curricula | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figura 22. Diseño de Interfaz Abstracta Página de Inicio Administrador

3.2.4.4. INTERFAZ ABSTRACTA ESTUDIANTE

La función del usuario estudiante en el sistema académico es de obtener información acerca de su rendimiento podrá observar las notas que obtuvo en el bimestre ya sea primero segundo tercero o cuarto, también podrá observar un historial de todas las notas de las gestiones pasada, el horario en el cual este asignado, el perfil nombres como apellidos y todos los datos que sean de su propiedad se visualizaran por el estudiante del mismo modo podrá acceder a la interfaz donde se plasman todas las observaciones que se le realizaron y los cuales fueron notificadas hacia su tutor.

El usuario tutor tendrá acceso al sistema por la misma cuenta que del estudiante tendrá todas las funciones que tendrá el estudiante de ese modo podrá realizar un seguimiento académico hacia sus dependientes.

| Interfaz abstracta Estudiante | | | | | | |
|-------------------------------|--|--|-----------|------------|--------------|-------|
| menu | | Colegio Nacional Mixto Bic Lib Simon Bolivar | | | cerrarsesion | |
| usuario | | NOTAS | | | | |
| | | materia | primer bi | segundo bi | tercer bi | final |
| | | | | | | |
| notas | | | | | | |
| historial | | | | | | |
| horario | | | | | | |
| perfil | | | | | | |
| observaciones | | | | | | |

Figura 23. Diseño de Interfaz Abstracta Pagina Inicio Estudiante

3.2.4.5. INTERFAZ ABSTRACTA PROFESOR

Profesor o docente es un usuarios con algunas capacidades en el sistema, primeramente el profesor es el que se encuentra más cerca del estudiante, es encargado de que un estudiante

aprenda la materia que imparte en caso de que exista ausencia de parte del tutor el estudiante no rendirá como debería ya que en algunos casos el estudiante evadirá los castigos que el profesor pueda asignar.

Las notificaciones siendo la funcionalidad más importante del profesor son aquellas que se enviarán hacia el tutor directamente hacia su correo electrónico de este modo es que el tutor estará informado en tiempo real si su hijo o hija está cometiendo algunas faltas en el establecimiento educativo.

El usuario profesor tendrá acceso a las notas de los estudiantes de sus paralelos podrá ver sus horarios y su perfil el cual contendrá sus datos personales

Interfaz abstracta Docente

menu
Colegio Nacional Mixto Bic Lib Simon Bolivar
cerrarsesion

usuario

lista estudiantes

horarios

observaciones

perfil

| estudiante | nombre | paralelo | periodo | operaciones |
|------------|--------|----------|---------|-------------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Figura 24. Diseño de Interfaz Abstracta Página de Inicio Docente

3.2.5. IMPLEMENTACIÓN

La obtención de requerimientos el cual fue realizado en el primer paso de esta metodología de diseño de hipertexto fue la parte fundamental por la cual se logró guiar el desarrollo del sistema académico ya que con ello se obtuvo las características primordiales para poder

iniciar la lógica de desarrollo, cabe mencionar que a lo largo de la metodología el primer paso llegaría a ser el importante.

En el diseño conceptual se observó el análisis de la semántica de los datos el cómo se realizara el manejo de los mismos todos estos datos se plasmaron en un diagrama relacional la cual se encuentra implementada en la base de datos en el caso de sistema académico será el gestor de base de datos postgresql, el manejo de datos debe realizarse de forma ordenada, si no se realiza de esta forma se llegaría a tener algunos tropiezos posteriormente, completando este paso se pudo proseguir al paso siguiente diseño navegacional.

El diseño navegacional fue la parte donde se realizó el estudio de la navegación que existirá en el sitio todas las condiciones para poder navegar en ciertos lugares como también acceder a sitios sin ser usuarios, todos estos diagramas que se presentan en este paso dan la idea de cómo guiarse para llegar a un cierto punto, del mismo modo nos ayudaron a observar cuantos nodos son necesarios para que el sitio web este completo una vez terminada se pasó al siguiente paso el diseño de interfaz abstracta.

La interfaz abstracta siendo el lugar principal para la elaboración de interfaces básicas que contendría el sistema académico, cada una de las interfaces abstractas plasmadas en este punto contienen las características mínimas que debería contener las interfaces reales, conteniendo por ello las características principales que el usuario requerirá al momento de acceder al sistema como pudimos observar en el administrador será el encargado de gestionar tanto las cuentas de usuarios, los horarios, las inscripciones, el plan de estudios y las notas de cada uno de los estudiantes a su vez el estudiante podrá acceder a muchos de sus datos que son requeridos como son las notas obtenidas, las observaciones, el historial del mismo modo el profesor tendrá un papel importante ya que este será el encargado de realizar notificaciones a los padres de aquel estudiante que no esté dando su mayor rendimiento una vez terminada este paso llegamos al último paso y no menos importante importante la implementación.

La implementación el paso final de la metodología de desarrollo orientado a objetos, en esta parte se visualiza el cómo se realizó toda la semántica del sitio web el cual lo explicamos a continuación punto por punto.

Servidor. La parte del servidor fue realizada en JavaScript con la ayuda de las nuevas tecnologías nodejs, express y como gestor de base de datos postgresql, primeramente la app web contiene un archivo principal el cual es el arranque de todo el sistema este archivo lleva el nombre de app.js es en este archivo donde arranca todo.

El servidor contiene rutas las cuales contienen seguridad de acceso el cual solo un usuario con los permisos necesarios podrá acceder a las mismas, existen tres archivos de rutas las rutas para el administrador, las rutas para el profesor y las rutas para el estudiante o tutor en estas rutas se definen los controladores necesarios para cada una de las tablas que existen en la base de datos.

Una carpeta con nombre models es donde se contendrá los archivos de todas las tablas que se crearon en la base de datos con ella es que se gestiona cada una de las tablas que contendrá la base de datos.

Express por su parte realizara la tarea de gestionar la conexión como también el levantamiento del servidor.

En la parte de las vistas es donde brilla vuejs quien es un compilador para el lado del cliente o también conocido como *frontend* vuejs contendrá una página de inicio el cual será la vista principal que llamara a los demás componentes cuando así se lo requiera.

Gracias a una librería axios es que se realiza la conexión con el servidor para de ese modo poder trabajar conjuntamente y de ese modo poder realizar peticiones y otras operaciones.

Al igual que nodejs vuejs cuenta con una librería vuerouter el cual gestiona la forma de navegación las rutas las cuales se deben seguir para llegar hacia cierta información.

Vuejs se caracteriza por trabajar reactivamente realizar tareas asíncronamente sin necesidad de cargar otras páginas.

Para las vistas se hizo uso de los estilos los cuales nos facilitó otra librería de vuejs que es vueify.

3.2.5.1. IMPLEMENTACIÓN PAGINA INICIAL

Como mencionamos anteriormente la página inicial es la parte principal ya que es la primera impresión que llega a dar al usuario, en la página inicial se plasman las últimas noticias que existe en la unidad educativa esta información será actualizada constantemente por el administrador en la parte superior se puede observar el menú de opciones para poder navegar hacia la información de misión y visión que tiene este lugar así mismo se podrá acceder a la reseña histórica y el sitio para poder realizar el llenado de las credenciales o también conocido como formulario de inicio de sesión también en esta parte inicial podremos encontrar la dirección y contacto podemos observarlo en la Figura 25.

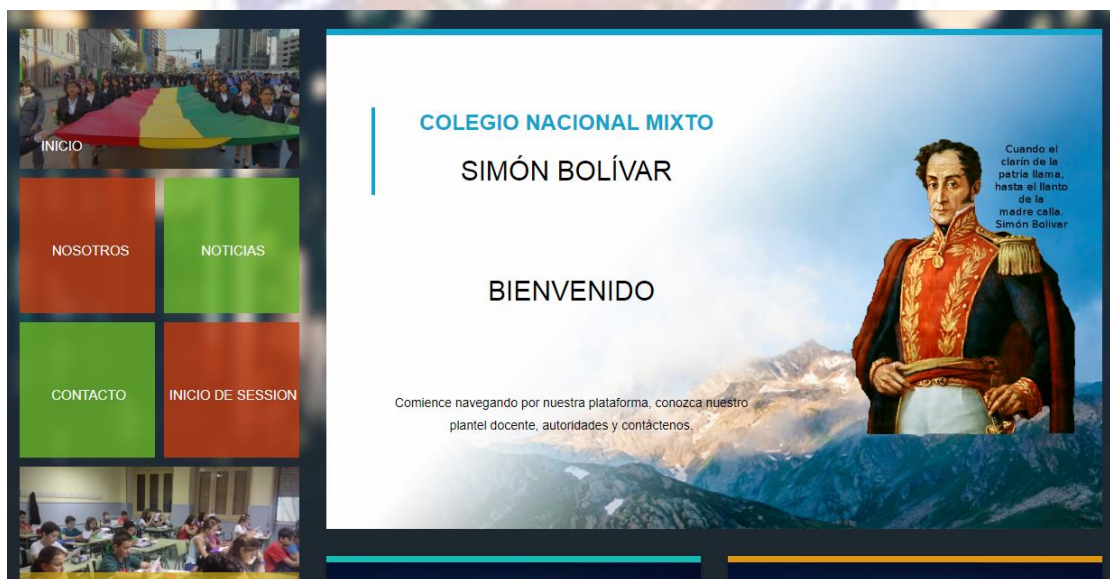


Figura 25. Implementación Página Inicial

3.2.5.2. IMPLEMENTACIÓN INICIO DE SESIÓN

El inicio de sesión es la puerta hacia el sistema académico, llegando a ser la parte que maneja la seguridad del sistema, la seguridad es conjunta tanto como en el servidor como en el lado del cliente, los datos de usuario y contraseña son enviados hacia el servidor el cual los procesara en el caso de no encontrar coincidencia con los datos de la base de datos nos mostrara asincrónicamente que el usuario o la contraseña es incorrecta.

