



DISEÑO Y DESARROLLO DE SOFTWARE PARA EL MANEJO INTEGRAL DE
PERSONAL ADMINISTRATIVO: CONTROL DE ASISTENCIA, LIQUIDACIÓN Y
PAGO DE NÓMINA. ESTUDIANTES: REGISTRO ACADÉMICO, CONTROL DE
ASISTENCIA Y CONTROL DE PAGO MENSUAL DE PENSIONES DE LA
INSTITUCIÓN NUEVO COLEGIO LUSADI LTDA.

Memoria presentada como requisito para optar al título de Ingeniero Electrónico de la
Universidad Santo Tomás

JUAN SEBASTIÁN RODRÍGUEZ PÉREZ
LUIS FERNANDO VÁSQUEZ FIERRO

UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
PROYECTO DE GRADO
BOGOTÁ D.C.
2016



DISEÑO Y DESARROLLO DE SOFTWARE PARA EL MANEJO INTEGRAL DE
PERSONAL ADMINISTRATIVO: CONTROL DE ASISTENCIA, LIQUIDACIÓN Y
PAGO DE NÓMINA. ESTUDIANTES: REGISTRO ACADÉMICO, CONTROL DE
ASISTENCIA Y CONTROL DE PAGO MENSUAL DE PENSIONES DE LA
INSTITUCIÓN NUEVO COLEGIO LUSADI LTDA.

Este proyecto de grado ha sido dirigido
por el **Dr. Tyronne Mejía**
y el **Ing. Saith Rodríguez**

PRESENTADO POR:
JUAN SEBASTIÁN RODRÍGUEZ PÉREZ
LUIS FERNANDO VÁSQUEZ FIERRO

PRESENTADO A:
COMITÉ DE GRADO
FACULTAD DE INGENIERÍA ELECTRÓNICA

UNIVERSIDAD SANTO TOMÁS
FACULTAD DE INGENIERÍA ELECTRÓNICA
PROYECTO DE GRADO
BOGOTÁ D.C.
2016

El trabajo de grado titulado diseño y desarrollo de software para el manejo integral de personal administrativo: control de asistencia, liquidación y pago de nómina. estudiantes: registro académico, control de asistencia y control de pago mensual de pensiones de la institución NUEVO COLEGIO LUSADI LTDA., presentado por los estudiante Luis Fernando Vásquez y Juan Sebastián Rodríguez como requisito para optar al título de Ingeniero Electrónico de la Universidad Santo Tomás ha sido revisado y aprobado.

Director del proyecto

Dr. Tyronne Mejía

Co-Director del proyecto

Ing. Saith Rodríguez

Jurado 1

Jurado 2

A Dios por permitirme llegar a este momento, A mis padres por guiarme día a día con su infinito esfuerzo, apoyo y amor, a mi esposa e hijas por ser mi compañía incondicional y darme el ánimo para cada día superar los obstáculos de la vida.

Luis Fernando Vásquez Fierro.

A mis padres, Adíela y Luis Alfredo, a mis abuelos y familiares por su incondicional apoyo que ayudaron al desarrollo integral como persona durante estos años de estudio, siempre guiándome con sus consejos y palabras de aliento para seguir adelante en mi camino.

Juan Sebastián Rodríguez Pérez

CONTENIDO

CAPÍTULO 1	10
INTRODUCCIÓN	10
.....	10
1.1.PROBLEMA	11
1.2.ANTECEDENTES	12
1.3.JUSTIFICACIÓN	13
1.4.OBJETIVOS.....	14
1.4.1.OBJETIVO GENERAL.....	14
1.4.2 OBJETIVOS ESPECÍFICOS.....	15
1.5.FACTIBILIDAD	16
1.6.COMPONENTE HUMANISTICO	17
CAPÍTULO 2	18
MARCO TEÓRICO.....	18
.....	18
2.1.MARCO TEÓRICO.....	19
2.2.LENGUAJES DE PROGRAMACION	20
LENGUAJES MÁQUINA, ENSAMBLADORES Y DE ALTO NIVEL	20
FORTTRAN, COBOL, PASCAL Y ADA.....	22
BASIC, VISUAL BASIC, VISUAL C++, C# Y .NET.....	23
HISTORIA DE C Y C++	24
HISTORIA DE JAVA	25
LENGUAJE JAVA.....	26
HTML5	27
CSS	29
CSS3	29
JAVASCRIPT	30
PHP	30
APACHE	31

MySQL	32
FRAMEWORK LARAVEL	32
2.3.BASES DE DATOS	33
2.4.¿QUÉ SON LOS SERVIDORES?.....	33
SERVICIOS NECESARIOS EN UNA EMPRESA	35
RESPONSABILIDADES PARA ASUMIR.....	36
RIESGOS EN LA ADMINISTRACIÓN	36
AMBIENTES DE TRABAJO.....	37
2.5.SISTEMAS CONTABLES	37
SISTEMA CENTRALIZADOR O DE DIARIOS MÚLTIPLES	38
2.6.NÓMINA	41
OBJETIVOS DE LA PREPARACIÓN DE LA NÓMINA	41
2.7.LIQUIDACION DE NÓMINA.....	41
CAPITULO 3	47
METODOLOGÍA.....	47
3.1.DISEÑO METODOLÓGICO.....	48
3.2.Modelo de base de datos estructurada en el software	49
CAPÍTULO 4	50
EJECUCIÓN DEL PROYECTO	50
4.1.DESARROLLO	51
4.2.ESQUEMA GENERAL DEL SOFTWARE.....	52
4.3.IMPLEMENTACIÓN DEL FRAMEWORK	53
SERVIDOR XAMPP	54
FRAMEWORK (VISTAS BOOSTRAP)	54
CONTROLADORES.....	56
MIGRACIONES (CREACION DE TABLAS DE DATOS)	57
MODELOS	58
RUTAS.....	59
4.4.AUTENTICACION DE USUARIO EN EL SOFTWARE	60
4.5.REGISTRO PERSONAL	61
4.6.REGISTRO POR PERFILES	62
PERFIL ADMINISTRATIVO	63
4.7.BÚSQUEDA Y MODIFICACIÓN DEL PERFIL.....	64

4.8.MATRÍCULA DEL ALUMNO	65
4.9.REGISTRO ACADÉMICO	66
PROCEDIMIENTO DE CALIFICACIÓN EN LA INSTITUCIÓN	66
FUNCIONAMIENTO SEGÚN TIPO DE USUARIO	66
PROCEDIMIENTO DE ASIGNACIÓN DE MATERIAS EN EL SOFTWARE.....	67
PROCEDIMIENTO PARA EL REGUISTRO DE NOTAS EN EL SOFTWARE	68
4.10.ADMINISTRACION DE NÓMINA.....	69
4.11.CONTROL DE ASISTENCIA.....	72
GENERACIÓN DE CÓDIGOS DE BARRAS	73
VERSIÓN FINAL DEL CARNET	75
GENERACION BASE DE DATOS	75
IMPLEMENTACIÓN DEL CONTROL DE ASISTENCIA EN EL SOFTWARE	76
CAPÍTULO 5	77
RESULTADOS	77
.....	77
5.1.ASISTENCIA.....	78
SISTEMA DE MANEJO ANTERIOR AL SOFTWARE.....	78
MANEJO ACTUAL CON EL SOFTWARE	78
COMPARACIÓN DE MANEJO (ANTES / ACTUAL).....	78
TABLA 5.1. Comparación de manejo respecto al tiempo.	78
VENTAJAS DEL SOFTWARE	79
5.2.REGISTRO DE NOTAS	79
SISTEMA DE MANEJO ANTERIOR AL SOFTWARE	79
MANEJO ACTUAL CON EL SOFTWARE.....	79
COMPARACIÓN DE MANEJO (ANTES / ACTUAL).....	79
TABLA 5.2. Comparación de manejo respecto al tiempo.	79
VENTAJAS DEL SOFTWARE	80
5.3.PAGOS	80
SISTEMA DE MANEJO ANTERIOR AL SOFTWARE	80
MANEJO ACTUAL CON EL SOFTWARE	80
COMPARACIÓN DE MANEJO (ANTES / ACTUAL).....	80
TABLA 5.3. Comparación de manejo respecto al tiempo.	80
VENTAJAS DEL SOFTWARE	81

CAPÍTULO 6	82
CONCLUSIONES	82
.....	82
CONCLUSIONES	83
CAPÍTULO 7	84
BIBLIOGRAFÍA.....	84

CAPÍTULO 1

INTRODUCCIÓN

1.1. PROBLEMA

La institución Nuevo Colegio Lusadi Ltda. Dedicada a la formación de estudiantes, en los grados pre kínder, kínder, transición, primaria y básica secundaria, actualmente cuenta con 2 sedes (sede A y sede B) y una población académica compuesta por 23 docentes, 550 estudiantes en todos los niveles y 7 empleados, esto conlleva a que el manejo integral de la información del personal se haga cada vez más complejo.

En la actualidad los procesos administrativos que se realizan son los siguientes:

- 1. Control de asistencia a docentes y administrativos.**
- 2. Liquidación y pago de nómina.**
- 3. Control de asistencia a estudiantes.**
- 4. Registro de notas.**

Actualmente todas estas actividades se manejan y desarrollan de forma manual, por lo cual se evidencia constantemente saturación de trabajo en el campo administrativo esto se puede evidenciar: al momento de la liquidación y pago de nómina en cuanto a la recopilación de la información, el procesamiento de los datos, el riesgo en la precisión de los resultados, de igual manera, se presentan situaciones similares en la parte académica en lo que se refiere al registro, control y entrega de notas en los informes a padres de familia, el control de asistencia manual al momento de ingresar al colegio, el control de pagos de las mensualidades, entre otros.

Debido al crecimiento que tiene la institución y teniendo en cuenta la visión y proyección de esta, sus directivos han visto la necesidad de mejorar, garantizar y gestionar la información integral de la institución, con miras a optimizar los procesos, haciéndolos más amigables con el desarrollo del trabajo del recurso humano, asegurando la transparencia y precisión de los datos.

Teniendo en cuenta la perspectiva anterior, el problema a resolver se centra en la necesidad de dar solución en un contexto tecnológico, a partir de la ingeniería electrónica y la informática para organizar, gestionar y controlar la información pertinente a la actividad administrativa y académica de la institución Nuevo Colegio Lusadi Ltda. con esto, se puede plantear el problema en términos de:

¿CÓMO A TRAVÉS DEL USO ADECUADO DE HERRAMIENTAS TECNOLÓGICAS BASADAS EN ELECTRÓNICA E INFORMÁTICA SE PUEDE CONTRIBUIR AL DESARROLLO DE PROCESOS ADMINISTRATIVOS Y ACADÉMICOS DE LA INSTITUCIÓN NUEVO COLEGIO LUSADI CON MIRAS A LA OPTIMIZACIÓN DE LOS RECURSOS?

1.2. ANTECEDENTES

Con relación a los antecedentes de la Institución y la evolución de la estructura administrativa y académica que ha logrado desde el momento de su fundación, esta ha estado acorde con el crecimiento de los diferentes niveles educativos, permitiendo el desarrollo que actualmente tiene y hace parte del contexto que se incluirá dentro del presente trabajo.

Como parte de esta información, se tiene que la Institución Nuevo Colegio Lusadi Ltda. fue creada en el año 1992, en ese entonces dio sus primeros pasos con un grupo de 35 niños(as), en los grados pre-kínder, kínder y transición, en ese momento la institución se encontraba con una nómina reducida y con operaciones administrativas totalmente manuales, nominas hechas a mano, sistema de pagos con talonarios, al igual que el registro de notas y los boletines se realizaban manualmente.

Más adelante, en el año 1993, se inició el servicio educativo en la primaria media vocacional, con lo cual se vio incrementada la actividad administrativa y académica; ya que en ese momento se contaba con un total de 70 estudiantes, luego, entro a operar la formación básica secundaria en el año 1997.

Actualmente la institución cuenta con áreas como secretaría, coordinación académica, de disciplina, dirección general, entre otros, requiriendo así para el registro de pagos y registro académico al estudiantado la adquisición de un software llamado DATALEY adquirido en el año 2002 y complementándolo con plantillas en Excel para la liquidación de nómina y el control de asistencia con el que se encuentra trabajando actualmente.

Todo esto, será ampliado en el desarrollo del trabajo.

1.3. JUSTIFICACIÓN

Las razones que motivan la realización de este proyecto se resumen en:

Aportar en la solución de la problemática que tiene la institución en cuanto al control de asistencia a docentes y administrativos, liquidación y pago de nómina, control de asistencia a estudiantes y registro académico.