La gran ventaja de trabajar con vuejs es la reactividad que existe en las vistas cuando se realiza una petición para autenticarnos se realiza el proceso de comparación de credenciales en la misma página no es necesario recargar otra para poder informar que las credenciales están mal, en caso de que las credenciales ingresadas si sean correctas se realizaran algunos procesos como el guardado del token y la redirección hacia la página de inicio tomando en cuenta el tipo de usuario que es ya que puede ser administrador, profesor o estudiante en la figura 26 se observa la interfaz gráfica del formulario inicio de sesión.

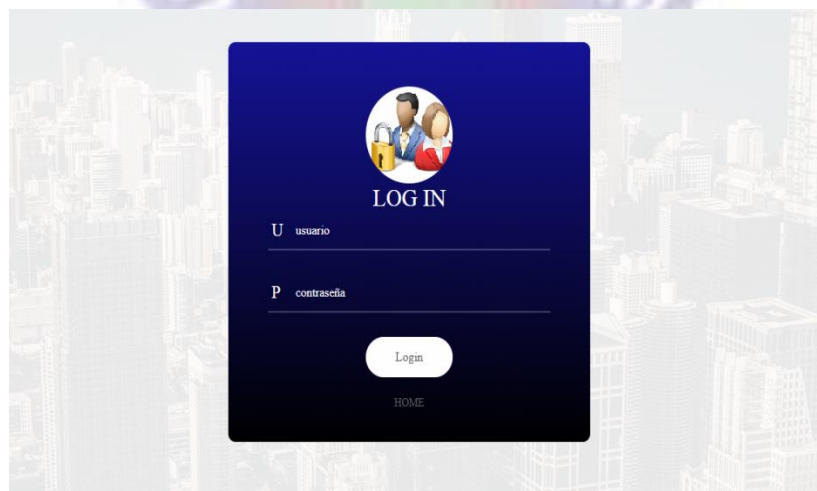


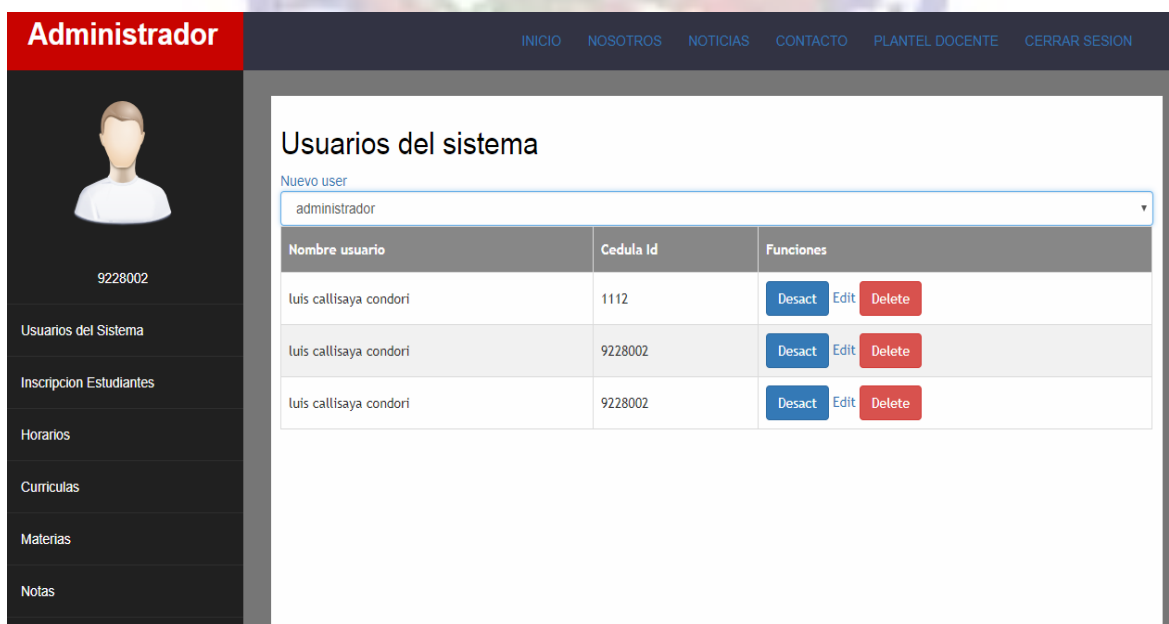
Figura 26. Implementación Inicio de Sesión

3.2.5.3. IMPLEMENTACIÓN ADMINISTRADOR

El usuario administrador luego de haber realizado la autenticación en el sistema será re direccionado hacia la página de inicio del administrador esta página contendrá datos principales de los usuarios podrá elegir entre si desea observar a los usuarios de tipo administrador, usuarios de tipo profesor, usuarios de tipo estudiante.

Con la ayuda de bootstrap conjuntamente con vuejs es que la tarea de renderizar los datos del usuario se logró de forma mucho más sencillas los datos de los usuarios se muestran de manera asíncrona ya que la pagina no tiene que volver a cargar para que muestre la tabla con los datos de los usuarios lo hace asíncronamente.

La página de inicio de sesión se la puede observar en la Figura 27 la cual muestra una interfaz muy sencilla pero con muchos métodos internos.



The screenshot displays the Administrator Dashboard. On the left is a dark sidebar with a red header 'Administrador' and a navigation menu including 'Usuarios del Sistema', 'Inscripción Estudiantes', 'Horarios', 'Curriculas', 'Materias', and 'Notas'. The main content area is titled 'Usuarios del sistema' and features a search bar with 'administrador' entered. Below the search bar is a table with three columns: 'Nombre usuario', 'Cedula Id', and 'Funciones'. The table contains three rows of user data, each with 'Desact', 'Edit', and 'Delete' buttons.

| Nombre usuario | Cedula Id | Funciones |
|------------------------|-----------|--|
| luis callisaya condori | 1112 | Desact Edit Delete |
| luis callisaya condori | 9228002 | Desact Edit Delete |
| luis callisaya condori | 9228002 | Desact Edit Delete |

Figura 27. Implementación Página Inicio Administrador

3.2.5.4. IMPLEMENTACIÓN PROFESOR

La vista de profesor es mucho más liviana ya que no realizara muchos procesos como en el caso del administrador toda la vista será autoajustable es decir que se podrá ajustar a cualquiera de los navegadores, cualquiera sea el tamaño que contengan aun así sea un navegador abierto desde un teléfono celular.

Con la ayuda de vuejs y los estilos es que se llegó a realizar una vista muy agradable y liviana en cuanto a carga de código para la renderización en el navegador contendrá opciones las cuales serán re direccionadas gracias al gestor de rutas vue-router todo este proceso igualmente se lo realizara de forma asíncrona es decir muchos de las funciones no necesitan recargar la página para mostrar datos la reactividad de vuejs lo hace dinámico y entretenido como podemos observar en la Figura 28.



The screenshot displays the 'Inicio Profesor' (Professor Home) page. At the top, there is a navigation bar with links: INICIO, NOSOTROS, NOTICIAS, CONTACTO, PLANTEL DOCENTE, and CERRAR SESION. The main content area is titled 'REGISTRO ESTUDIANTES INSCRITOS' (Enrolled Students Register). Below the title is a table with columns: Fecha, Nombre, Personal, and operacion. The table contains two rows of data. On the left side, there is a sidebar with a user profile icon, the ID '112233', and buttons for 'PROFESOR' and 'GESTIONAR CONTRASEÑA'.

| Fecha | Nombre | Personal | operacion |
|--------------------------|--------------------------|----------|-----------|
| 2018-04-05T22:00:00.000Z | marcos palenque jimenes | 9228002 | Observar |
| 2018-04-05T22:00:00.000Z | adolfo marquez fernandez | 9228002 | Observar |

Figura 28. Implementación Pagina Inicio Profesor

3.2.5.5.IMPLEMENTACIÓN ESTUDIANTE - TUTOR

Tanto el estudiante como el tutor tienen acceso a la página de inicio estudiante – tutor, se necesita autenticarse con las credenciales necesarias que la dirección otorgo a cada usuario, luego de realizar dicho proceso si todo es correcto este le re direccionara hacia la página de inicio del estudiante – tutor.

Aquí es donde el tutor podrá realizar el seguimiento académico constantemente de su hijo e hija ya que las notas de cada bimestre como también las notas generales estarán contenidas en esta página y si existiese cualquier cambio por parte del administrador en tiempo real es que la página se actualizara ya que está constantemente esperando si existe algún cambio por parte del administrador.

Las observaciones que un profesor puede realizar son la parte fundamental para el tutor ya que con ello podrá informarse de como es el comportamiento de su protegido en la unidad educativa de ese modo poder tomar alguna decisión esta implementación podemos observarlo en la figura 29.

| Materia | Primer | Segundo | Tercero |
|-------------------|--------|---------|---------|
| lenguaje | | | 67 |
| ciencias sociales | | | 76 |

Figura 29. Implementación Pagina Inicio Estudiante

4. CALIDAD Y SEGURIDAD

4.1. CALIDAD DEL SOFTWARE

La calidad del software es el grado con el que se mide un sistema, componente o proceso de un proyecto, al mismo tiempo verifica los requerimientos especificados y las necesidades o expectativas del cliente o usuario los cuales son la prioridad del sistema.

4.2. CALIDAD ISO/IEC 9126

La calidad del software se evalúa teniendo en cuenta la etapa del desarrollo, se deben fijar la metas de la calidad tanto para el software final como para desarrollos intermedios y tener en cuenta que es imposible que las metas y criterios sean iguales para un software pequeño y un gran software empresarial. Se establecen categorías para las cualidades de la calidad externa e interna y calidad en uso del software, teniendo en cuenta estos 7 indicadores: funcionalidad, confiabilidad, utilidad, eficiencia, capacidad de mantenimiento, portabilidad y calidad en uso, que se subdividen a su vez en varios indicadores estas se pueden medir por métrica interna o externa.