Contribuir con la implantación de procesos que aseguren un manejo óptimo, adecuado y seguro de la información teniendo en cuenta que actualmente la institución se encuentra muy expuesta a errores o inconsistencias en el manejo de la información.

Apoyar con herramientas dinámicas que den mayor eficiencia al desarrollo de procesos en la Institución, lo cual se verá reflejado en la mejora del rendimiento y disminución en el tiempo de realización de actividades.

Dar solución a las falencias y problemas que tiene la institución en su plataforma informática, por lo cual se creará un software a la medida de sus necesidades.

1.4. OBJETIVOS

1.4.1. OBJETIVO GENERAL

Diseñar y desarrollar un software para el manejo integral del personal (administrativo, de servicio y estudiantes), involucrando los siguientes procesos según correspondan: manejo de liquidación de nómina, control de asistencia, registro académico y control de pago mensual de pensión.

1.4.2 OBJETIVOS ESPECÍFICOS

- Realizar la carnetización adecuada para el control de acceso de la institución.
- Capacitar al personal en general del manejo del software, para el uso adecuado y familiarización con éste.
- Implementar la base de datos completa de todos los miembros de la institución para el desarrollo del software.
- Realizar un estudio completo de la manera en que la institución liquida el valor a recibir en pensión de cada estudiante, y la liquidación de nómina.
- Realizar un ingreso y control de notas del estudiante acorde con los estándares de calificación de la Institución.

1.5. FACTIBILIDAD

La factibilidad del proyecto está asegurada por cuanto se tienen los conocimientos necesarios en el contexto disciplinar de ingeniería electrónica adquiridos en el transcurso de la carrera, así mismo se cuenta con el apoyo de la institución Nuevo Colegio Lusadi, la cual se compromete con el suministro de los datos y demás información necesaria para la realización del trabajo, así mismo, ofrece los recursos de infraestructura y materia prima que se requieran para el logro de los objetivos.

1.6. COMPONENTE HUMANISTICO

La Institución Nuevo Colegio Lusadi Ltda, está ubicada en el barrio Lucero Medio de la localidad de Ciudad Bolívar en donde la mayoría de los estudiantes se encuentran en estratos 1 y 2, con este proyecto además de dar una solución eficiente para la institución se pretende brindar una herramienta tecnológica que ayude a sus estudiantes y los acerque y familiarice con las Tecnologías de la Información y comunicación de esta forma dichos estudiantes se educaran interactuando con la tecnología lo cual ayudara a su formación para el futuro, de igual forma los padres de familia también tendrán interacción a la tecnología y podrán estar más informados acerca del rendimiento académico de sus hijos y mejora la comunicación entre las Institución y estos, también ayuda a la calidad laboral de sus empleados ya que brinda las herramientas para agilizar diferentes procesos que correspondía hacerlos de forma manual generando saturación de trabajo en ciertas épocas del año, lo cual genera diferentes problemas en la salud de sus empleados debido al estrés y el agotamiento físico y mental.

Lo anterior se plantea siguiendo la misión institucional de la universidad la cual brinda una educación integral de profesionales con formación humanística y proyección social inspirada en el pensamiento Cristiano de Santo Tomas de Aquino.

CAPÍTULO 2

MARCO TEÓRICO

2.1. MARCO TEÓRICO

En el planteamiento del marco teórico se trabajará con base en los principios de programación teniendo como base los conocimientos en las áreas de digitales I, digitales II e interfaces al pc, vistas durante el periodo universitario y permitiendo el poder diseñar, implementar y desarrollar un software con los requerimientos deseados por la institución. Teniendo en cuenta que dichos conocimientos han brindado la lógica y las bases necesarias para poder adquirir un nuevo lenguaje de programación como lo es PHP, parte fundamental en el desarrollo del proyecto.

PHP es un lenguaje de programación que tiene una alta gama de compatibilidad con respecto al entorno de aplicaciones Web, cuenta con gran flexibilidad en fines a la conexión con las bases de datos, brindando un gran soporte para la conexión con el hosting siendo así uno de los lenguajes preferidos para desarrollar aplicaciones web.

Se hará uso de los conocimientos adquiridos en las diferentes materias acordes a los sistemas, para el manejo y diseño de base de datos, que se utilizarán en el desarrollo del control de acceso como parte de la confrontación del registro de asistencia del personal y los estudiantes en la plataforma MySQL.

Las bases de datos se realizarán en MySQL ya que es una plataforma lo suficientemente robusta para la aplicación, es fácilmente compatible con PHP, Javascript, HTML5 y CSS3, existe gran cantidad de documentación para su manejo e integración con PHP. MySQL cuenta con una versión gratuita la cual se acopla a las necesidades del software.

Para efectos de la gestión administrativa, se utilizarán los conceptos y la normatividad vigente relacionada con la parte laboral, de esta forma se hará la realización adecuada en la liquidación y pago de nómina de la Institución.

2.2. LENGUAJES DE PROGRAMACION

LENGUAJES MÁQUINA, ENSAMBLADORES Y DE ALTO NIVEL

Los programadores escriben instrucciones en diversos lenguajes de programación, algunos de los cuales comprende directamente la computadora, mientras que otros requieren pasos intermedios de traducción. En la actualidad se utilizan cientos de lenguajes de computación. Éstos se dividen en tres tipos generales:

1. Lenguajes máquina.
2. Lenguajes ensambladores.
3. Lenguajes de alto nivel.

Cualquier computadora puede entender de manera directa sólo su propio lenguaje máquina; que es su “lenguaje natural”, y como tal, está definido por el diseño del hardware de dicha computadora. Por lo general, los lenguajes máquina consisten en cadenas de números (que finalmente se reducen a 1s y 0s) que instruyen a las computadoras para realizar sus operaciones más elementales, una a la vez. Los lenguajes máquina son dependientes de la máquina (es decir, un lenguaje máquina en particular puede usarse solamente en un tipo de computadora). Dichos lenguajes son difíciles de comprender para los humanos, el siguiente ejemplo muestra uno de los primeros programas en lenguaje máquina, el cual suma el pago de las horas extras al sueldo base y almacena el resultado en el sueldo bruto:

```
+1300042774
+1400593419
+1200274027
```

La programación en lenguaje máquina era demasiado lenta y tediosa para la mayoría de los programadores. En vez de utilizar las cadenas de números que las computadoras podían entender directamente, los programadores empezaron a utilizar abreviaturas del inglés para representar las operaciones elementales. Estas abreviaturas formaron la base de los lenguajes ensambladores. Los programas traductores conocidos como ensambladores se desarrollaron para convertir los primeros programas en lenguaje ensamblador a lenguaje máquina, a la velocidad de la computadora. A continuación se muestra un ejemplo de un programa en lenguaje ensamblador, que también suma el pago de las horas extras al sueldo base y almacena el resultado en el sueldo bruto¹:

```
load sueldobase
add sueldoextra
```

¹ (como programar en java séptima edición- P.J Deitel pag 6-7)

store sueldoBruto

Aunque este código es más claro para los humanos, las computadoras no lo pueden entender sino hasta que se traduce en lenguaje máquina.

El uso de las computadoras se incrementó rápidamente con la llegada de los lenguajes ensambladores, pero los programadores aún requerían de muchas instrucciones para llevar a cabo incluso hasta las tareas más simples. Para agilizar el proceso de programación se desarrollaron los lenguajes de alto nivel, en donde podían escribirse instrucciones individuales para realizar tareas importantes. Los programas traductores, denominados compiladores, convierten, a lenguaje máquina, los programas que están en lenguaje de alto nivel. Estos últimos permiten a los programadores escribir instrucciones que son muy similares al inglés común, y contienen la notación matemática común. Un programa de nómina escrito en un lenguaje de alto nivel podría contener una instrucción como la siguiente:

sueldoBruto = sueldoBase + sueldoExtra

Obviamente, desde el punto de vista del programador, los lenguajes de alto nivel son mucho más recomendables que los lenguajes máquina o ensamblador. C, C++ y los lenguajes .NET de Microsoft (por ejemplo, Visual Basic .NET, Visual C++ .NET y C#) son algunos de los lenguajes de programación de alto nivel que más se utilizan; sin embargo, Java es *el* más utilizado.

El proceso de compilación de un programa escrito en lenguaje de alto nivel a un lenguaje máquina puede tardar un tiempo considerable en la computadora. Los programas intérpretes se desarrollaron para ejecutar programas en lenguaje de alto nivel directamente, aunque con más lentitud. Los intérpretes son populares en los entornos de desarrollo de programas, en los cuales se agregan nuevas características y se corrigen los errores. Una vez que se desarrolla un programa por completo, se puede producir una versión compilada para ejecutarse con la mayor eficiencia.

Actualmente se sabe que existen dos formas de traducir un programa en lenguaje de alto nivel a un formato que la computadora pueda entender: compilación e interpretación.²

² (como programar en java séptima edición- P.J Deitel pag 6-7)

FORTTRAN, COBOL, PASCAL Y ADA

Se han desarrollado cientos de lenguajes de alto nivel, pero sólo unos cuantos han logrado una amplia aceptación. Fortran (FORmula TRANslator, Traductor de fórmulas) fue desarrollado por IBM Corporation a mediados de la década de los cincuenta para utilizarse en aplicaciones científicas y de ingeniería que requerían cálculos matemáticos complejos. En la actualidad, Fortran se utiliza ampliamente en aplicaciones de ingeniería.

COBOL (Common Business Oriented Language, Lenguaje común orientado a negocios) fue desarrollado a finales de la década de los cincuenta por fabricantes de computadoras, el gobierno estadounidense y usuarios de computadoras de la industria. COBOL se utiliza en aplicaciones comerciales que requieren de una manipulación precisa y eficiente de grandes volúmenes de datos. Gran parte del software de negocios aún se programa en COBOL.

Durante la década de los sesenta, muchos de los grandes esfuerzos para el desarrollo de software encontraron severas dificultades. Los itinerarios de software generalmente se retrasaban, los costos rebasaban en gran medida a los presupuestos y los productos terminados no eran confiables. La gente comenzó a darse cuenta de que el desarrollo de software era una actividad mucho más compleja de lo que habían imaginado. Las actividades de investigación en la década de los sesenta dieron como resultado la evolución de la programación estructurada (un método disciplinado para escribir programas que fueran más claros, fáciles de probar y depurar, y más fáciles de modificar que los programas extensos producidos con técnicas anteriores).

Uno de los resultados más tangibles de esta investigación fue el desarrollo del lenguaje de programación Pascal por el profesor Niklaus Wirth, en 1971. Pascal, cuyo nombre se debe al matemático y filósofo Blaise Pascal del siglo diecisiete, se diseñó para la enseñanza de la programación estructurada en ambientes académicos, y de inmediato se convirtió en el lenguaje de programación preferido en la mayoría de las universidades. Pascal carece de muchas de las características necesarias para poder utilizarse en aplicaciones comerciales, industriales y gubernamentales, por lo que no ha sido muy aceptado en estos entornos³.

El lenguaje de programación Ada se desarrolló bajo el patrocinio del Departamento de Defensa de los Estados Unidos (DOD) durante la década de los setenta y los primeros años de la década de los ochenta. Cientos de lenguajes independientes se utilizaron para producir los sistemas de software masivos de comando y control del departamento de defensa. Éste quería un solo lenguaje que pudiera satisfacer la mayoría de sus necesidades. El nombre del lenguaje es en honor de Lady Ada Lovelace, hija del poeta Lord Byron. A Lady Lovelace se le atribuye

³ (como programar en java séptima edición- P.J Deitel pag 9-10)

el haber escrito el primer programa para computadoras en el mundo, a principios de la década de 1800 (para la Máquina Analítica, un dispositivo de cómputo mecánico diseñado por Charles Babbage). Una de las características importantes de Ada se conoce como multitarea, la cual permite a los programadores especificar que muchas actividades ocurran en paralelo. Java, a través de una técnica que se conoce como *subprocesamiento múltiple*, también permite a los programadores escribir programas con actividades paralelas.⁴

BASIC, VISUAL BASIC, VISUAL C++, C# Y .NET

El lenguaje de programación BASIC (Beginner's All-Purpose Symbolic Instruction Code, Código de instrucciones simbólicas de uso general para principiantes) fue desarrollado a mediados de la década de los sesenta en el Dartmouth College, como un medio para escribir programas simples. El propósito principal de BASIC era que los principiantes se familiarizaran con las técnicas de programación.

El lenguaje Visual Basic de Microsoft se introdujo a principios de la década de los noventa para simplificar el desarrollo de aplicaciones para Microsoft Windows, y es uno de los lenguajes de programación más populares en el mundo.