4.2.1. FUNCIONALIDAD

Funcionalidad es la capacidad del software de cumplir y contemplar las funciones para satisfacer las necesidades primarias y secundarias cuando es utilizado en condiciones específicas. Ejemplo, lo que hace el software para satisfacer las necesidades sin tener en cuenta de cómo lo hace y cuando. La funcionalidad se divide en 5 criterios:

- Adecuación: La capacidad del software para proveer un adecuado conjunto de funciones que cumplan las tareas y objetivos especificados por el usuario.
- Exactitud: La capacidad del software para hacer procesos y entregar los resultados solicitados con precisión o de forma esperada.
- Interoperabilidad: La capacidad del software de interactuar con uno o más sistemas específicos.
- Seguridad: La capacidad del software para proteger la información y los datos de manera que los usuarios o los sistemas no autorizados no puedan acceder a ellos para realizar operaciones, y la capacidad de aceptar el acceso a los datos de los usuarios o sistemas autorizados.
- Conformidad de la funcionalidad: La capacidad del software de cumplir los estándares referentes a la funcionalidad La funcionalidad de un sistema no puede ser medido directamente, entonces corresponde derivar mediante otras medidas directas como el punto función, para esto se tiene la siguiente relación:

$$PF = Cuenta_Total * \left[\sum_{k=0}^n (Fi) + 0,01 + 0,65 \right]$$

El cálculo del punto función será muy útil para hallar una referencia hacia la calidad como a la funcionalidad con la que cuenta el sistema en la Tabla 6 podemos observar el cálculo

Tabla 6 Cálculo del Punto función

| Parámetros de Medida | Cuenta | Factor Complejidad | | | Total |
|---------------------------------|--------|--------------------|-------|----------|-------|
| | | Simple | Medio | Complejo | |
| Número de Entradas del usuario | 11 | 3 | 4 | 6 | 44 |
| Número de Salidas del usuario | 13 | 4 | 5 | 7 | 65 |
| Número de Peticiones de usuario | 10 | 3 | 4 | 6 | 40 |
| Número de Registros | 11 | 7 | 10 | 14 | 109 |
| Numero de Interfaces | 2 | 5 | 7 | 10 | 14 |
| | 47 | | | | 272 |

$\sum (F_i)$ puede ser de 1 a 14 los valores de ajuste de la complejidad según las respuestas a las siguientes preguntas:

1. ¿El sistema académico requiere copias de seguridad y recuperación fiables?
2. ¿Se requiere comunicación de datos?
3. ¿Existen funciones de procesamiento distribuido?
4. ¿Es crítico el rendimiento?
5. ¿Se ejecutará el sistema en un entorno operativo existente y fuertemente utilizado?
6. ¿Requiere el sistema entrada de datos interactiva?
7. ¿Requiere la entrada de datos interactiva que las transacciones de datos se lleven a cabo sobre múltiples pantallas u operaciones?
8. ¿Se actualizarán los datos de forma asíncrona?
9. ¿Son complejas las entradas, las salidas, los registros o las peticiones?
10. ¿Es complejo el procesamiento interno?
11. ¿Se ha diseñado código para que sea reutilizable?
12. ¿Se ha diseñado el sistema para soportar múltiples instalaciones en diferentes organizaciones?

Cada una de las preguntas anteriores es respondida usando la escala de rangos desde 0 (No importante o aplicable) hasta 5 (Absolutamente esencial) como se muestra en la Tabla 7.

Una vez que se han calculado los puntos de función se usa de forma analógica a las LDC como forma de normar a la productividad, calidad y otros atributos del software.

Tabla 7 Rangos de calificación Punto Función

| | | | | | |
|----------------|------------|----------|-------|---------------|----------|
| 0 | 1 | 2 | 3 | 4 | 5 |
| Sin influencia | Incidencia | Moderado | Medio | Significativo | Esencial |

$\sum (F_i)$: ($P_i=1$ hasta 14) son los valores de ajuste de complejidad.

**P1=5; P2=5; P3=4; P4=3; P5=5; P6=4; P7=4; P8=5; P9=3; P10=4; P11=3; P12=3;
P13=3; P14=5;**

$$\sum (F_i) = 5+5+4+3+5+4+4+5+3+4+3+3+3+5 = 56$$

$$PF = 240 * (0.65 + (0.01 * 56))$$

$$PF = 240.4 \text{ valor en punto función.}$$

Si consideramos el máximo valor de complejidad como $\sum (F_i) = 70$, se tiene:

$$PF = 240 * (0.65 + (0.01 * 70))$$

$$PF = 324$$

Entonces si $\sum (F_i)$ es considerada como el 100%, la relación obtenida entre los puntos será:

$$PF = 290.4 / 324 = 0.9$$

$$PF = 90\%$$

Por lo tanto la funcionalidad que tiene el sistema es del 90%, tomando en cuenta el punto de función máximo.

4.2.2. CONFIABILIDAD

La confiabilidad es la capacidad del software para asegurar un nivel de funcionamiento adecuado cuando es utilizando en condiciones específicas. La definición de confiabilidad en la **ISO/IEC 2382** es la habilidad de la unidad funcional de realizar una función requerida. En este caso la confiabilidad se amplía a sostener un nivel especificado de funcionamiento y no una función requerida.

La confiabilidad se divide en 4 criterios:

- Madurez. La capacidad que tiene el software para evitar fallas cuando encuentra errores. Ejemplo, la forma como el software advierte al usuario cuando no encuentra espacio suficiente el disco duro donde esta almacenando los datos.
- Tolerancia a errores. La capacidad que tiene el software para mantener un nivel de funcionamiento en caso de errores.
- Recuperabilidad. La capacidad que tiene el software para restablecer su funcionamiento adecuado y recuperar los datos afectados en el caso de una falla.
- Conformidad de la fiabilidad. La capacidad del software de cumplir a los estándares o normas relacionadas a la fiabilidad.

La confiabilidad del sistema es directamente proporcional a la calidad de sus componentes

$$R(t) = e^{-\lambda t}$$

Dónde:

R(t): Función de confiabilidad de un componente en el tiempo t.

λ : Taza constante de fallo.

T: Periodo de operación de tiempo.

Si n componentes funcionan independientemente conectados en serie y el i-esimo componente tiene confiabilidad $R_i(t)$, entonces la confiabilidad total es dada por:

$$R(t) = R_1(t) * R_2(t) * R_3(t) \dots R_n(t)$$

Si n componentes funcionan independientemente y actúan en paralelo y el i-esimo componente tiene confiabilidad $R_i(t)$, entonces la confiabilidad total es dada por:

$$R(t) = 1 - [1 - R_1(t)] * [1 - R_2(t)] * [1 - R_3(t)] * \dots * [1 - R_n(t)]$$

Realizando un análisis del sistema académico planteado se tiene el siguiente diagrama de transferencia:

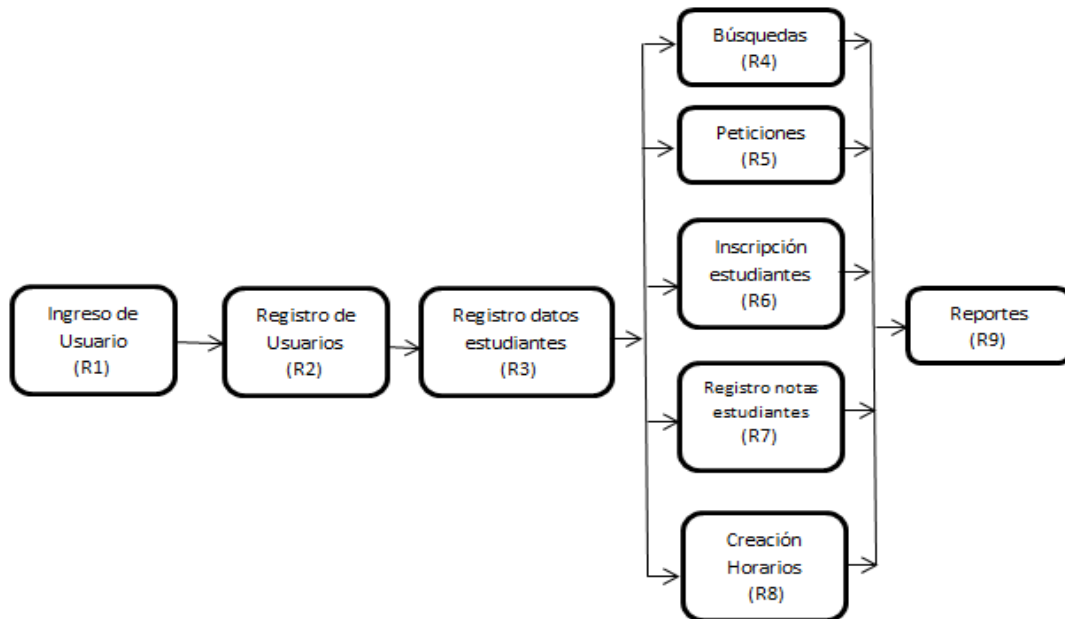


Figura 30. Diagrama de Transferencia

Seguidamente podemos realizar análisis de datos de cada una de las funciones descritas en la figura en la Tabla 8 encontramos el cálculo de confiabilidad que se le realiza al sistema.

Tabla 8 Calculo de confiabilidad

| Ri() | λ | T(semanas) | $e^{\lambda t}$ |
|------|-----------|------------|-----------------|
| R1() | 0.005 | 6 | 0.97 |
| R2() | 0.005 | 6 | 0.97 |
| R3() | 0.005 | 6 | 0.97 |
| R4() | 0.01 | 6 | 0.94 |
| R5() | 0.01 | 6 | 0.94 |
| R6() | 0.01 | 6 | 0.94 |
| R7() | 0.01 | 6 | 0.94 |
| R8() | 0.01 | 6 | 0.94 |
| R9() | 0.01 | 6 | 0.94 |

Entonces Si todos tienen la misma confiabilidad para cada uno de los componentes es $R(t)=$

$e^{\lambda(-0.01t)}$ la confiabilidad dentro de seis semanas será:

$$RT(t) = \{1 - [1 - R4(t)] * [1 - R5(t)] * [1 - R6(t)] * [1 - R7(t)] * [1 - R8(t)]\} * [R1(t)] [R2(t)] [R3(t)] [R9(t)]$$

$$RT(t) = 0.999 * 0.887$$

$$RT(t) = 0.887$$

$$RT(t) = 88.7\% \text{ de confiabilidad del sistema}$$

4.2.3. USABILIDAD

La usabilidad es la capacidad del software de ser entendido, aprendido, y usado en forma fácil y atractiva. Algunos criterios de funcionalidad, fiabilidad y eficiencia afectan la

usabilidad, pero para los propósitos de la ISO/IEC 9126 ellos no clasifican como usabilidad. La usabilidad está determinada por los usuarios finales y los usuarios indirectos del software, dirigidos a todos los ambientes a la preparación del uso y el resultado obtenido.