Las herramientas de desarrollo más recientes de Microsoft forman parte de su estrategia a nivel corporativo para integrar Internet y Web en las aplicaciones de computadora. Esta estrategia se implementa en la plataforma .NET de Microsoft, la cual proporciona a los desarrolladores las herramientas que necesitan para crear y ejecutar aplicaciones de computadora que puedan ejecutarse en computadoras distribuidas a través de Internet. Los tres principales lenguajes de programación de Microsoft son Visual Basic .NET (basado en el lenguaje BASIC original), Visual C++ .NET (basado en C++) y C# (basado en C++ y Java, y desarrollado expresamente para la plataforma .NET). Los desarrolladores que utilizan .NET pueden escribir componentes de software en el lenguaje con el que estén más familiarizados y formar aplicaciones al combinar esos componentes con los ya escritos en cualquier lenguaje .NET.⁵

⁴ (como programar en java séptima edición- P.J Deitel pag 9-10)

⁵ (como programar en java séptima edición- P.J Deitel pag. 10)

HISTORIA DE C Y C++

Java evolucionó de C++, el cual evolucionó de C, que a su vez evolucionó de BCPL y B. En 1967, Martin Richards desarrolló BCPL como un lenguaje para escribir software para sistemas operativos y compiladores. Ken Thompson modeló muchas características en su lenguaje B a partir del trabajo de sus contrapartes en BCPL, y utilizó a B para crear las primeras versiones del sistema operativo UNIX, en los laboratorios Bell en 1970.

El lenguaje C evolucionó a partir de B, gracias al trabajo de Dennis Ritchie en los laboratorios Bell, y se implementó originalmente en 1972. Inicialmente, se hizo muy popular como lenguaje de desarrollo para el sistema operativo UNIX. En la actualidad, la mayoría del código para los sistemas operativos de propósito general (por ejemplo, los que se encuentran en las computadoras portátiles, de escritorio, estaciones de trabajo y pequeños servidores) se escribe en C o C++.

A principios de la década de los ochenta, Bjarne Stroustrup desarrolló una extensión de C en los laboratorios Bell: C++. Este lenguaje proporciona un conjunto de características que “pulen” al lenguaje C pero, lo más importante es que proporciona la capacidad de una *programación orientada a objetos*. C++ es un lenguaje híbrido: es posible programar en un estilo parecido a C, en un estilo orientado a objetos, o en ambos.

Una revolución se está gestando en la comunidad del software. Escribir software de manera rápida, correcta y económica es aún una meta difícil de alcanzar, en una época en que la demanda de nuevo y más poderoso software se encuentra a la alza. Los *objetos*, o dicho en forma más precisa, las clases a partir de las cuales se crean los objetos, son en esencia componentes reutilizables de software. Hay objetos de: fecha, hora, audio, automóvil, personas, etcétera; de hecho, casi cualquier sustantivo puede representarse como objeto de software en términos de atributos (como el nombre, color y tamaño) y comportamientos (como calcular, desplazarse y comunicarse). Los desarrolladores de software están descubriendo que utilizar una metodología de diseño e implementación modular y orientada a objetos puede hacer más productivos a los grupos de desarrollo de software, que mediante las populares técnicas de programación anteriores, como la programación estructurada. Los programas orientados a objetos son, a menudo, más fáciles de entender, corregir y modificar. Java es el lenguaje de programación orientada a objetos que más se utiliza en el mundo.⁶

⁶ (como programar en java séptima edición- P.J Deitel pag 7-8)

HISTORIA DE JAVA

La contribución más importante a la fecha, por parte de la revolución del microprocesador, es que hizo posible el desarrollo de las computadoras personales, que ahora suman miles de millones a nivel mundial. Las computadoras personales han tenido un profundo impacto en la vida de las personas, y en la manera en que las empresas realizan y administran su negocio.

Los microprocesadores están teniendo un profundo impacto en los dispositivos electrónicos inteligentes para uso doméstico. Al reconocer esto, Sun Microsystems patrocinó en 1991 un proyecto interno de investigación denominado Green, el cual desembocó en el desarrollo de un lenguaje basado en C++ al que su creador, James Gosling, llamó Oak debido a un roble que tenía a la vista desde su ventana en las oficinas de Sun. Posteriormente se descubrió que ya existía un lenguaje de computadora con el mismo nombre. Cuando un grupo de gente de Sun visitó una cafetería local, sugirieron el nombre Java (una variedad de café) y así se quedó.

Pero el proyecto Green tuvo algunas dificultades. El mercado para los dispositivos electrónicos inteligentes de uso doméstico no se desarrollaba tan rápido a principios de los noventa como Sun había anticipado. El proyecto corría el riesgo de cancelarse. Pero para su buena fortuna, la popularidad de World Wide Web explotó en 1993 y la gente de Sun se dio cuenta inmediatamente del potencial de Java para agregar **contenido dinámico**, como interactividad y animaciones, a las páginas Web. Esto trajo nueva vida al proyecto.

Sun anunció formalmente a Java en una importante conferencia que tuvo lugar en mayo de 1995. Java generó la atención de la comunidad de negocios debido al fenomenal interés en World Wide Web. En la actualidad, Java se utiliza para desarrollar aplicaciones empresariales a gran escala, para mejorar la funcionalidad de los servidores Web (las computadoras que proporcionan el contenido que vemos en nuestros exploradores Web), para proporcionar aplicaciones para los dispositivos domésticos (como teléfonos celulares, radiolocalizadores y asistentes digitales personales) y para muchos otros propósitos.⁷

⁷ (como programar en java séptima edición- P.J Deitel pag 8)

LENGUAJE JAVA

En 1991, la empresa Sun Microsystems crea el lenguaje Oak (de la mano del llamado proyecto Green). Mediante este lenguaje se pretendía crear un sistema de televisión interactiva. Este lenguaje sólo se llegó a utilizar de forma interna. Su propósito era crear un lenguaje independiente de la plataforma y para uso en dispositivos electrónicos. Se intentaba con este lenguaje solucionar el problema fundamental del C++; que consiste en que al compilar se produce un fichero ejecutable cuyo código sólo vale para la plataforma en la que se realizó la compilación.

Sun deseaba un lenguaje para programar pequeños dispositivos electrónicos. La dificultad de estos dispositivos es que cambian continuamente y para que un programa funcione en el siguiente dispositivo aparecido, hay que rescribir el código. Por eso Sun quería crear un lenguaje independiente del dispositivo. En 1995 pasa a llamarse Java y se da a conocer al público. Adquiere notoriedad rápidamente. Java pasa a ser un lenguaje totalmente independiente de la plataforma y a la vez potente y orientado a objetos. Esa filosofía y su facilidad para crear aplicaciones para redes TCP/IP ha hecho que sea uno de los lenguajes más utilizados en la actualidad.

La versión actual de Java es el llamado Java 2. Sus ventajas sobre C++ son:

- Su sintaxis es similar a C y C++.
- No hay punteros (lo que le hace más seguro).€
- Totalmente orientado a objetos.
- Muy preparado para aplicaciones TCP/IP.
- Implementa excepciones de forma nativa. €
- Es interpretado (lo que acelera su ejecución remota, aunque provoca que las aplicaciones Java se ejecuten más lentamente que las C++ en un ordenador local). €
- Permite multihilos.€
- Admite firmas digitales. €
- Tipos de datos y control de sintaxis más rigurosa. €
- Es independiente de la plataforma.

La última ventaja (quizá la más importante) se consigue ya que el código Java no se compila, sino que se pre compila, de tal forma que se crea un código intermedio que no es ejecutable. Para ejecutarle hace falta pasarle por un intérprete que va ejecutando cada línea. Ese intérprete suele ser la máquina virtual de Java.⁸

⁸ (<http://jorgesanchez.net/programacion/manuales/Java.pdf>)

HTML5

HTML5 no es una nueva versión del antiguo lenguaje de etiquetas, ni siquiera una mejora de esta ya antigua tecnología, sino un nuevo concepto para la construcción de sitios web y aplicaciones en una era que combina dispositivos móviles, computación en la nube y trabajos en red. Todo comenzó mucho tiempo atrás con una simple versión de HTML propuesta para crear la estructura básica de páginas web, organizar su contenido y compartir información.

El lenguaje y la web misma nacieron principalmente con la intención de comunicar información por medio de texto. El limitado objetivo de HTML motivó a varias compañías a desarrollar nuevos lenguajes y programas para agregar características a la web nunca antes implementadas. Estos desarrollos iniciales crecieron hasta convertirse en populares y poderosos accesorios. Simples juegos y bromas animadas pronto se transformaron en sofisticadas aplicaciones, ofreciendo nuevas experiencias que cambiaron el concepto de la web para siempre.

De las opciones propuestas, Java y Flash fueron las más exitosas; ambas fueron masivamente adoptadas y ampliamente consideradas como el futuro de Internet. Sin embargo, tan pronto como el número de usuarios se incrementó e Internet pasó de ser una forma de conectar amantes de los ordenadores a un campo estratégico para los negocios y la interacción social, limitaciones presentes en estas dos tecnologías probaron ser una sentencia de muerte.

El mayor inconveniente de Java y Flash puede describirse como una falta de integración. Ambos fueron concebidos desde el principio como complementos (plug-ins), algo que se inserta dentro de una estructura pero que comparte con la misma solo espacio en la pantalla. No existía comunicación e integración alguna entre aplicaciones y documentos. La falta de integración resultó ser crítica y preparó el camino para la evolución de un lenguaje que comparte espacio en el documento con HTML y no está afectado por las limitaciones de los plugins. Javascript, un lenguaje interpretado incluido en navegadores, claramente era la manera de mejorar la experiencia de los usuarios y proveer funcionalidad para la web. Sin embargo, después de algunos años de intentos fallidos para promoverlo y algunos malos usos, el mercado nunca lo adoptó plenamente y pronto su popularidad declinó.

Los detractores tenían buenas razones para oponerse a su adopción. En ese momento, Javascript no era capaz de reemplazar la funcionalidad de Flash o Java. A pesar de ser evidente que ambos limitaban el alcance de las aplicaciones y aislaban el contenido web, populares funciones como la reproducción de video se estaban convirtiendo en una parte esencial de la web y solo eran efectivamente ofrecidas a través de estas tecnologías. A pesar del suceso inicial, el uso de Java comenzó a declinar⁹.

La naturaleza compleja del lenguaje, su evolución lenta y la falta de integración disminuyeron su importancia hasta el punto en el que hoy día no es más usado en aplicaciones web de importancia. Sin Java, el mercado volcó su atención a Flash. Pero el hecho de que Flash

⁹ (El gran libro de HTML5, CSS3 y Javascript - Juan Diego Gauchat editorial marcombo)

comparte las mismas características básicas que su competidor en la web lo hace también susceptible de correr el mismo destino. Mientras esta competencia silenciosa se llevaba a cabo, el software para acceder a la web continuaba evolucionando. Junto con nuevas funciones y técnicas rápidas de acceso a la red, los navegadores también mejoraron gradualmente sus intérpretes Javascript. Más potencia trajo más oportunidades y este lenguaje estaba listo para aprovecharlas. En cierto punto durante este proceso, se hizo evidente para algunos desarrolladores que ni Java o Flash podrían proveer las herramientas que ellos necesitaban para crear las aplicaciones demandadas por un número creciente de usuarios. Estos desarrolladores, impulsados por las mejoras otorgadas por los navegadores, comenzaron a aplicar Javascript en sus aplicaciones de un modo nunca visto.

La innovación y los increíbles resultados obtenidos llamaron la atención de más programadores. Pronto lo que fue llamado la “Web 2.0” nació y la percepción de Javascript en la comunidad de programadores cambió radicalmente. Javascript era claramente el lenguaje que permitía a los desarrolladores innovar y hacer cosas que nadie había podido hacer antes en la web.

En los últimos años, programadores y diseñadores web alrededor del mundo surgieron con los más increíbles trucos para superar las limitaciones de esta tecnología y sus iniciales deficiencias en portabilidad. Gracias a estas nuevas implementaciones, Javascript, HTML y CSS se convirtieron pronto en la más perfecta combinación para la necesaria evolución de la web. HTML5 es, de hecho, una mejora de esta combinación, el pegamento que une todo.

HTML5 propone estándares para cada aspecto de la web y también un propósito claro para cada una de las tecnologías involucradas. Apartir de ahora, HTML provee los elementos estructurales, CSS se encuentra concentrado en cómo volver esa estructura utilizable y atractiva a la vista, y Javascript tiene todo el poder necesario para proveer dinamismo y construir aplicaciones web completamente funcionales. Las barreras entre sitios webs y aplicaciones finalmente han desaparecido.

Las tecnologías requeridas para el proceso de integración están listas. El futuro de la web es prometedor y la evolución y combinación de estas tres tecnologías (HTML, CSS y Javascript) en una poderosa especificación está volviendo a Internet la plataforma líder de desarrollo. HTML5 indica claramente el camino¹⁰.

HTML5 provee básicamente tres características: estructura, estilo y funcionalidad. Nunca fue declarado oficialmente pero, incluso cuando algunas APIs (Interface de Programación de Aplicaciones) y la especificación de CSS3 por completo no son parte del mismo, HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. Estas tecnologías son altamente dependientes y actúan como una sola unidad organizada bajo la especificación de HTML5.

¹⁰ (El gran libro de HTML5, CSS3 y Javascript - Juan Diego Gauchat editorial marcombo)

HTML está a cargo de la estructura, CSS presenta esa estructura y su contenido en la pantalla y Javascript hace el resto que (como veremos más adelante) es extremadamente significativo. Más allá de esta integración, la estructura sigue siendo parte esencial de un documento. La misma provee los elementos necesarios para ubicar contenido estático o dinámico, y es también una plataforma básica para aplicaciones.

Con la variedad de dispositivos para acceder a Internet y la diversidad de interfaces disponibles para interactuar con la web, un aspecto básico como la estructura se vuelve parte vital del documento. Ahora la estructura debe proveer forma, organización y flexibilidad, y debe ser tan fuerte como las bases de un edificio.¹¹

CSS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados "*documentos semánticos*"). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para *marcar* los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.¹²

CSS3

Cascading Style Sheets, versión número 3, viene ligado a HTML5. CSS es lo que hace bonito a un sitio web. Se trata del lenguaje que utilizamos para dar colores, tipos y tamaños de letra, y, valga la redundancia, estilo a nuestra página. Al escribir un sitio web con HTML, debemos enlazarlo a una hoja de estilos (archivo con extensión .css) que le dará características.

¹¹ (El gran libro html5-css3 y javascript Juan Diego Gauchat Editorial marcombo)

¹² (http://librosweb.es/libro/css/capitulo_1.html)

Con CSS3 simplificamos tareas antes difíciles de efectuar, como aplicar bordes redondeados, sombras y animaciones (!) a cualquier elemento del archivo HTML, de manera sencilla, rápida y bastante eficaz, ya que funciona en casi todos los navegadores modernos.¹³

JAVASCRIPT

Javascript es un lenguaje interpretado usado para múltiples propósitos pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos Javascript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código Javascript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejorada capacidad permitió superar viejas limitaciones de rendimiento y confirmar el lenguaje Javascript como la mejor opción para la web.

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, Javascript fue expandido en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (APIs) fueron incorporadas por defecto en cada navegador para asistir al lenguaje en funciones elementales. Estas nuevas APIs (como Web Storage, Canvas, y otras) son interfaces para librerías incluidas en navegadores. La idea es hacer disponible poderosas funciones a través de técnicas de programación sencillas y estándares, expandiendo el alcance del lenguaje y facilitando la creación de programas útiles para la web.¹⁴

PHP

El lenguaje PHP es un lenguaje de programación de estilo clásico, es decir que es un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. No es un lenguaje de etiquetas como podría ser HTML, XML o WML. Está más cercano a JavaScript o a C, para aquellos que conocen estos lenguajes.

Pero a diferencia de Java o JavaScript que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor como por ejemplo podría ser una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, sin embargo para que las páginas PHP funcionen, el servidor donde están alojadas debe soportar PHP.¹⁵

¹³ (<http://tc2proyectodegradoingdesistemas.blogspot.com.co/2012/05/introduccion.html>)

¹⁴ (El gran libro de HTML5, CSS3 y Javascript - Juan Diego Gauchat editorial marcombo)

¹⁵ (Manual de programación php- Marcelo Maraboli rosselott)

APACHE

El servidor Apache HTTP , también llamado Apache, es un servidor web HTTP de código abierto para la creación de páginas y servicios web. Es un servidor multiplataforma, gratuito, muy robusto y que destaca por su seguridad y rendimiento.

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Para entender mejor lo que es Apache, primeramente definiremos lo que es un servidor web. La definición más sencilla de servidor web, que es un programa especialmente diseñado para transferir datos de hipertexto, es decir, páginas web con todos sus elementos (textos, widgets, banners, etc). Estos servidores web utilizan el protocolo http.

Los servidores web están alojados en un ordenador que cuenta con conexión a Internet. El web server, se encuentra a la espera de que algún navegador le haga alguna petición, como por ejemplo, acceder a una página web y responde a la petición, enviando código HTML mediante una transferencia de datos en red.

La historia de Apache se remonta a febrero de 1995, donde empieza el proyecto del grupo Apache, el cual está basado en el servidor Apache httpd de la aplicación original de NCSA. El desarrollo de esta aplicación original se estancó por algún tiempo tras la marcha de Rob McCool.

Fueron Brian Behlendorf y Cliff Skolnick quienes a través de una lista de correo coordinaron el trabajo y lograron establecer un espacio compartido de libre acceso para los desarrolladores.

En 1999, se formó la Fundación de Software Apache (Apache Software Foundation) para obtener apoyo financiero, organizativo y legal para el servidor¹⁶.

Ventajas

- Instalación/Configuración. Software de código abierto.
- Coste. El servidor web Apache es completamente gratuito.
- Funcional y Soporte. Alta aceptación en la red y muy popular, esto hace que muchos programadores de todo el mundo contribuyen constantemente con mejoras, que están disponibles para cualquier persona que use el servidor web y que Apache se actualice constantemente.
- Multi-plataforma. Se puede instalar en muchos sistemas operativos, es compatible con Windows, Linux y MacOS.
- Rendimiento. Capacidad de manejar más de un millón de visitas/día.
- Soporte de seguridad SSL y TLS.¹⁷

¹⁶ (<http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>)

MySQL

MySQL es un *sistema de administración de bases de datos relacional* (RDBMS). Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios conocidos, como Oracle, SQL Server y DB2. MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger y hacer volcados de datos. Puede desarrollar sus propias aplicaciones de base de datos en la mayor parte de los lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, incluyendo algunos de los que probablemente no ha oído nunca hablar. MySQL utiliza el lenguaje de consulta estructurado (SQL). Se trata del lenguaje utilizado por todas las bases de relacionales, que presentaremos en una sección posterior. Este lenguaje permite crear bases de datos, así como agregar, manipular y recuperar datos en función de criterios específicos.

Pero nos estamos adelantando. En este capítulo, se analizan brevemente los conceptos relativos a las bases de datos relacionales. Aprenderemos que se entiende exactamente por una base de datos relacional y como funciona, además de comentar terminología clave. Armados con esta información, podremos crear una sencilla base de datos y trabajar con sus datos.¹⁸

FRAMEWORK LARAVEL

Laravel, propone una forma de desarrollar aplicaciones web de un modo mucho más ágil. Por ejemplo, en Laravel opcionalmente podemos usar el patrón de diseño MVC (Modelo-Vista-Controlador) tradicional, donde al igual que otros frameworks PHP, el controlador es programado como una clase. Por lo tanto, un Controlador es una clase PHP que dispone de métodos públicos que son el punto de entrada final de una petición HTTP (Request PHP) a nuestra aplicación. Pero, Laravel propone además una forma distinta y más directa de responder a la solicitud HTTP.

Laravel 'entrega la opción' de seguir usando la metodología tradicional MVC. Sin embargo, el framework propone una vía más rápida en PHP, la cual consiste en programar la interacción HTTP directamente como una función anónima asociada a una Ruta. Esto tiene la ventaja

¹⁷ (<http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>)

¹⁸ (La Biblia de MySQL Ian Giffan editorial Anaya)

de reducir la cantidad de código, especialmente cuando sólo necesitamos incluir una funcionalidad.¹⁹

2.3. BASES DE DATOS

Un sistema de bases de datos es básicamente un *sistema computarizado para llevar registros*. Es posible considerar a la propia base de datos como una especie de armario electrónico para archivar; es decir, es un depósito o contenedor de una colección de archivos de datos computarizados.

Los usuarios del sistema pueden realizar una variedad de operaciones sobre dichos archivos, por ejemplo:

- Agregar nuevos archivos vacíos a la base de datos;
- Insertar datos dentro de los archivos existentes;
- Recuperar datos de los archivos existentes;
- Modificar datos en archivos existentes;
- Eliminar datos de los archivos existentes;
- Eliminar archivos existentes de la base de datos.²⁰

2.4. ¿QUÉ SON LOS SERVIDORES?

Un servidor puede encontrarse en un típico local que ofrece el uso de computadoras a sus clientes. La máquina que tiene el cajero da un servicio; es un servidor, encargado de habilitar o deshabilitar una PC para que pueda ser usada para navegar o jugar. Si deja de funcionar, el negocio no factura, y ninguna de las máquinas cliente podría ser utilizada.

Los servidores son equipos informáticos que brindan un servicio en la red. Dan información a otros servidores y a los usuarios. Son equipos de mayores prestaciones y dimensiones que una PC de escritorio.

Una computadora común tiene un solo procesador, a veces de varios núcleos, pero uno solo. Incluye un disco rígido para el almacenamiento de datos con una capacidad de 250 GB a 300 GB, en tanto que la memoria RAM suele ser de 2 a 16 GB. Un servidor, en cambio, suele ser más potente. Puede tener varios procesadores con varios núcleos cada uno; incluye grandes cantidades de memoria RAM, entre 16 GB a 1 TB, o más; mientras que el espacio de almacenamiento ya no se limita a un disco duro, sino que puede haber varios de ellos, con capacidad del orden del TB. Debido a sus capacidades, un servidor puede dar un solo servicio o más de uno.

¹⁹ (<http://desarrollandowebsdinamicas.blogspot.com.co/2013/03/que-es-laravel.html>)

²⁰ (https://kesquivel.files.wordpress.com/2011/03/clase_no1_quc3a9-es-un-sistema-de-base-de-datos.pdf)

EJEMPLOS EN EQUIPOS DELL	SERVIDOR POWEREDGE R910	PC DE ESCRITORIO OPTIPLEX 960
Microprocesador	Eight-Core Intel Xeon 7500 and 6500 Series, hasta 24 MB de caché L3	Intel Core2 Quad Processor, hasta 12 MB de caché L2
Disco duro	Hasta 9 TB SSD y SAS	Hasta 320 GB SATA II
Memoria RAM	Hasta 1 TB, ECC DDR3, 1066 MHz	Hasta 16 GB, DDR2 SDRAM, 800 MHz
Placa gráfica	Matrox® G200eW w/ 8 MB	512 MB NVIDIA NVS 420 Quad Monitor

Tabla 2.1. Comparación entre las características de una PC de escritorio y un servidor.

TIPOS DE SERVIDORES

Existen distintos tipos de servidores, y pueden ser virtuales o físicos. Podemos clasificarlos según sus capacidades, fabricantes y servicios prestados. A continuación, describiremos esta última categorización²¹:

- **Servidores de impresión:** tienen conectadas varias impresoras de red y administran las colas de impresión según la petición de sus clientes.
- **Servidores web:** este tipo de servidores se encargan de almacenar sitios en la red interna (intranet). Pueden publicar cualquier aplicación web, brindarle la seguridad correspondiente y administrarla por completo.
- **Servidores de base de datos:** lo más importante de estos servidores es la posibilidad de manejar grandes cantidades de datos y generar información. Para contener todo ese material generalmente se conectan a un storage.
- **Servidores de correo electrónico:** son capaces de administrar todos los correos de la empresa en un solo lugar. También trabajan con un storage, debido a la gran cantidad de datos que manejan. Allí se almacenan los correos, y se los re direcciona a clientes y servidores de seguridad, analizadores y replicadores. Algunos también brindan opciones de seguridad, como antispam, lista blanca, lista negra y antivirus.
- **Servidores de directorio:** se ocupan de almacenar los datos de todos los usuarios de la red, propiedades y características que los identifican. Veremos más sobre este tema en el Capítulo 2, en el apartado Directorio de dominio.
- **Servidores de comunicaciones:** brindan servicios de chat, telefonía IP, teleconferencia, video, etc. También son capaces de entregar servicios de pre atendedor si se los conecta a una consola telefónica.