Las métricas de usabilidad son:

- Entendimiento. La capacidad que tiene el software para permitir al usuario entender si es adecuado, y de una manera fácil como ser utilizado para las tareas y las condiciones particulares de la aplicación. En este criterio se debe tener en cuenta la documentación y de las ayudas que el software entrega.
- Aprendizaje. La forma como el software permite al usuario aprender su uso. También es importante considerar la documentación.
- Operabilidad. La manera como el software permite al usuario operarlo y controlarlo.
- Atracción. La presentación del software debe ser atractivo al usuario. Esto se refiere a las cualidades del software para hacer más agradable al usuario, ejemplo, el diseño gráfico.
- Conformidad de uso: La capacidad del software de cumplir los estándares o normas relacionadas a su usabilidad.

Como se mencionó anteriormente la usabilidad es la capacidad de ser tanto entendido como aprendido y usado por el usuario final el cálculo de la usabilidad se la hace a través de datos obtenidos por usuarios del sistema en la tabla 9 podemos observar el análisis de datos obtenidos para el cálculo de la usabilidad.

Tabla 9 Calculo de Usabilidad

| Usuarios | Facilidad de Comprensión | Facilidad de Aprendizaje | Facilidad de Operación |
|-----------|--------------------------|--------------------------|------------------------|
| Usuario 1 | 90% | 95% | 100% |
| Usuario 2 | 90% | 94% | 95% |
| Usuario 3 | 98% | 94% | 90% |
| Usuario 4 | 94% | 95% | 96% |
| Promedio | 93% | 94.50% | 95.25% |

$$\text{Facilidad de uso} = (93\% + 94.50\% + 95.25\%) / 3 = 94.25\%$$

Finalmente podemos concluir que la facilidad de uso es de **94.25%**

4.2.4. CAPACIDAD DE MANTENIMIENTO

La capacidad de mantenimiento es la cualidad que tiene el software para ser modificado. Incluyendo correcciones o mejoras del software, a cambios en el entorno, y especificaciones de requerimientos funcionales.

- Capacidad de ser analizado. La forma como el software permite diagnósticos de deficiencias o causas de fallas, o la identificación de partes modificadas.
- Cambiabilidad. La capacidad del software para que la implementación de una modificación se pueda realizar, incluye también codificación, diseño y documentación de cambios.
- Estabilidad. La forma como el software evita efectos inesperados para modificaciones del mismo.
- Facilidad de prueba. La forma como el software permite realizar pruebas a las modificaciones sin poner el riesgo los datos.

- Conformidad de facilidad de mantenimiento. La capacidad que tiene el software para cumplir con los estándares de facilidad de mantenimiento.

Para calcular la mantenibilidad del sistema utilizaremos las medidas directas proporcionadas por la IEEE 982.1 – 1998 el cual sugiere un índice de madurez del sistema, que consiste en los cambios que producen en cada versión del producto para lo cual se tiene la siguiente relación.

$$\text{IMS} = [\text{MT} - (\text{Fa} + \text{Fb} + \text{Fc})] / \text{MT}$$

Dónde:

MT = Número de módulos en la versión actual.

Fa = Número de módulos en la versión actual que se han cambiado.

Fb = Número de módulos en la versión actual que se han añadido.

Fc = Número de módulos en la versión anterior que se han borrado en la versión actual.

Si el valor de IMS se aproxima a 1, el sistema empieza a estabilizarse.

Entonces los valores encontrados del sistema académico son:

$$\text{MT} = 6$$

$$\text{Fa} = 1$$

$$\text{Fb} = 0$$

$$\text{Fc} = 0$$

Reemplazando estos valores en:

$$\text{IMS} = [\text{MT} - (\text{Fa} + \text{Fb} + \text{Fc})] / \text{MT}$$

$$\text{IMS} = [6 - (1 + 0 + 0)] / 6$$

$$\text{IMS} = 0.83 * 100$$

$$\text{IMS} = 83\%$$

Por tanto puede indicarse que el sistema tiende a estabilizarse en un 83% y el 17% indica que aún no es estable debido a los cambios.

4.2.5. PORTABILIDAD

Es la capacidad que tiene el software para ser trasladado de un entorno a otro, cuenta con las siguientes métricas de calidad:

- Adaptabilidad. Es como el software se adapta a diferentes entornos especificados (hardware o sistemas operativos) sin que implique reacciones negativas ante el cambio. Incluye la escalabilidad de capacidad interna (Ejemplo: Campos en pantalla, tablas, volúmenes de transacciones, formatos de reporte, etc.).
- Facilidad de instalación. La facilidad del software para ser instalado en un entorno específico o por el usuario final.
- Coexistencia. La capacidad que tiene el software para coexistir con otro o varios software, la forma de compartir recursos comunes con otro software o dispositivo.
- Reemplazabilidad. La capacidad que tiene el software para ser reemplazado por otro software del mismo tipo, y para el mismo objetivo. Ejemplo, la reemplazabilidad de una nueva versión es importante para el usuario, la propiedad de poder migrar los datos a otro software de diferente proveedor.
- Conformidad de portabilidad. La capacidad que tiene el software para cumplir con los estándares relacionados a la portabilidad.

Además de estos aspectos también podemos considerar la facilidad de instalación, ajuste y adaptación al cambio dado por la siguiente formula:

$$GP=1-[CT/CRD]$$

Dónde:

GP = Grado de Portabilidad

CT = Costo de Transportar

CRD = Costo de Re- desarrollo

Si $GP > 0$, la portabilidad es más rentable que el re-desarrollo

Si $GP = 1$, la portabilidad es perfecta

Si $GP < 0$, el re-desarrollo es más rentable que la portabilidad.

Según el sistema académico planteado podemos reemplazar los datos en la formula general:

$$GP=1-[CT/CRD]$$

$$GP = 1-[75/1000]$$

$$GP = 0.9625$$

Por lo tanto el sistema puede transportarse de un entorno a otro en un 96% más rentable que el re-desarrollo.

4.2.6. EFICIENCIA

La eficiencia es el conjunto de atributos relacionados con la relación entre el nivel de desempeño del software y la calidad de recursos necesitados bajo condiciones establecidas.

Las métricas de calidad son:

- Comportamiento en el tiempo. Atributos del software relativos a los tiempos de respuesta y de procesamiento de los datos.
- Comportamiento de recursos. Atributos del software relativos a la cantidad de recursos usados y la duración de su uso en la realización de sus funciones.

Bajo estos atributos evaluamos la eficiencia mostrada en la tabla 10 de forma global.

Tabla 10 Calculo de la Eficiencia

| Atributo | CrE(Xi) | IEi% |
|---|---------|------|
| Eficiencia | CNV | 78.5 |
| Desempeño | CNV | 80 |
| Páginas de Acceso Rápido | CDP | 80 |
| Accesibilidad | CNV | 77.5 |
| Accesibilidad de Información | CNV | 75 |
| Soporte nueva Versión | CMN | 60 |
| Legitimidad al activar y desactivar usuario | CNV | 90 |
| Imagen dinámica | CB | 100 |
| Legitimidad Global | CDP | 80 |
| Accesibilidad de Ventanas | CNV | 80 |
| Cantidad de vistas considerando tipo user | CMN | 60 |
| Versión sin bootstrap | CB | 100 |

4.3. SEGURIDAD

4.3.1. SEGURIDAD DE LA BASE DE DATOS

La seguridad del sistema se divide en tres partes Encriptación de datos, Autenticación de Usuarios, Manejo de Sesiones que se describirán a continuación.

4.3.1.1. ENCRIPCIÓN DE DATOS

Consiste en transformar un valor legible en uno ilegible mediante un algoritmo lo cual permite que cierta información sea inútil para aquellas personas que por alguna razón tengan acceso al sistema o a la base de datos. En nodejs existe la librería encryption-jwt se utiliza la función MD5 que es una encriptación segura el cual algun individuo que intentara rastrear la contraseña ingresada no podrá descifrar la misma, es una función hash irreversible (de un sólo sentido), gracias a esta librería que es compatible con nodejs es que

obtenemos seguridad para el manejo de datos en el sistema. Por esto mismo no hay problema de que alguien pueda acceder al campo encriptado de la base de datos.

El sistema Web de Seguimiento académico se encuentra implementado en el lenguaje JavaScript y gestor de base de datos postgresql hace uso de express para el levantamiento del servidor la base de datos cuenta con un usuario y una contraseña el cual son un requisito imprescindible a la hora de realizar alguna petición hacia la base de datos siendo este una seguridad para gestionar los datos.

4.3.1.2. AUTENTICACIÓN DE USUARIOS

Un elemento importante para la seguridad del sistema además de ser un requisito para ingresar al sistema, la autenticación de usuarios consta de un usuario y una contraseña variables definidas en el momento del primer uso de la cuenta en el sistema.

4.3.1.3. MANEJO DE SESIONES

Las sesiones son un tipo de variable que tiene como tarea guardar el estado de un usuario en particular. Además el uso de sesiones permite restringir el acceso a un url, puesto de no existir una sesión, el sistema re direccionará a la ventana de Formulario para autenticación de usuarios. En el momento del ingreso se le asigna una sesión al usuario mediante su nombre de usuario y su contraseña, esta sesión guarda los datos del usuario identificando el tipo de usuario para determinar si el usuario es administrador, profesor o estudiante.

Cada una de las rutas en el sistema realiza el proceso de verificación si el usuario tiene los permisos para poder ingresar a cierta información del sistema en caso de no tener permisos necesarios nos re direccionara hacia la página inicial que nos corresponde.