²¹ (Administrador de servidores, por Marchionni, Enzo Augusto , pag 23-30)

- **Servidores de archivos:** nos permiten compartir el material y guardarlo de manera segura, y ofrecen una mayor capacidad de almacenamiento que los equipos de escritorio. Pueden tener conectados varios storage de distintas capacidades.
- **Servidores de seguridad:** se dedican a escanear la red en busca de virus, máquinas desactualizadas por falta de parches del sistema operativo, equipos con determinado software instalado, y muchas otras acciones más.
- **Servidores proxy:** brindan acceso a Internet. En ellos generalmente residen firewalls a los que se les configuran reglas para permitir la navegación por ciertas páginas y bloquear otras. Pueden redirigir la navegación y mostrarnos algún cartel de advertencia o violación de la política empresarial.
- **Servidores de servidores virtuales:** un solo servidor físico puede contener varios servidores virtuales, pero el usuario final no distinguirá las diferencias. Solo desde su administración podremos explotar todas sus características. Para obtener más información sobre servidores virtuales.
- **Servidores particulares:** se instalan para cada aplicación que utilicemos en la red. Por ejemplo, servidores de workflows, de CRM, de RR.HH., de contaduría, etc.

Los servidores, por sus diferencias físicas, de tamaño y de diseño, también se dividen en rackeables, tipo tower y blades. Los rackeables son aquellos que podemos colocar dentro de un armario con correderas (rack); suelen ser delgados como una laptop de grandes dimensiones. Los servidores tower son los más típicos, parecidos a una PC físicamente, pero más potentes. Por último, los blades son equipos grandes que permiten cambiar o agregar hardware de forma caliente, esto es, mientras el servidor está activo.²²

SERVICIOS NECESARIOS EN UNA EMPRESA

Al margen de las distintas categorías de servidores que puede tener una empresa, debemos considerar que hay ciertos servicios esenciales que ellos brindan y que son críticos en nuestro trabajo diario. Por ejemplo, un servicio indispensable es el correo electrónico, que debe ser bien administrado y tratado con sumo cuidado. Sin dudas, también es infaltable el servidor de archivos, conocido como file server, cuya función es almacenar gran cantidad de información para que los empleados puedan acceder a ella desde cualquier lugar de la red. Brindan un servicio de resguardo, y deben asegurar la integridad y solidez de los datos. En estos servidores podemos administrar los permisos a las carpetas y las cuotas para cada una de ellas.

²² (administración de servidores- Enzo Augusto Marchionni- Manuales USERS)

Según el tamaño de la empresa, existen otros servicios indispensables. Si es de mediana a grande, debemos pensar en un servidor de backup con su correspondiente red de backup para no saturar el tráfico de la red actual. También debemos tener en cuenta los servicios de un departamento de bases de datos (DBAs), de helpdesk, de seguridad y de procesos.

RESPONSABILIDADES PARA ASUMIR

En todos los trabajos existen responsabilidades; en este, básicamente, es la continuidad del negocio. Esto quiere decir que la empresa debe seguir manteniendo sus servicios y servidores funcionando continuamente. Es preciso tratar de obtener un servicio de 24 horas durante los 365 días del año.

RIESGOS EN LA ADMINISTRACIÓN

La administración de servidores es todo un desafío diario, porque nadie sabe resolver todos los problemas que pueden llegar a presentarse. Principalmente, hay que tener sentido común, ser serenos y conscientes. Tratar de automatizar al máximo las tareas frecuentes es una buena idea para estar preparados ante problemas más serios y ofrecer una rápida respuesta. Este es un trabajo difícil, en el que, en ciertas situaciones, las ventas de la empresa, la conformidad de algún cliente y/o la facturación del mes quedan supeditadas al buen funcionamiento del sistema administrado. El negocio debe continuar para seguir facturando, y depende de nosotros que esto suceda.

Antes de poner un sistema en producción, debemos crear un escenario parecido al real para probar el que vamos a instalar. Esto se denomina **laboratorio**. Por ejemplo,

Si queremos instalar una función nueva para el correo en nuestra red, sabemos que necesitamos configurar un servidor Exchange y uno de Active Directory. Entonces, en el laboratorio tendremos que instalar dos servidores que ejerzan esas funciones y una PC para hacer las pruebas del lado del cliente, es decir, del usuario final. Si tratamos cada nueva instalación de esta manera, tendremos la mínima pérdida posible de la continuidad del negocio a la hora de implementar un ambiente productivo. Debemos dedicar horas a estas situaciones de **testeo** previo, para asegurarnos de que todo salga bien. Por ejemplo, si programamos los reinicios de todas las estaciones de trabajo (**workstations**) de la empresa y no tenemos en cuenta que hay personal trabajando fuera de horario, la continuidad del negocio caerá abruptamente²³.

²³ (Administrador de servidores, por Marchionni, Enzo Augusto , pag 23-30)

AMBIENTES DE TRABAJO

Para cumplir con las exigencias y minimizar los riesgos, debemos mantener un buen ambiente de trabajo, y hay normas internacionales que lo regulan. Dentro de las Normas ISO 9001:2000, por ejemplo, se menciona el ambiente de trabajo como un requisito: **la Organización debe determinar y gestionar el ambiente de trabajo necesario para lograr la conformidad con los requisitos del producto.**

La Norma ISO 9004:2000 se refiere un poco más a qué es el ambiente de trabajo. Cubre aspectos como las metodologías de trabajo, las reglas de seguridad, la interacción social, las instalaciones y la higiene. Los departamentos de recursos humanos deben asegurarse de asignar personas idóneas para realizar estas tareas, de modo de alcanzar un buen estado psicológico del empleado y generar un compromiso constante. También las **normas de seguridad** juegan un papel muy importante en el ambiente del trabajo. Existen normas que prohíben la navegación por las redes sociales o bloquean las comunicaciones IM. Podemos encontrar otra postura de administración, en la cual los directivos piensan y admiten que el empleado pasa más tiempo en la empresa que en su casa, por lo que debería sentirse como en ella. Debemos tener en cuenta que quienes están en esta situación tienen otros privilegios, se mantienen relajados y rinden al máximo.²⁴

2.5. SISTEMAS CONTABLES

Sistema Caja – Diario

En los negocios en general, la mayoría de las transacciones están relacionadas con el movimiento de dinero, lo cual se traduce en numerosos asientos repetitivos en el Libro Diario de contabilidad.

Con el objeto de evitar el continuo registro de asientos de ingreso y egreso de dinero se decidió llevar en forma separada todo lo correspondiente a movimientos de efectivo, es decir, de aquí nació el Sistema Caja – Diario.

Su operativo es similar al Sistema Normalizador, con la única diferencia que registra en forma separada las transacciones que significan movimiento de dinero²⁵.

Este sistema permitió empezar con la división del trabajo pero, también se encuentra en desuso, ya que no contempla todos los Diarios Auxiliares exigidos por las disposiciones legales vigentes.

²⁴ (Administrador de servidores, por Marchionni, Enzo Augusto , pag 23-30)

²⁵ (CONTABILIDAD GENERAL DE GUILLERMO GONZÁLEZ SAAVEDRA, 2003, paginas 55,56,57)

La secuencia grafica de este sistema es la siguiente:

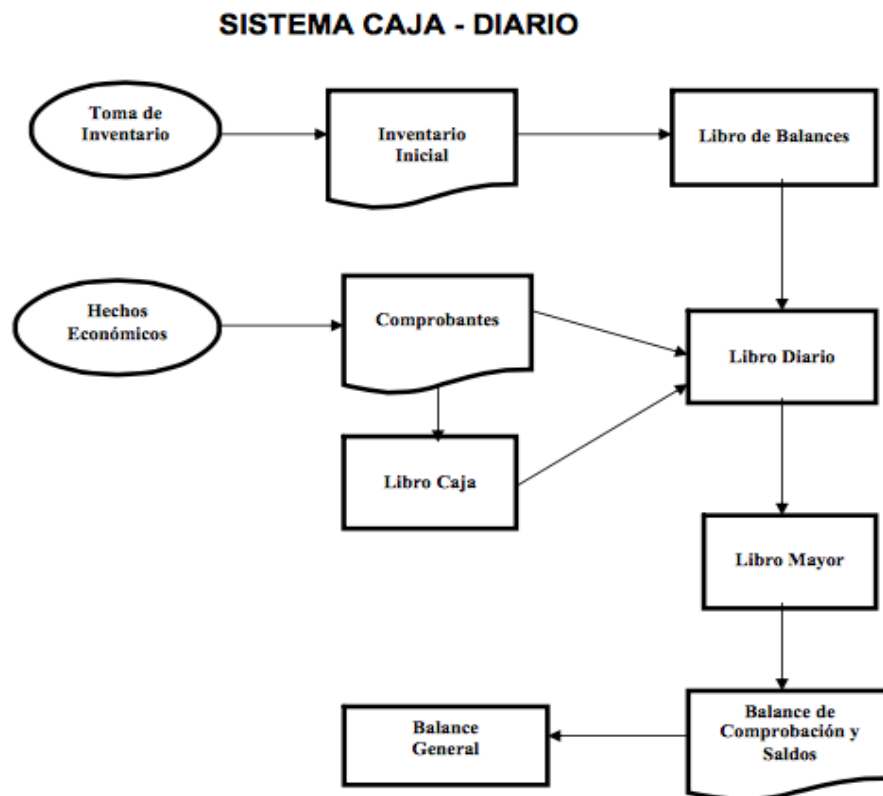


Figura 2.1 Sistema Caja-Diario²⁶.

SISTEMA CENTRALIZADOR O DE DIARIOS MÚLTIPLES

Pronto se advirtió que no solo las transacciones relacionadas con Caja eran repetitivas en una Empresa, existían muchas otras como por ejemplo: Compras, Ventas, Aceptación de Documentos que tenían iguales características²⁷.

Se hizo necesario implantar nuevas subdivisiones al Libro Diario creando Registros Auxiliares para todas aquellas transacciones de una misma naturaleza que se repetían con frecuencia. Es así como nace el Sistema Centralizador o de Diarios Múltiples.

La base de este sistema está en el hecho de que se pueden crear tantos Diarios como sean necesarios para la Empresa, no existe un número limitados de ellos.

El buen funcionamiento de este sistema depende:

- Del diseño que tengan los Libros Auxiliares.

²⁶ (CONTABILIDAD GENERAL DE GUILLERMO GONZÁLEZ SAAVEDRA, 2003, paginas 55)

²⁷ (CONTABILIDAD GENERAL DE GUILLERMO GONZÁLEZ SAAVEDRA, 2003, paginas 55,56,57)

- De su facilidad de uso y adaptación a las necesidades de información de la Empresa
- De la correcta sincronización y coordinación de los diferentes auxiliares entre sí. Ambos factores deben ser cuidadosamente considerados al implantar este sistema de modo que los auxiliares creados abarquen todas las actividades y operaciones principales que la Empresa realice y que su diseño permita la relación y complementación entre ellos.

Después de un periodo, generalmente un mes, se acostumbra a sumar todos los Diarios Auxiliares y traspasar esas cifras al Diario General a través de un asiento de centralización, que es el resumen de todas las anotaciones efectuadas en el registro auxiliar.