5. CONCLUSIONES Y RECOMENDACIONES

Luego de haber culminado todas las fases de la metodología (OOHDM), del mismo modo habiendo concluido la parte del desarrollo del proyecto Sistema de Información y Seguimiento Académico Colegio Nacional Mixto Bicentenario Libertador Simón Bolívar, se llega a las siguientes conclusiones y recomendaciones.

5.1. CONCLUSIONES

- La implementación del sistema servirá como herramienta de gestión que ayudará a mejorar el tratamiento de datos de la Unidad Educativa.
- La aplicación permite la realización del manejo de datos y operación con los mismos, tratando con esto que toda la gestión de datos de la unidad educativa se la realice de forma ordenada como de forma segura.
- La herramienta sendgrid es la encargada de realizar notificaciones directas a los padres de familia ayudara mucho en el monitoreo de la conducta del estudiante realizado por el tutor este a su vez ayudara a la comunicación entre tutor y profesor.
- El sitio web se encuentra en dos servidores distintos en uno se encuentra la parte de la comunicación con la base de datos y en la otra se encuentra toda la parte de la interfaz a esta forma de trabajo también se la conoce como backend y frontend.

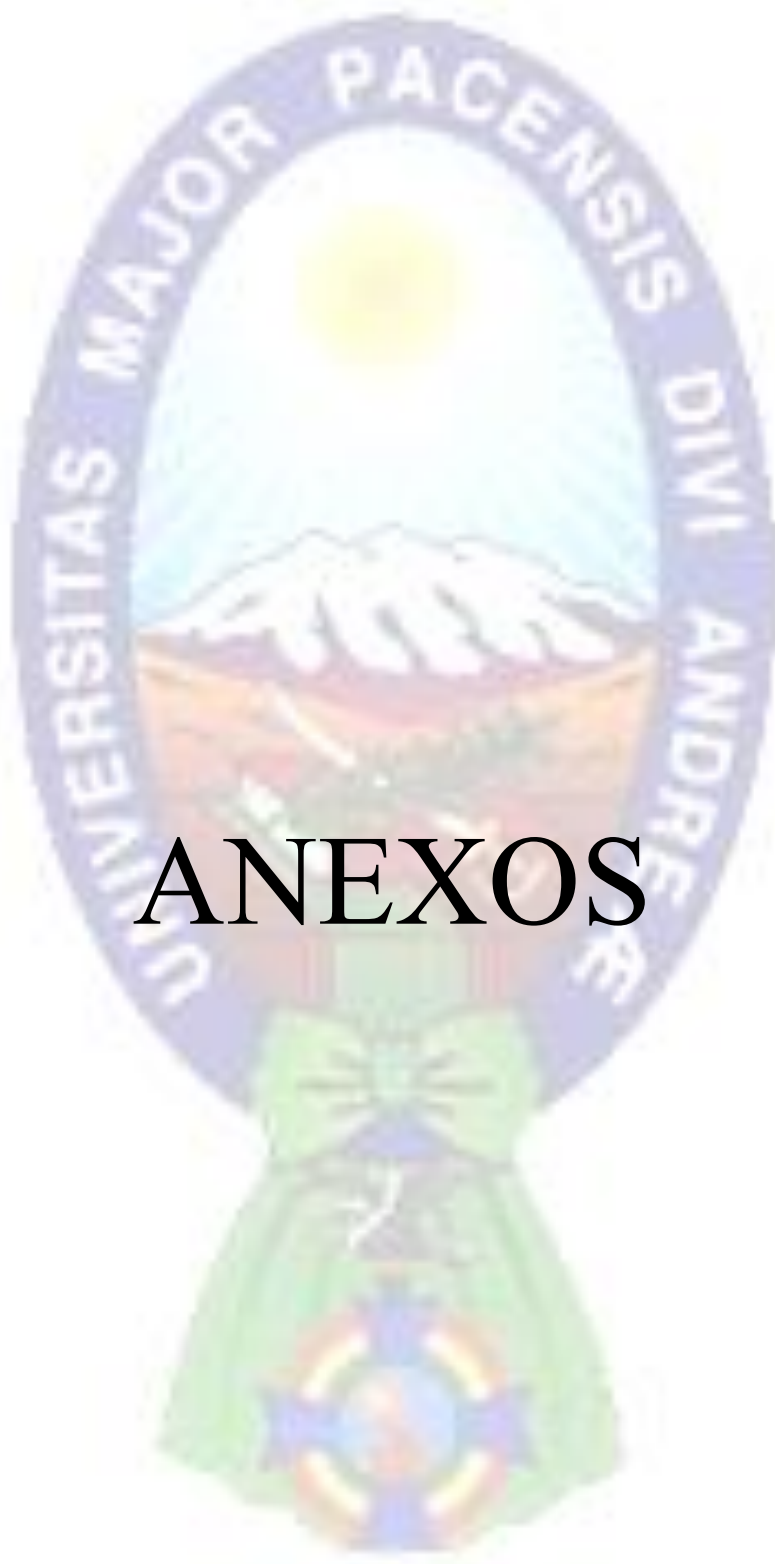
5.2. RECOMENDACIONES

Las siguientes recomendaciones contribuirían en la ampliación del nivel de alcance del presente proyecto.

- Se recomienda la incorporación de un módulo de control de personal por medio de biométrico ya que con ello se da mucha claridad de los datos.
- Para agregar cualquier actividad en tiempo real como ser examen en línea, videoconferencia, chat entre otros se recomienda hacer uso de la librería socket.io al ser muy compatible con nodejs.
- Se recomienda la incorporación de normas de manejo de datos de instituciones educativas para que de este modo no exista ningún problema con el ente mayor que es la distrital.
- Se recomienda incorporar un método de encriptación con licencia para que de este modo la encriptación tenga soporte en cada cambio que exista con la tecnología.

Bibliografía

- Brendan Eich. (18 de 03 de 2018). *Lenguaje Javascript*. Recuperado el 05 de 05 de 2018, de Lenguaje Javascript: <https://www.javascript.com>
- Express - Infraestructura de aplicaciones web Node.js. (12 de 03 de 2018). Recuperado el 05 de 05 de 2018, de Express - Infraestructura de aplicaciones web Node.js: <http://expressjs.com/es/>
- Colombia, M. d. (11 de 02 de 2011). *OOHDM*. Recuperado el 03 de 03 de 2018, de OOHDM: <https://oohdm.wordpress.com/2011/02/01/hello-world/>
- Google. (24 de 04 de 2018). *Framework Nodejs*. Recuperado el 03 de 05 de 2018, de Framework Nodejs: <https://nodejs.org/es/>
- Grupo desarrollo open sour. (14 de 01 de 2017). *Vue.js - The Progressive JavaScript Framework*. Recuperado el 05 de 05 de 2018, de Vue.js - The Progressive JavaScript Framework: <https://vuejs.org/>
- Lapuente, M. J. (10 de 06 de 2016). *Modelo OOHDM - Hipertexto*. Recuperado el 05 de 03 de 2018, de Modelo OOHDM - Hipertexto: www.hipertexto.info/documentos/oohdm.htm
- open source. (10 de 03 de 2017). *Manejador de paquetes NPM*. Recuperado el 05 de 05 de 2018, de Manejador de paquetes NPM: <https://www.npmjs.com>
- PostgreSQL Global Development Group. (29 de 01 de 1997). *El mundo de base de datos de codigo abierto*. Recuperado el 05 de 05 de 2018, de El mundo de base de datos de codigo abierto: <https://www.postgresql.org/>
- Pressman, R. S. (2010). *Ingenieria de Software Un Enfoque Practico*. Mexico: McGRAW-HILL INTERAMERICANA EDITORES, S.A. DE C.V.
- Sanchez, K. (24 de 05 de 2017). *Metodologia OOHDM*. Recuperado el 03 de 03 de 2018, de Metodologia OOHDM: <https://prezi.com/xohf4fhthjv/metodologia-oohdm>
- Schmuller, J. (2001). *Aprendiendo Uml En 24 Horas*. Zaragoza: Pearson Spanish; Edición: Pap/Cdr (1 de diciembre de 2001).
- Universidad de Alicante. (01 de 10 de 2017). *Modelo vista controlador (MVC)*. Recuperado el 02 de 04 de 2018, de Modelo vista controlador (MVC): <https://si.ua.es/es/documentacion/asp-net-mvc-3/1-dia/modelo-vista-controlador-mvc.html>
- Wikipedia Enciclopedia. (23 de 01 de 2018). *Transferencia de Estado Representacional*. Recuperado el 06 de 04 de 2018, de Transferencia de Estado Representacional: https://es.wikipedia.org/wiki/Transferencia_de_Estado_Representacional



ANEXOS

ANEXO A. MANUAL DE USUARIO

1. INTRODUCCIÓN

Empezar a trabajar con una nueva aplicación web es como intentar orientarse en una ciudad donde no ha estado nunca. Por ejemplo, en la ciudad de Nueva York, una vez se comprende la cuadrícula de calles numeradas, se puede averiguar de forma rápida donde esta cualquier dirección.

La aplicación de Sistema de seguimiento académico, está pensada y diseñada para facilitar a los usuarios del colegio Nacional Mixto Bicentenario Libertador Simón Bolívar, en los mecanismos de gestión de datos tanto de estudiantes, personal administrativo y profesores, actualmente existen tres roles para acceder al sitio web: usuario estudiante, usuario profesor y usuario administrador.

2. PORTAL DE ENTRADA

Para acceder al Sistema de Seguimiento académico Colegio Nacional Mixto Bicentenario Libertador Simón Bolívar primeramente ingrese al siguiente enlace:

<http://www.simonbolivarweb.info>

Al dar click en la Url podrá ver el portal de entrada a la aplicación, parte en la que se cuenta con la información de la unidad educativa como ser: misión y visión, noticias y eventos, plantel docente y contacto en la figura 1 se observa la interfaz de inicio del sitio web.

El enlace con más importancia será el de “INICIO DE SESIÓN” ya que este es la puerta hacia la administración del sistema para poder ingresar hacia el formulario de inicio de sesión primeramente haga click en el enlace de “INICIO DE SESIÓN” este le redireccionara hacia la página que le pedirá las credenciales así como se ve en la figura 1.