El Sistema Centralizador o de Diarios Múltiples tiene más ventajas que limitaciones:

- Permite al máximo la división del trabajo
- Proporciona información completa y detallada
- Al estar las operaciones agrupadas por funciones, se simplifica la anotación pudiendo encomendar ciertas labores simples a personas que no sean técnicos en la materia
- Se mantiene la cronología en cada Registro Auxiliar, pero se pierde el Diario General²⁸

²⁸ (CONTABILIDAD GENERAL DE GUILLERMO GONZÁLEZ SAAVEDRA, 2003, paginas 55,56,57)

SISTEMA CENTRALIZADOR

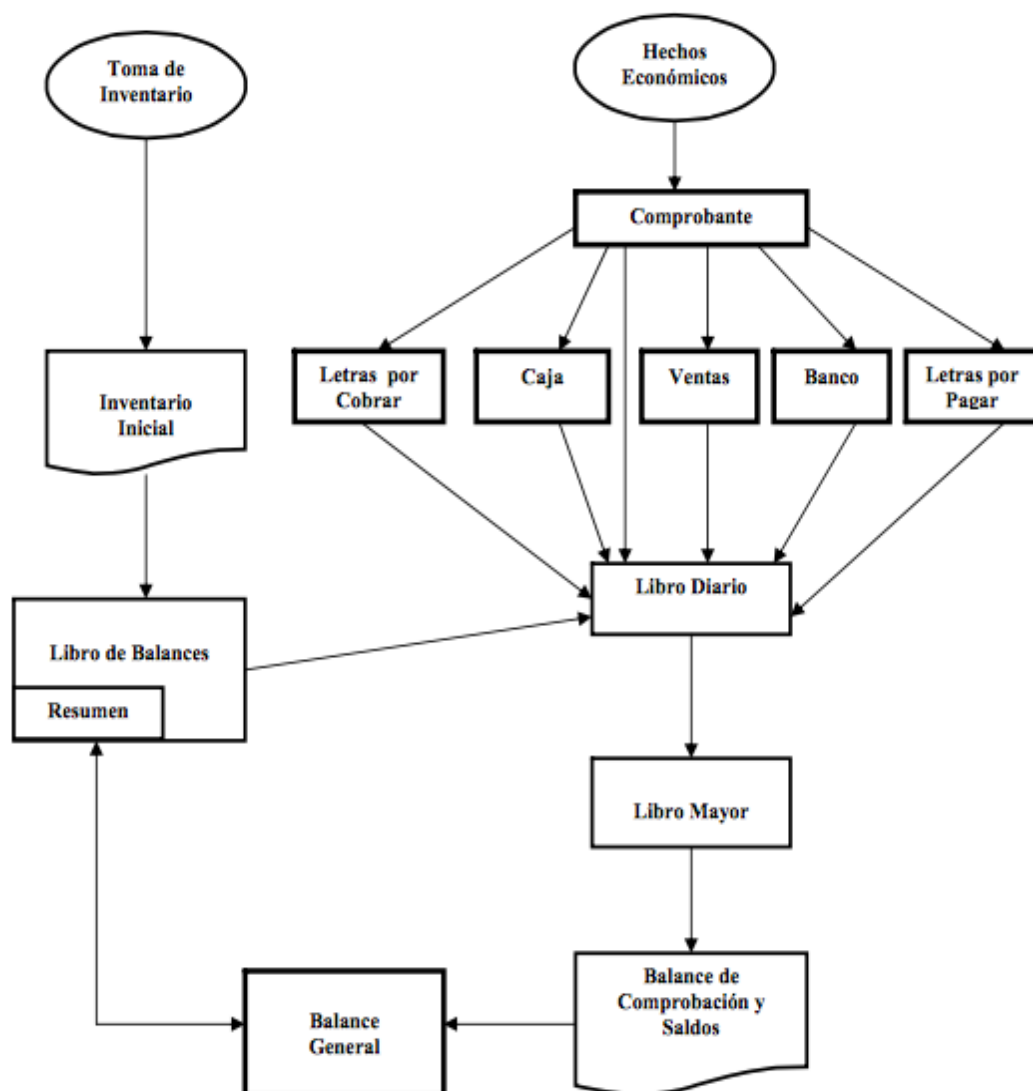


Figura 2.2 Sistema Centralizador²⁹.

²⁹ (CONTABILIDAD GENERAL DE GUILLERMO GONZÁLEZ SAAVEDRA, 2003, paginas 55,56,57)

2.6. NÓMINA

OBJETIVOS DE LA PREPARACIÓN DE LA NÓMINA

El objetivo principal de la labor de preparación de la nómina es el de calcular el importe de los sueldos o salarios devengados por el personal y pagar prontamente dichos importes. Otro objetivo es el de clasificar correctamente los pagos efectuados al personal cargar estos importes a las cuentas de gastos correspondientes. Hasta mediados del decenio de los años 30, la contabilidad de la nómina implicaba pocas otras consideraciones.

En 1935 se aprobó la ley federal sobre seguridad social. Con arreglo a esta legislación, fue necesario que las empresas llevaran unos registros detallados de la plantilla de personal y sus devengos. En 1943 empezó la retención del impuesto sobre la renta en origen, lo que aumento la labor de contabilización de la nómina.

Las disposiciones sobre salarios y horas de la ley federal de normas laborales justas de 1936(modificada) afecto al cálculo de los devengos. Las varias leyes estatales sobre seguro de accidentes de los trabajadores son también una ulterior incumbencia de la contabilidad de la nómina porque exigen una cuidadosa clasificación de las nóminas con arreglo al tipo de trabajo realizado por los trabajadores.³⁰

2.7. LIQUIDACION DE NÓMINA

- Valores vigentes para el 2015
Salario mínimo \$644.350 (decreto 2552 de 2015)
Auxilio de transporte \$74.000 (decreto 2553 de 2015)

Salario mínimo integral \$8.376.550 (\$6.443.500 salario y \$1.933.050 factor prestacional)

- Aportes parafiscales:
Sena 2% (Ley 21 de 1982)
ICBF 3% (Ley 89 de 1988)
Cajas de Compensación Familiar 4% (Ley 21 de 1982)
- Cargas Prestacionales
Cesantías 8.33% (Artículo 249 C.S.T)
Prima de servicios 8.33% (Artículo 306 C.S.T)

³⁰ (Contabilidad: principios y aplicaciones By Horace R Brock, Ch. E Palmer capitulo 15 pag 249)

Vacaciones 4.17% (Artículo 186 C.S.T)
Intereses sobre las Cesantías 1% mensual (Artículo 249 C.S.T)
Seguridad social:
Salud: (Ley 1122 del 2007 Artículo 10)
 Empresa 8.5%.
 Empleado 4%
Pensión:(Ley 797 de 2003 Artículo 7)
 Empresa 12%
 Empleado 4%

(ARTICULO 168 C.S.T). TASAS Y LIQUIDACION DE RECARGOS. Modificado por el art. 24, Ley 50 de 1990:

1. El trabajo nocturno por el solo hecho de ser nocturno se remunera con un recargo del treinta y cinco por ciento (35%) sobre el valor del trabajo diurno, con excepción del caso de la jornada de treinta y seis (36) horas semanales previstas en el artículo 20 161 literal c) de esta ley.
2. El trabajo extra diurno se remunera con un recargo del veinticinco por ciento (25%) sobre el valor del trabajo ordinario diurno.
3. El trabajo extra nocturno se remunera con un recargo del setenta y cinco por ciento (75%) sobre el valor del trabajo ordinario diurno.
4. Cada uno de los recargos antedichos se produce de manera exclusiva, es decir, sin acumularlo con alguno otro.

(ARTICULO 179 C.S.T). TRABAJO DOMINICAL Y FESTIVO. Modificado por el art. 26, Ley 789 de 2002:

1. El trabajo en domingo y festivos se remunerará con un recargo del setenta y cinco por ciento (75%) sobre el salario ordinario en proporción a las horas laboradas.
2. Si con el domingo coincide otro día de descanso remunerado solo tendrá derecho el trabajador, si trabaja, al recargo establecido en el numeral anterior.
3. Se exceptúa el caso de la jornada de treinta y seis (36) horas semanales previstas en el artículo 20 literal c) de la Ley 50 de 1990.

CONTRATO DE TRABAJO

• Contrato de trabajo a término fijo (ARTICULO 46 C.S.T Subrogado por el art. 3, Ley 50 de 1990)

El contrato de trabajo a término fijo debe constar siempre por escrito y su duración no puede ser superior a tres años, pero es renovable indefinidamente:

1. Si antes de la fecha del vencimiento del término estipulado, ninguna de las partes avisare por escrito a la otra su determinación de no prorrogar el contrato, con una antelación

no inferior a treinta (30) días, éste se entenderá renovado por un período igual al inicialmente pactado, y así sucesivamente.

2. No obstante, si el término fijo es inferior a un (1) año, únicamente podrá prorrogarse sucesivamente el contrato hasta por tres (3) períodos iguales o inferiores, al cabo de los cuales el término de renovación no podrá ser inferior a un (1) año, y así sucesivamente.

PARÁGRAFO. En los contratos a término fijo inferior a un año, los trabajadores tendrán derecho al pago de vacaciones y prima de servicios en proporción al tiempo laborado cualquiera que éste sea. (ARTÍCULO 46 CST).

• Contrato de trabajo a término indefinido (ARTICULO 47. Modificado por el art. 5, Decreto 2351 de 1965)

1. El contrato de trabajo no estipulado a término fijo, o cuya duración no esté determinada por la de la obra, o la naturaleza de la labor contratada, o no se refiera a un trabajo ocasional o transitorio, será contrato a término indefinido.

2. El contrato a término indefinido tendrá vigencia mientras subsistan las causas que le dieron origen, y la materia del trabajo. Con todo, el trabajador podrá darlo por terminado mediante aviso escrito con antelación no inferior a treinta (30) días, para que el patrono lo reemplace. En caso de no dar aviso oportunamente o de cumplirlo solo parcialmente, se aplicará lo dispuesto en el artículo 8o., numeral 7o., para todo el tiempo, o para el lapso dejado de cumplir. (ARTÍCULO 47 CST)

JORNADA DE TRABAJO (ARTICULO 160. Modificado por el art. 25 de la Ley 789 de 2002)

• Trabajo ordinario y nocturno

1. Trabajo ordinario es el que se realiza entre las seis horas (6:00 a.m.) y las veintidós horas (10:00 p.m.).

2. Trabajo nocturno es el comprendido entre las veintidós horas (10:00 p.m.) y las seis horas (6:00 a.m.).ARTÍCULO 160 CST.

SALARIO

Pagos que constituyen salario (ARTICULO 127 C.S.T, Modificado por el art. 14 de la Ley 50 de 1990)

Constituye salario no sólo la remuneración ordinaria, fija o variable, sino todo lo que recibe el

trabajador en dinero o en especie como contraprestación directa del servicio, sea cualquiera la forma o denominación que se adopte, como primas, sobresueldos, bonificaciones habituales, valor del trabajo suplementario o de las horas extras, valor del trabajo en días de descanso obligatorio, porcentajes sobre ventas y comisiones.

Pagos que no constituyen salario (ARTICULO 128 C.S.T, Modificado por el artículo 15 de la Ley 50 de 1990.)

No constituyen salario las sumas que ocasionalmente y por mera liberalidad recibe el trabajador del empleador, como primas, bonificaciones o gratificaciones ocasionales, participación de utilidades, excedentes de las empresas de economía solidaria y lo que recibe en dinero o en especie no para su beneficio, ni para enriquecer su patrimonio, sino para desempeñar a cabalidad sus funciones, como gastos de representación, medios de transporte, elementos de trabajo y otros semejantes. Tampoco las prestaciones sociales de que tratan los títulos VIII y IX, ni los beneficios o auxilios habituales u ocasionales acordados convencional o contractualmente u otorgados en forma extralegal por el {empleador}, cuando las partes hayan dispuesto expresamente que no constituyen salario en dinero o en especie, tales como la alimentación, habitación o vestuario, las primas extralegales, de vacaciones, de servicios o de navidad. Importante recordar que los pagos no constitutivos de salario no pueden superar el 40%³¹.

Explicación

Salario: Sueldo básico + auxilio de transporte + horas extras + c omisiones + viáticos.

*Sueldo Básico: Asignación básica mensual que se le da a la persona.

*Auxilio de transporte: tienen derecho al auxilio de transporte quienes devenguen un salario fijo, igual o inferior a 2 veces el S.M.L.V; su objetivo es que le trabajador cobre parte del desplazamiento entre el sitio de trabajo y su lugar de residencia.

Deducciones.

Valores que se aplican con descuentos, pueden existir deducciones por concepto de libranzas, embargos judiciales entre otras, pero independiente de ello, deducciones obligatorias para el trabajador como los aportes a la seguridad a la seguridad social y las retenciones.

Aportes parafiscales:

³¹ (Contabilidad: principios y aplicaciones By Horace R Brock, Ch. E Palmer capitulo 15 pag 249)

1. Cajas de Compensación Familiar. Ley 21 de 1982
2. Servicio Nacional de Aprendizaje (SENA). Ley 21 de 1982
3. Instituto Colombiano de Bienestar Familiar (ICBF) Ley 89 de 1988

Seguridad social:

La ley 100/93 creo en Colombia el sistema de Seguridad Social Integral (SSSI) constituido por tres regímenes:

- a. Régimen Pensional. Ley 797 de 2003 Artículo 7
- b. Régimen salud. Ley 1122 de 2007 Artículo 10
- c. Régimen Riesgos Profesionales. Decreto 1772 de 1994 Artículo 13

La ley 1607 de 2012, exonera de los aportes que sobre la nómina se efectúan al SENA, al ICBF y los obligatorios de salud, todos a cargo del empleador. Este dinero que se dejará de recaudar por estos conceptos, se cubrirá con el impuesto de renta para la equidad –CREE–.

Las persona beneficiarias de esta exoneración según la citada ley reglamentada por el decreto 862 del 26 de Abril de 2013 son³²:

- Todos las personas contribuyentes del CREE, es decir las personas jurídicas contribuyentes declarantes del impuesto de renta, solo sobre la nómina de trabajadores que devenguen, individualmente considerados, menos de diez (10) salarios mínimos mensuales legales vigentes.
- Las personas naturales empleadoras que tengan dos o más trabajadores a su cargo, solo sobre la nómina de trabajadores que devenguen, individualmente considerados, menos de diez (10) salarios mínimos mensuales legales vigentes.