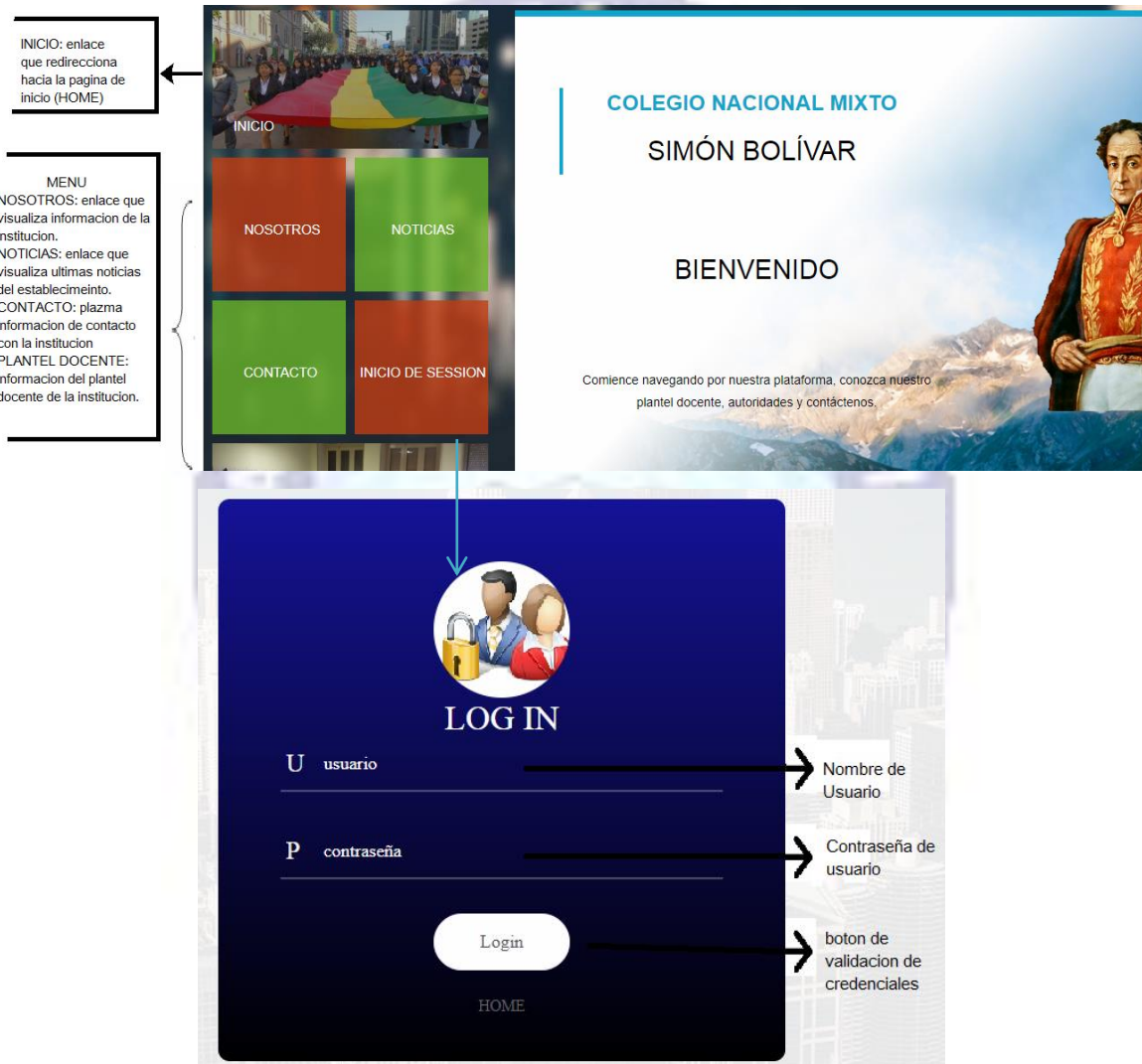


Figura 1. Página inicio

Para acceder al sistema se hace uso de las credenciales el nombre de usuario y la contraseña. En la parte inferior del formulario puede hacer click en Login para ingresar el sistema le re direccionara automáticamente según su tipo de cuenta hacia administrador, estudiante o profesor.

IMPORTANTE

- En caso que el usuario no cuente con las credenciales de acceso debe ponerse en contacto con la administración del sistema.
- En caso de que el usuario este deshabilitado tendrá que apersonarse en administración para recibir ayuda.

3. ENTENDER LA PERSONALIDAD DEL SISTEMA

Antes de pasar a una explicación detallada de la interfaz de usuario, quisiéramos ofrecerle una percepción del estilo o personalidad del Sistema Académico. La primera vez que observe la ventana principal del Sistema Académico expuesto anteriormente todo parecerá muy simple. Después se preguntara como adicono nuevos usuarios, ¿cómo inscribir a nuevos y antiguos estudiantes?, usted vera un menú en la parte izquierda que contiene todas las opciones que se necesita para administrar los datos requeridos.

4. USUARIO ADMINISTRADOR

El usuario administrador encargado de realizar la administración de muchas funciones como ser: gestión de cuentas de usuario, inscripción de nuevos y antiguos estudiantes, creación de nuevo tutor, inserción de notas de estudiantes, creación de horarios.

4.1. NUEVA CUENTA DE USUARIO

Si se desea registrar un nuevo usuario administrador, profesor o estudiante pulse en el enlace nuevo usuario como se observa en la figura 2 con ello accederá al formulario.

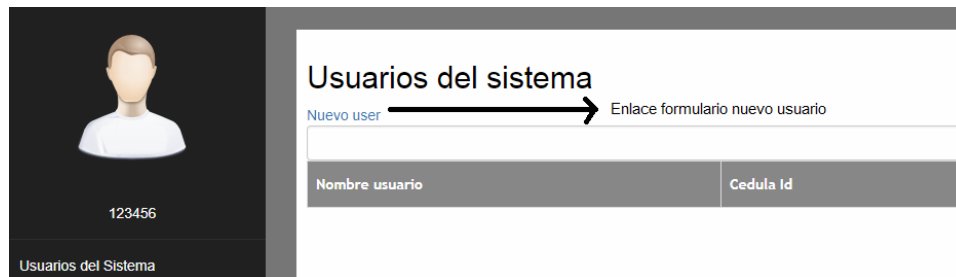


Figura 2. Nuevo usuario

El sistema académico permite crear nuevos usuarios dando a elegir el tipo de usuario a crear administrativo, profesor o estudiante todos los campos del formulario se encuentran estrictamente con funciones de validación de datos en caso de ingresar algún dato incorrecto el sistema anunciara que existe un dato erróneo en la figura 3 observamos el formulario de creación de nuevo usuario.

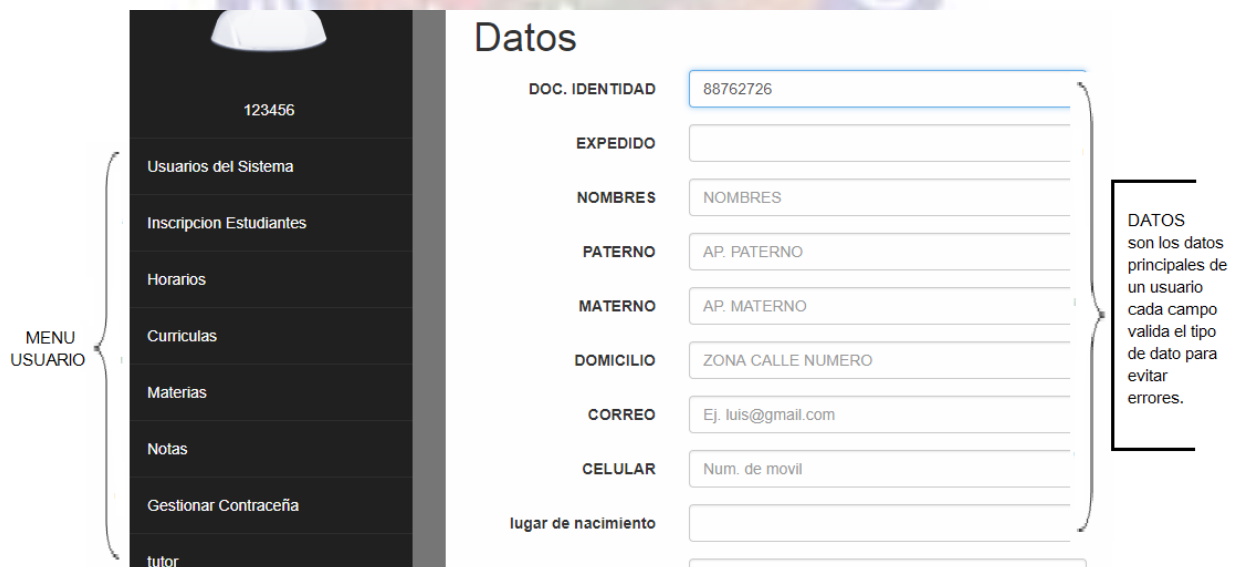


Figura 3. Registro de usuario

4.2. GESTIONAR Y LISTAR USUARIOS

Cada usuario es identificado por el usuario y contraseña que poseen, todos los usuarios al inicio se encuentran habilitados para poder ingresar al sistema, el sistema nos ayuda a gestionar estas cuentas como ser eliminar, editar y desactivar una cuenta por cierto tiempo en la figura 4 se observa las cuentas tipo estudiante y las operaciones que se pueden realizar con las mismas.