Ahora, las personas que no son beneficiarias de esta exoneración de aportes sobre la nómina son:

- Las personas jurídicas no contribuyentes del impuesto sobre la renta, por ejemplo, los sindicatos, las juntas de acción comunal, las juntas de copropiedad horizontal y en general las personas detalladas en los artículos 22,23,23-1 y 23-2 del Estatuto Tributario Nacional.
- Las personas jurídicas sin ánimo de lucro, por ejemplo las cooperativas, los fondos de empleados, las asociaciones, corporaciones y fundaciones, todas sin ánimo de lucro.

³² http://www.consultame.co/index.php?option=com_content&view=article&id=141:exoneracion-de-aportes-de-nomina-ley-1607-de-2012&catid=43:impuestos&Itemid=155

- Las Zonas Francas declaradas al 31 de Diciembre de 2012 o con solicitud en trámite a esa fecha.
- Los usuarios de las anteriores Zonas Francas que se hayan calificado o se califiquen a futuro en éstas que se encuentren sujetos a la tarifa especial del impuesto sobre la renta del 15% establecida en el inciso primero del artículo 240-1 del Estatuto Tributario.
- Las personas naturales que tengan menos de dos trabajadores a su cargo.

CAPITULO 3

METODOLOGÍA

3.1.DISEÑO METODOLÓGICO

1. Se elaboró un diagnóstico inicial que reúne de forma integral todas las actividades relacionadas con la problemática planteada.
2. Se organizó la información acorde con las prioridades planteadas por la Institución y acorde con el orden sistemático que requiere el proceso.
3. Se adquirió la información necesaria de la institución para la creación de las bases de datos (figura 3.1 Estructura de base de datos en el software).
4. Desarrollo de actividades de programación con base en la estructura del plan de acción.
5. Se dará la capacitación necesaria al personal administrativo que lo requiera.
6. Se harán las recomendaciones y sugerencias necesarias que permitan el mejoramiento de las actividades.

3.2. Modelo de base de datos estructurada en el software

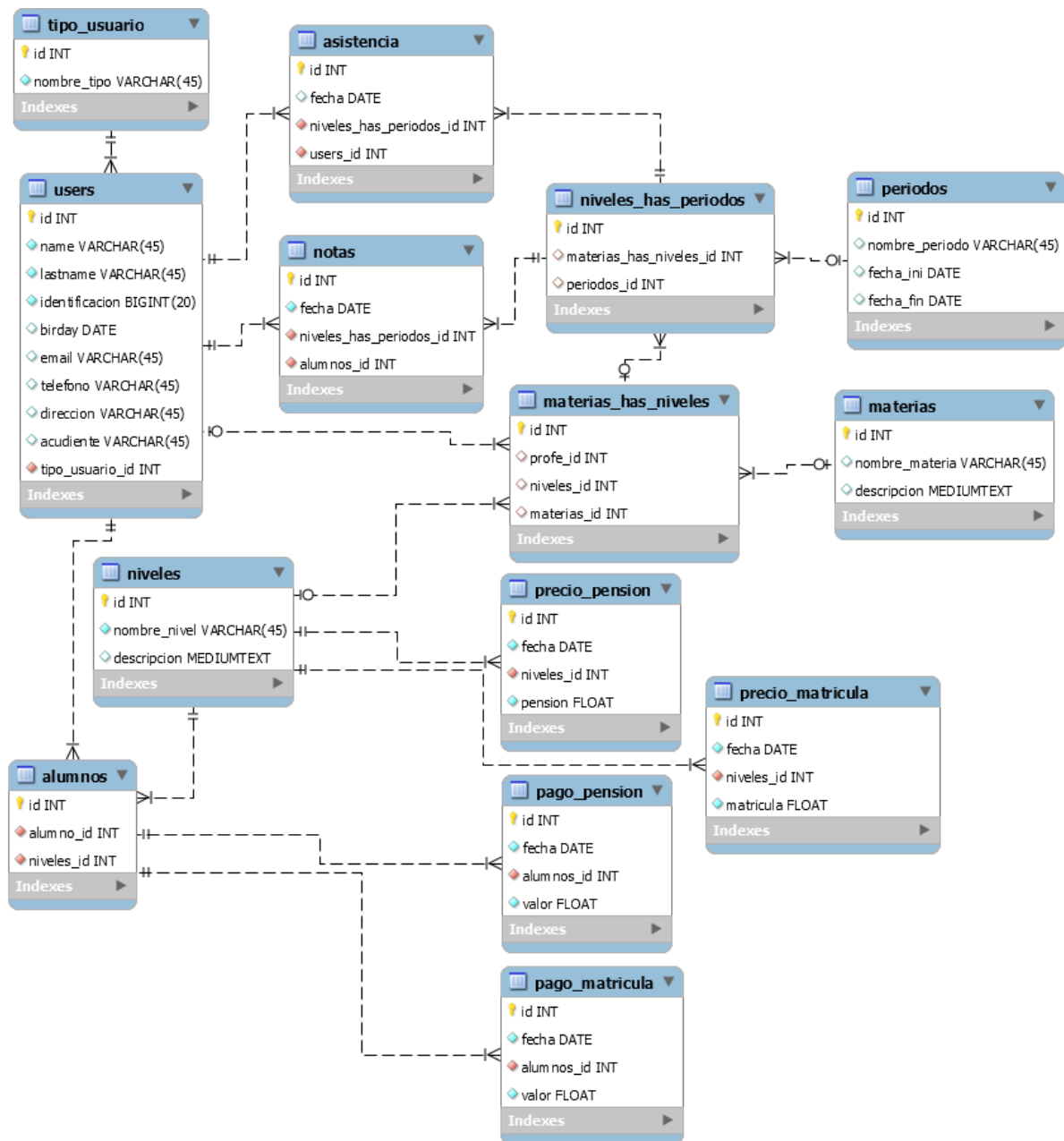


Figura 3.1 Estructura base de datos del software.

CAPÍTULO 4

EJECUCIÓN DEL PROYECTO

4.1. DESARROLLO

Inicialmente se planteó el desarrollo del proyecto usando Netbeans IDE, programando bajo lenguaje Java, debido a que es bastante conocido por lo cual existe muy amplia documentación sobre su manejo, librerías, etc; pero durante este desarrollo se encontró un gran tropiezo ya que los servidores para aplicaciones web hechas en java necesitan tener una máquina virtual del mismo, lo cual eleva el costo de alojamiento de la aplicación de una forma exagerada, perdiendo toda funcionalidad para una empresa y un desarrollo de este tipo, ya que a nivel competitivo económicamente sería inferior.

Teniendo en cuenta las razones expuestas anteriormente se procedió a cambiar el lenguaje de programación por PHP el cual se programó bajo el framework Laravel, combinando las funcionalidades de HTML5, teniendo la ventaja que para la implementación de la aplicación web los costos son bastante inferiores a los que se requieren con JAVA, ya que de esta forma, el alojamiento en los servidores tiene un costo anual muy bajo lo cual aumenta su competitividad en cuanto a costo y desarrollo, está abierto a futuras actualizaciones incluso a nivel de aplicación móvil.

La base de datos se realizó en MySQL ya que es una plataforma lo suficientemente robusta para la aplicación, es fácilmente compatible con PHP, y existe gran cantidad de documentación para su manejo e integración con PHP, MySQL cuenta con una versión gratuita la cual se acopla a las necesidades del software.

El software se realizó para aplicación Web, de esta forma se puede consultar y actualizar información desde distintos puntos, esto es significativamente útil en la digitación y consulta de notas.

Se tienen distintos tipos de usuarios, los cuales pueden visualizar y/o modificar información, según corresponda a su debido perfil de usuario, básicamente se maneja un usuario administrador el cual tiene acceso total a la aplicación y de igual manera podrá modificar información, los otros 2 tipos de usuarios que se tienen son empleado y estudiante, estos podrán visualizar y modificar información de forma restringida según lo requiera la institución.

Todo el personal y estudiantado fue carnetizado por la institución, el diseño fue realizado de manera propia, asignando un código de barras a cada persona para su identificación y uso del sistema que se instalará. Este dará campo al control de ingreso y salida del personal o estudiantado por medio de lectores infrarrojos que serán previamente instalados en lugares estratégicos de acuerdo con las normas y peticiones de la institución.

Lo anterior muestra el funcionamiento general del software. A continuación se muestra de forma gráfica el contenido general del software.

4.2. ESQUEMA GENERAL DEL SOFTWARE

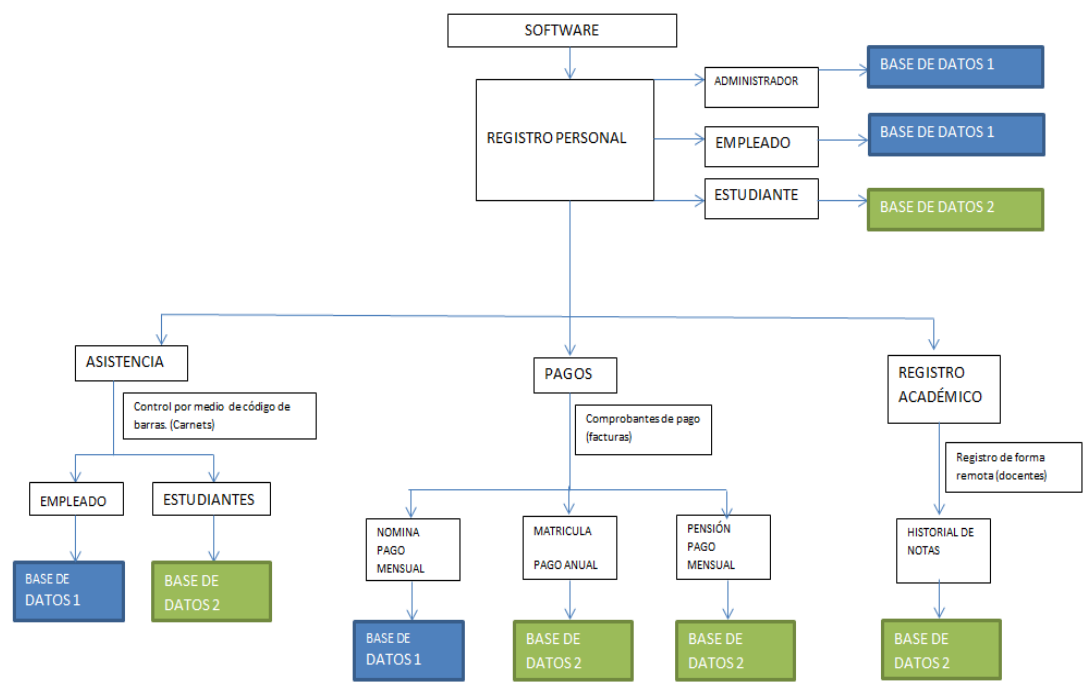


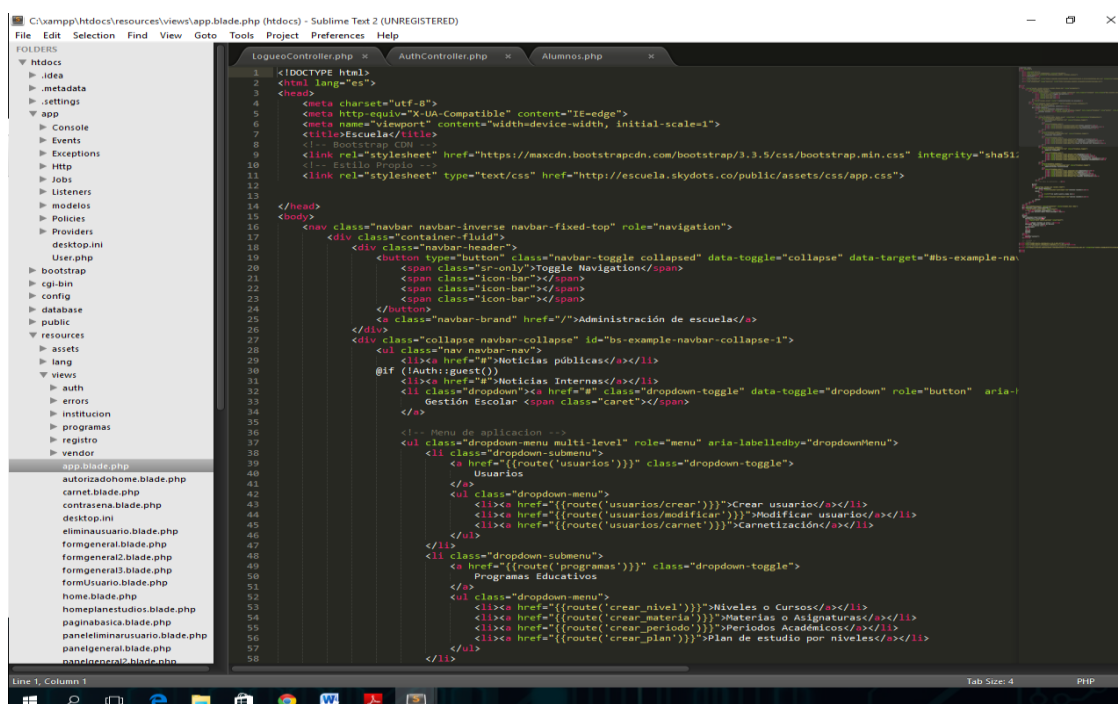
Figura 4.1. Esquema general del software.