Selección tipo de usuario ← estudiante

USUARIOS TIPO ESTUDIANTE

| Nombre usuario | Cedula id | Funciones |
|--------------------------|-----------|--------------------|
| marcos palenque jimenes | 22 | Desact Edit Delete |
| maria ramirez flores | 33 | Desact Edit Delete |
| adoifo marquez fernandez | 44 | Desact Edit Delete |
| kamilia jimenes ramires | 4040 | Desact Edit Delete |
| marcelo quizpe quizpe | 789 | Desact Edit Delete |

Annotations: ELIMINAR (Delete), EDITAR (Edit), DESACTIVAR (Desact), ACTIVAR (missing button)

Figura 4. Gestión de usuarios

4.3. INSCRIPCIÓN DE ESTUDIANTES

En el menú principal izquierdo se encuentra la opción más importante inscripción de nuevos como antiguos estudiantes, un estudiante al ser nuevo en el establecimiento educativo ingresara como nuevo usuario al Sistema Académico, se tendrá la opción de elegir el paralelo y nivel al cual inscribirse, contrariamente un estudiante antiguo solo tendrá la opción de inscribirse en el nivel siguiente a su inscripción anterior. El proceso de inscripción de estudiantes se la programo para que sea de forma muy sencilla, solo ingresar el número de cedula de identidad del estudiante para que nos muestre toda la información del mismo si no existió ningún cambio en sus datos pues se finaliza la inscripción en la figura 5 se observa la interfaz de inscripción de nuevo estudiante.

INSCRIPCIONES

CI ESTUDIANTE: 22 ci estudiante renderiza automaticamente

Estudiante: marcos palenque jimenes Datos automaticos

Tutor: anastasio carrasco limachi

NIVEL: 2 A

FECHA: 25/06/2018

OBSERVACIONES: observaciones

Inscribir Realizar inscripcion

Figura 5. Formulario inscripción

4.4. CREACIÓN TUTOR

El menú izquierdo contiene todas las opciones necesarias por el administrador en ella se encuentra la opción de nuevo tutor, haga click sobre la opción este le llevara al formulario de creación de nuevo tutor como se observa en la figura 6, dicho formulario se encuentra estrictamente validado, en caso de ingresar algún dato erróneamente el sistema informara que existe error en algún campo del formulario, el correo electrónico es un campo muy importante a este correo se enviaran las notificaciones que algún profesor haga hacia su hijo o hija con ello podrá hacer un seguimiento del comportamiento del estudiante.

Ingreso de Tutor

Datos

DOC. IDENTIDAD: Num. Cedula Identidad

NOMBRES: NOMBRES

PATERNO: AP. PATERNO

MATERNO: AP. MATERNO

OCUPACION: ocupacion

CORREO: Ej. luis@gmail.com

CELULAR: Num. de movil

MAYOR_GRADO: Ej. 5º basico

FORMULARIO datos personales tutor de estudiante

Figura 6. Creación tutor

4.5.INSERCIÓN NOTAS DE ESTUDIANTE

La parte más delicada del sistema llega a ser las notas de cada estudiante ya que serán los datos que los usuarios finales demandaran, como el usuario estudiante, usuario tutor y usuario profesor, para realizar la inserción de datos de cada estudiante primeramente diríjase al menú izquierdo, haga click en la opción “Notas” seguidamente vaya al enlace “registrar nuevas notas”. Una vez en la página de registro de notas tendrá algo parecido a la figura 7, en ella usted podrá adicionar nuevas notas primeramente seleccione el paralelo, luego seleccione la materia a la cual pertenecen las notas y por ultimo seleccione el periodo al cual corresponde con ello se obtendrá una lista de todos los estudiantes inscritos en dicho paralelo y tendrá la opción de adicionar notas a cada uno de los estudiantes, en caso de realizar una adición de notas erróneas solo vuelva a ingresar la nota correcta esta se sobrescribirá sobre la nota anterior.

localhost:8080 dice
adicionado

PLANTEL DOCENTE CERRAR SESION

Aceptar

REGISTRO NOTAS

Alerta de adición de notas satisfactoria

1 A seleccion de paralelo

matematica seleccion de materia

II periodo seleccion de periodo

| Fecha | Nombre | Nota | operacion |
|--------------------------|-------------------------|----------------------------------|--|
| 2018-06-25T00:00:00.000Z | marcos palenque jimenes | <input type="text" value="100"/> | <input type="button" value="adicionar"/> |
| 2018-06-25T00:00:00.000Z | marcelo quispe quispe | <input type="text" value="51"/> | <input type="button" value="adicionar"/> |

Campo de nota Confirmacion adiccion

Figura 7. Inserción de notas

Para visualizar las notas actuales de los estudiantes ingrese a la opción “Notas” del menú izquierdo seleccione el paralelo al cual corresponda, seguidamente seleccione el estudiante el cual se desee visualizar las notas en la figura 8 se observa un ejemplo de cómo visualizar las notas de un estudiante en particular.

NOTAS ACTUALES
 registrar nuevas notas

1 A **seleccion paralelo**

marcos patenque jimenes **seleccin estudiante**

| Materia | Primer | Segundo | Tercero | Cuarto |
|--------------------|--------|---------|---------|--------|
| matematica | 20 | 51 | | |
| ciencias naturales | 20 | 61 | | |
| lenguaje | 29 | 90 | | |
| ciencias sociales | 60 | 80 | | |

Notas del estudiante

Figura 8. Notas del estudiante

4.6. CREACIÓN DE HORARIOS

La creación de horarios se lo debe realizar de forma muy ordenada ya que este puede llegar a confusión en parte de los usuarios finales estudiantes como profesores, para realizar la creación de horario primeramente debe dirigirse al menú izquierdo dar click en la opción de “Horarios” seguidamente agá click en el enlace “crear horario” primeramente debe seleccionar el paralelo al cual pertenecerá el horario nuevo, seguidamente elija uno por uno la materia, profesor, y aula correspondientes a la hora y el día, se observa un ejemplo en la figura 9. Después de haber seleccionado todas las opciones pulse el botón “Crear Horario” y se finalizara la adición del horario.

1 A **seleccion paralelo**

| Hora | Lunes | Martes | Miercoles | Jueves | Viernes |
|-----------|----------------|-------------|-------------------|-------------|-------------------|
| 2:00-2:35 | matematica | lenguaje | ciencias sociales | psicologia | ciencias sociales |
| | armando mamañi | elija profe | elija profe | elija profe | elija profe |
| | aula-123 | aula-321 | aula-321 | aula-321 | aula-112 |

seleccion de materia
 seleccion de profesor
 seleccion aula

Figura 9. Creación de horario

5. USUARIO PROFESOR

El usuario profesor es un usuario final del sistema ya que puede visualizar datos de estudiantes, posee las siguientes funciones: visualizar inscritos, notificar falta de estudiante.

5.1. LISTA DE INSCRITOS

Los datos de los estudiantes inscritos en el colegio son los cuales tienen importancia para un profesor, para poder acceder a la lista de los estudiantes inscritos primeramente debe dirigirse hacia el menú izquierdo ingresar a la opción “PROFESOR” seguidamente debe seleccionar el paralelo del cual quiere la lista de estudiantes, el sistema automáticamente plasmará la lista de los estudiantes del paralelo seleccionado como se muestra en la figura 10.



| Fecha | Nombre | Personal | operacion |
|--------------------------|-------------------------|----------|-----------|
| 2018-06-25T00:00:00.000Z | marcos palenque Jimenes | 123456 | Observar |
| 2018-06-25T00:00:00.000Z | marcelo quispe quispe | 123456 | Observar |

Figura 10. Lista de estudiantes

5.2. NOTIFICACIÓN FALTA DE ESTUDIANTE

Siendo el papel más importante por parte del profesor notificar faltas y errores cometidos por el estudiante, estas notificaciones viajarán directamente al correo del tutor de este modo el tutor se encontrará informado de las fallas de su hijo o hija, para realizar la observación a un estudiante primeramente debe dar click al enlace “Observar” que cada estudiante posee, este redireccionará hacia el formulario para poder observar al estudiante específico, en la figura 11 se observa el formulario de notificaciones.

OBSERVACION

Estrudiante: **marcospalequejimenes** → **Nombre de estudiante**

ingrese su observacion → **Descripcion de la observacion**

FECHA
 dd/mm/aaaa **Fecha de obs.**

Figura 11. Notificaciones

6. USUARIO ESTUDIANTE

El estudiante llega a denominarse el usuario final del sistema posee las siguientes funciones: visualizar notas por periodo, acceso al horario designado por dirección, observaciones que posee.

6.1. NOTAS ESTUDIANTE

Para acceder a las notas del estudiante primeramente debe seleccionar la opción “NOTAS” que se encuentra en el menú izquierdo este le re direccionara hacia una tabla con todas las notas que posee en la gestión actual.

6.2. HORARIO DE ESTUDIANTE

El horario se le asigna al estudiante en el momento de realizar la inscripción, los pasos siguientes llevan a observar el horario que posee un estudiante, primeramente debe haber realizado el logeo hacia el sistema, seguidamente debe dirigirse al menú izquierdo y pulsar el enlace “HORARIO” este nos plasmara el horario el cual se le asignó al estudiante.

6.3. OBSERVACIONES

Para visualizar las observaciones debe presionar el enlace “OBSERVACIONES” que se encuentra en el menú izquierdo este nos llevara hacia el registro de observaciones hechas hacia su persona como se visualiza en la figura 12.

| OBSERVACIONES | | |
|----------------------|--------------------------|------------------------|
| Profesor | Fecha | Observacion |
| armando mamani apaza | 2018-06-09T00:00:00.000Z | no llego a clases |
| armando mamani apaza | 2018-06-02T00:00:00.000Z | mala ortografia |
| armando mamani apaza | 2018-06-23T00:00:00.000Z | observacion de retraso |

Figura 12. Observaciones estudiante

ANEXO B. MANUAL TÉCNICO

1. INTRODUCCIÓN

Este manual describe los pasos necesarios para cualquier persona que tenga ciertas bases de sistemas pueda realizar la instalación del sistema creado para administración de la base de datos o cambio de plataforma.

Es importante tener en cuenta que en el presente manual se hace mención a las especificaciones mínimas de hardware y software para la correcta instalación del sistema.

2. REQUERIMIENTOS MÍNIMOS

Requerimientos mínimos de hardware

- Procesador : 1 Core
- Memoria RAM : 2 Gigabytes (GB)
- Disco Duro : 500 Gb.

3. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

JAVASCRIPT

Es el lenguaje para trabajar páginas web ofreciendo la ventaja de trabajar conjuntamente con HTML. Las ejecuciones son realizadas en el backend (NODEJS) y en el frontend (VUEJS) es el encargado de recibir los resultados de recibir la ejecución. Si el frontend realiza una petición, se ejecuta el intérprete javascript y se genera el contenido de manera dinámica, permite la conexión con la base de datos el cual es Postgresql. Este lenguaje de programación puede ser ejecutado en gran mayoría de los sistemas operativos y la puede interactuar con servidores WEB populares.

POSTGRESQL

Es un manejador de base de datos, el cual permite múltiples hilos y múltiples usuarios, fue desarrollado como software libre.

Se la puede usar sobre varias plataformas es muy utilizado sobre el sistema operativo Linux. Es libre para uso de servidores web.

4. BASE DE DATOS

Para la base de datos se usó el gestor postgresql, siendo un gestor de base de datos que plasma la información en formato JSON el cual da ventaja en la parte de velocidad como de la seguridad, la estructura de la base de datos se la plasma a continuación en la figura 1.

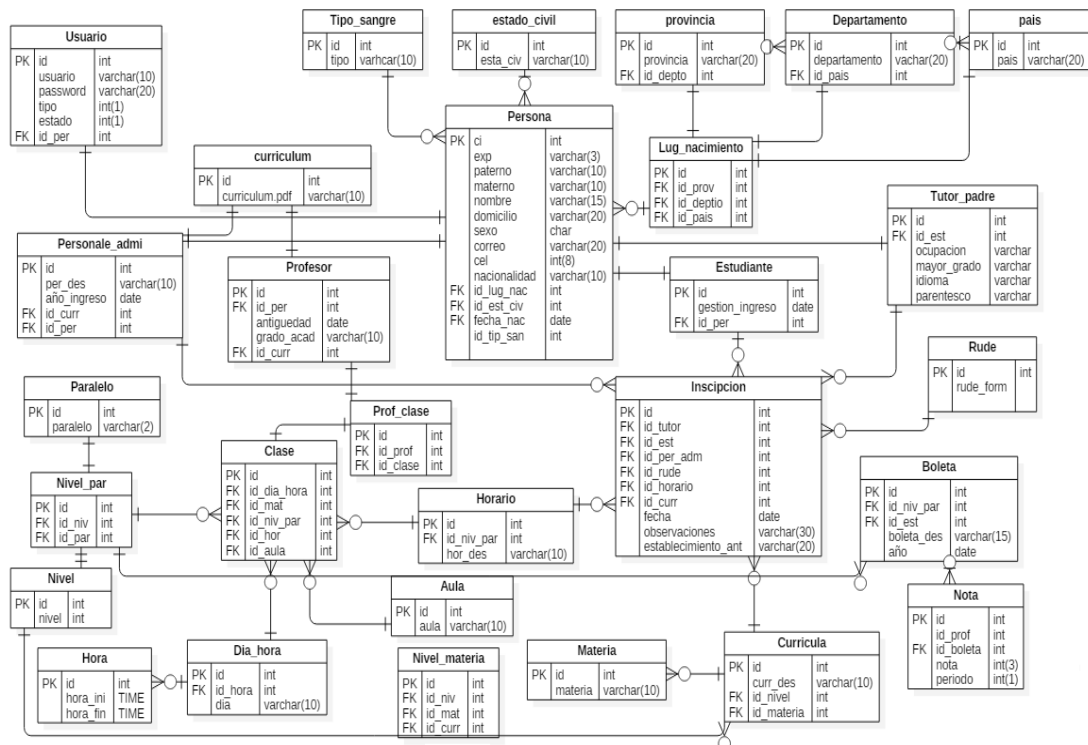


Figura 1. Diagrama Relacional

5. INSTALACIÓN DEL SERVIDOR

Primeramente instalar al servidor nodejs en la versión 4.2. o superior que será en servidor del sistema para la instalación seguir los siguientes pasos.

```
#sudo apt-get install build-essential libssl-dev
#npm ls-remote
#npm install 5.3.0
#npm use 4.2.4
#npm ls
```

Seguidamente se debe realizar la instalación de GIT versionado de código para ello se debe seguir los siguientes pasos:

```
#sudo apt-get install git
```

En la base de datos se debe crear una base de datos llamada todos con el usuario postgres y la clave “123456” con ello es suficiente para tener la base de datos configurada.

para obtener el proyecto se debe clonar el proyecto que se encuentra en el repositorio github.

```
#git clone https://github.com/luluisyo/front.git
#git clone https://github.com/luluisyo/back.git
```

Como último paso en el sistema se debe ingresar al directorio front se debe ejecutar el siguiente comando.

```
#npm install
#npm start
```

Del mismo modo ingresando en el directorio back se debe ejecutar el siguiente comando.

```
#npm install
#npm start
```

Con ello se levantara el servidor tanto del servidor como del cliente solo quedara verificar en el navegador en el puerto 8081 si todo funciona de manera correcta.

The logo of the Universidad Mayor de San Andrés is an oval emblem. It features a central sun, a mountain range, and a figure holding a staff. The text "UNIVERSITAS MAJOR PACENSIS DIVI ANDREAE" is inscribed around the perimeter. Below the oval is a green banner with a blue and red emblem.

DOCUMENTACIÓN

La Paz, 03 Julio de 2018

Señora:

LIC. MENFY MORALES RIOS

DIRECTORA

CARRERA DE INFORMÁTICA

FAC. CIENCIAS PURAS Y NATURALES

UNIVERSIDAD MAYOR DE SAN ANDRÉS

Presente

Ref. **AVAL PARA LA DEFENSA DE PROYECTO DE GRADO**

De mi mayor consideración.

Mediante la presente, me dirijo a su autoridad y en calidad de Tutor Metodológico, para informar que luego de haber realizado el seguimiento de la Tesis de Grado titulado: “**SISTEMA DE INFORMACION Y SEGUIMIENTO ACADEMICO COLEGIO NACIONAL MIXTO BICENTENARIO LIBERTADOR SIMON BOLIVAR**”, presentado por el Univ. Edwin Luis Callisaya Condori con CI. 9228002 L.P., para optar al título de **LICENCIADO EN INFORMÁTICA MENCIÓN: INGENIERÍA DE SISTEMAS INFORMÁTICOS**.

En este sentido, expreso mi conformidad y aval respectivo para la defensa pública del Proyecto de Grado de acuerdo a reglamento vigente en la Universidad Mayor de San Andrés.

Sin otro particular, me despido con las consideraciones más distinguidas.

Atentamente.

M. Sc. Rosa Flores Morales
Tutor Metodológico

av/waqq
cc/archivo

La Paz, 05 Junio de 2018

Señora

M. Sc. Rosa Flores Morales

Tutor Metodológico

Presente.-

REF. CONFORMIDAD Y AVAL DE TESIS DE GRADO

De mi consideración.

Tengo a bien dirigirme a su persona para darle a conocer, que luego de efectuar el seguimiento a la estructura y contenido del de Proyecto de Grado, titulado “**SISTEMA DE INFORMACIÓN Y SEGUIMIENTO ACADÉMICO COLEGIO NACIONAL MIXTO BICENTENARIO LIBERTADOR “SIMÓN BOLÍVAR”**”, elaborado por el universitario Callisaya Condori Edwin Luis, con CI: 9228002 L.P., en calidad de Asesor, expreso mi conformidad con el contenido y la forma de trabajo, dando mi Aval, para que el postulante pueda realizar la defensa de Proyecto de Grado, para optar al título de **LICENCIATURA EN INFORMÁTICA MENSIÓN: INGENIERIA DE SISTEMAS INFORMÁTICOS**, de acuerdo a normas y reglamentos vigentes.

Sin otro particular, me despido de usted con las consideraciones más distinguidas
Atentamente.

M. Sc. Carlos Mullisaca Choque
Docente Asesor

Av/waqq
cc/archivo

**ACTA DE ENTREGA DEL SOFTWARE SISTEMA DE INFORMACIÓN Y
SEGUIMIENTO ACADÉMICO COLEGIO NACIONAL MIXTO BICENTENARIO
LIBERTADOR SIMÓN BOLÍVAR**

En las instalaciones de la UNIDAD EDUCATIVA SIMÓN BOLÍVAR, el 02 de julio de 2018, se reunieron para dar entrega y recibo del software SISTEMA DE INFORMACIÓN Y SEGUIMIENTO ACADÉMICO COLEGIO NACIONAL MIXTO BICENTENARIO DEL LIBERTADOR SIMÓN BOLÍVAR, el universitario EDWIN LUIS CALLISAYA CONDORI, con C.I. 9228002 LP., hace entrega formal, material y real del software a Lic. UBALDO QUISPE ALI, director unidad educativa Simón Bolívar, de acuerdo a las siguientes:

CONSIDERACIONES

- 1) La carrera de Informática de la Universidad Mayor de San Andrés aprobó que el universitario Callisaya Condori Edwin Luis realice el Sistema de Información y Seguimiento Académico Colegio Nacional Mixto Bicentenario Libertador Simón Bolívar para que lo presente como proyecto de grado con el propósito de que la Unidad Educativa cuente con un centralizador de su información.
- 2) Que el Director de la Unidad Educativa Lic. Ubaldo Quispe Ali manifestó su conformidad de contar con un sistema académico, y en tal sentido mediante comunicación de fecha 06 de noviembre de 2016 se izó un compromiso verbal.
- 3) Que las obligaciones que existen entre las partes con ocasión de la entrega del sistema académico se formaliza mediante este documento.

Por lo anterior, las partes

ACUERDAN:

Suscribir la presente Acta de Entrega del Software Sistema de Información y Seguimiento Académico Colegio Nacional Mixto Bicentenario Libertador Simón Bolívar con las siguientes características:

- | | |
|----------------------------|-----------------------------|
| a) VPS con S.O. Debian 9.0 | d) Plataforma frontend |
| b) Dominio Godaddy | e) Estructura base de datos |
| c) Plataforma backend | f) licencia Sengrid (SMTP) |

Para constancia de lo anterior, se firma el presente documento.

EDWIN LUIS CALLISAYA CONDORI
Universitario

Director Colegio Nacional Mixto
Bic. Lib. Simón Bolívar