4.3. IMPLEMENTACIÓN DEL FRAMEWORK

LARAVEL

Se hace la apertura de Laravel en cualquier editor de texto, en este caso se utilizó Sublime Text 2 como editor, ya que ofrece una forma organizada para escribir código y es bastante recomendado por los programadores debido a que fue creado para este tipo de aplicaciones de desarrollo de software.

Como servidor de aplicación web, se utilizó el servidor XAMPP el cual tiene sistema de gestión de bases de datos en MySQL, servidor web apache y también interprete para lenguaje PHP, lo cual cumple con las características de la aplicación, cabe resaltar que este servidor se usa en las etapas de pruebas, y cuando la Institución lo crea conveniente se contratara un servicio de Hosting con su respectivo dominio en internet.



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1">
7 <title>Escuela</title>
8 <!-- Bootstrap CSS -->
9 <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css" integrity="sha512-Lda7254742p58AD4F6YQ6a0CH6N1CLW0y3E8NXas9T" crossorigin="anonymous">
10 <!-- Estilos Propios -->
11 <link rel="stylesheet" type="text/css" href="http://escuela.skydots.co/public/assets/css/app.css">
12
13 </head>
14 <body>
15 <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
16 <div class="container-fluid">
17 <div class="navbar-header">
18 <button type="button" class="navbar-toggle collapsed" data-toggle="collapse" data-target="#bs-example-navbar-collapse-1">
19 <span class="sr-only">Toggle Navigation</span>
20 <span class="icon-bar"></span>
21 <span class="icon-bar"></span>
22 <span class="icon-bar"></span>
23 </button>
24 <a class="navbar-brand" href="/">Administración de escuela</a>
25 </div>
26 <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
27 <ul class="nav navbar-nav">
28 <li><a href="/">Inicio</a></li>
29 <li><a href="/noticias">Noticias públicas</a></li>
30 <li><a href="/noticias-internas">Noticias Internas</a></li>
31 <li><a href="/gestion-escolar">Gestión Escolar</a></li>
32 </ul>
33 <div class="dropdown"><a href="#" class="dropdown-toggle" data-toggle="dropdown" role="button" aria-expanded="false">
34 <span class="caret"></span>
35 </a>
36 <div class="dropdown-menu" role="menu" aria-labelledby="dropdownMenu">
37 <li class="dropdown-submenu">
38 <a href="{{route('usuarios')}}" class="dropdown-toggle">
39 <span>Usuarios</span>
40 </a>
41 <div class="dropdown-menu">
42 <li><a href="{{route('usuarios/crear')}}">Crear usuario</a></li>
43 <li><a href="{{route('usuarios/modificar')}}">Modificar usuario</a></li>
44 <li><a href="{{route('usuarios/carnet')}}">Carnetización</a></li>
45 </div>
46 </li>
47 <li class="dropdown-submenu">
48 <a href="{{route('programas')}}" class="dropdown-toggle">
49 <span>Programas Educativos</span>
50 </a>
51 <div class="dropdown-menu">
52 <li><a href="{{route('crear_nivel')}}">Niveles o Cursos</a></li>
53 <li><a href="{{route('crear_materia')}}">Materias o Asignaturas</a></li>
54 <li><a href="{{route('crear_periodo')}}">Periodos Académicos</a></li>
55 <li><a href="{{route('crear_plan')}}">Plan de estudio por niveles</a></li>
56 </div>
57 </li>
58 </div>
59 </div>
60 </div>
61 </body>
62 </html>
```

Figura 4.2. Visualización del código en Sublime Text 2

SERVIDOR XAMPP

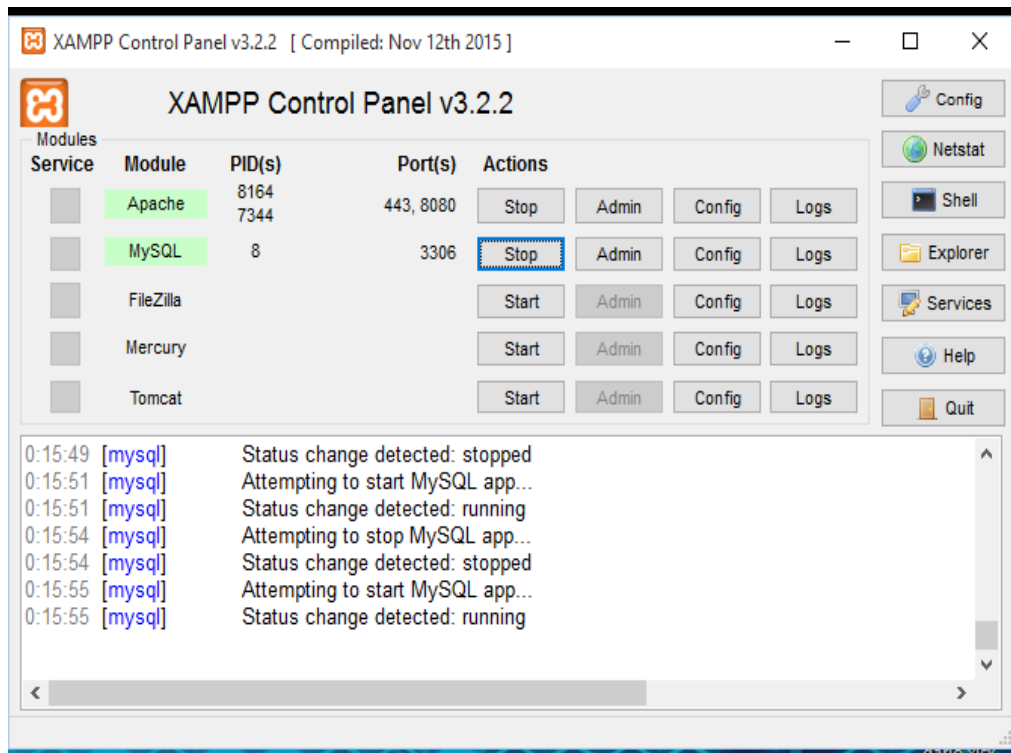


Figura 4.3. Iniciando servicios del servidor web.

FRAMEWORK (VISTAS BOOSTRAP)

El Framework de Bootstrap permite el uso de diseños predefinidos en CSS y JavaScript de fácil configuración para las páginas web, con esto se puede reducir en gran medida el tiempo de programación, se deben vincular los archivos Bootstrap al proyecto direccionando el código de programación a un URL donde se encuentran previamente diseñadas las acciones graficas a usar, posteriormente al direccionamiento es necesario configurar los atributos de cada herramienta.

En la siguiente imagen se muestra la vinculación de Bootstrap al proyecto utilizando el método de CDN en el cual se vincula desde los Servidores de Bootstrap sin tener que descargar los archivos localmente.

```

LogueoController.php x
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="utf-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1">
7   <title>Escuela</title>
8   <!-- Bootstrap CDN -->
9   <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css" integrity="sha512-dtfge/zgoMypP7Qb"
10  <!-- Estilo Propio -->
11   <link rel="stylesheet" type="text/css" href="http://escuela.skydots.co/public/assets/css/app.css">
12
13
14 </head>
15 <body>
16   <nav class="navbar navbar-inverse navbar-fixed-top" role="navigation">
17     <div class="container-fluid">

```

Figura 4.4 Vinculación de Bootstrap al proyecto

```

35
36 <!-- Menu de aplicacion -->
37 <ul class="dropdown-menu multi-level" role="menu" aria-labelledby="dropdownMenu">
38   <li class="dropdown-submenu">
39     <a href="{{route('usuarios')}}" class="dropdown-toggle">
40       Usuarios
41     </a>
42     <ul class="dropdown-menu">
43       <li><a href="{{route('usuarios/crear')}}">Crear usuario</a></li>
44       <li><a href="{{route('usuarios/modificar')}}">Modificar usuario</a></li>
45       <li><a href="{{route('usuarios/carnet')}}">Carnetización</a></li>
46     </ul>
47   </li>
48   <li class="dropdown-submenu">
49     <a href="{{route('programas')}}" class="dropdown-toggle">
50       Programas Educativos
51     </a>
52     <ul class="dropdown-menu">
53       <li><a href="{{route('crear_nivel')}}">Niveles o Cursos</a></li>
54       <li><a href="{{route('crear_materia')}}">Materias o Asignaturas</a></li>
55       <li><a href="{{route('crear_periodo')}}">Periodos Académicos</a></li>
56       <li><a href="{{route('crear_plan')}}">Plan de estudio por niveles</a></li>
57     </ul>
58   </li>
59   <li class="dropdown-submenu">
60     <a href="{{route('registro')}}" class="dropdown-toggle">
61       Registro Académico
62     </a>
63     <ul class="dropdown-menu">
64       <li><a href="{{route('crear_alumno')}}">Alumnos</a></li>
65       <li><a href="{{route('crear_profesor')}}">Profesores</a></li>
66       <li><a href="{{route('crear_asistencia')}}">Asistencia</a></li>
67       <li><a href="{{route('crear_rendimiento')}}">Rendimiento Académico</a></li>
68       <li><a href="{{route('crear_estudiantil')}}">Registro estudiantil</a></li>
69     </ul>
70   </li>
71   <li class="dropdown-submenu">
72     <a href="{{route('institucion')}}" class="dropdown-toggle">
73       Administración Institucional
74     </a>
75     <ul class="dropdown-menu">
76       <li><a href="{{route('crear_empleado')}}">Empleados</a></li>
77       <li><a href="{{route('crear_nomina')}}">Nómina</a></li>
78       <li><a href="{{route('crear_pension')}}">Pensiones</a></li>
79       <li><a href="{{route('crear_matricula')}}">Matrículas</a></li>
80       <li><a href="{{route('crear_estado')}}">Estado Financiero</a></li>
81     </ul>
82   </li>
83 </ul>
84 <!-- Fin Menu de aplicacion --></li>

```

Figura 4.5. Ejemplo de uso de un dropdown de Bootstrap.

CONTROLADORES

Se definió la acción que puede tener cada espacio o botón que interactúa con el usuario para el funcionamiento acertado y acorde con la ruta de direccionamiento del software.

```
1 <?php
2
3 namespace App\Http\Controllers\logueo;
4
5 use Illuminate\Http\Request;
6
7 use App\Models\Users;
8 use App\Http\Requests;
9 use App\Http\Requests\VerificarLoginRequest;
10 use App\Http\Controllers\Controller;
11 use Auth;
12
13 class LogueoController extends Controller
14 {
15     /**
16      * Show the form for creating a new resource.
17      *
18      * @return \Illuminate\Http\Response
19      */
20     public function getLogin()
21     {
22         return view('login');
23     }
24
25     /**
26      * Store a newly created resource in storage.
27      *
28      * @param \Illuminate\Http\Request $request
29      * @return \Illuminate\Http\Response
30      */
31     public function postLogin(VerificarLoginRequest $request)
32     {
33         if(Auth::attempt(['identificacion'=>$request->input('identificacion'),'password'=>$request->input('password')],$request->input('remember')))
34             return redirect('autorizado/home');
35         }else{
36             return redirect('/');
37         }
38     }
39
40     /**
41      * Show the form for creating a new resource.
42      *
43      * @return \Illuminate\Http\Response
44      */
45     public function getLogout()
46     {
47         Auth::logout();
48         return redirect('/');
49     }
50
51
52
53 }
```

Figura 4.6. Controlador de autenticación de ingreso en el software.

MIGRACIONES (CREACION DE TABLAS DE DATOS)

En las migraciones se definió el esquema y contenido de las tablas de datos del software, estipulando por cada elemento su clase y magnitud de almacenamiento en cada tabla.

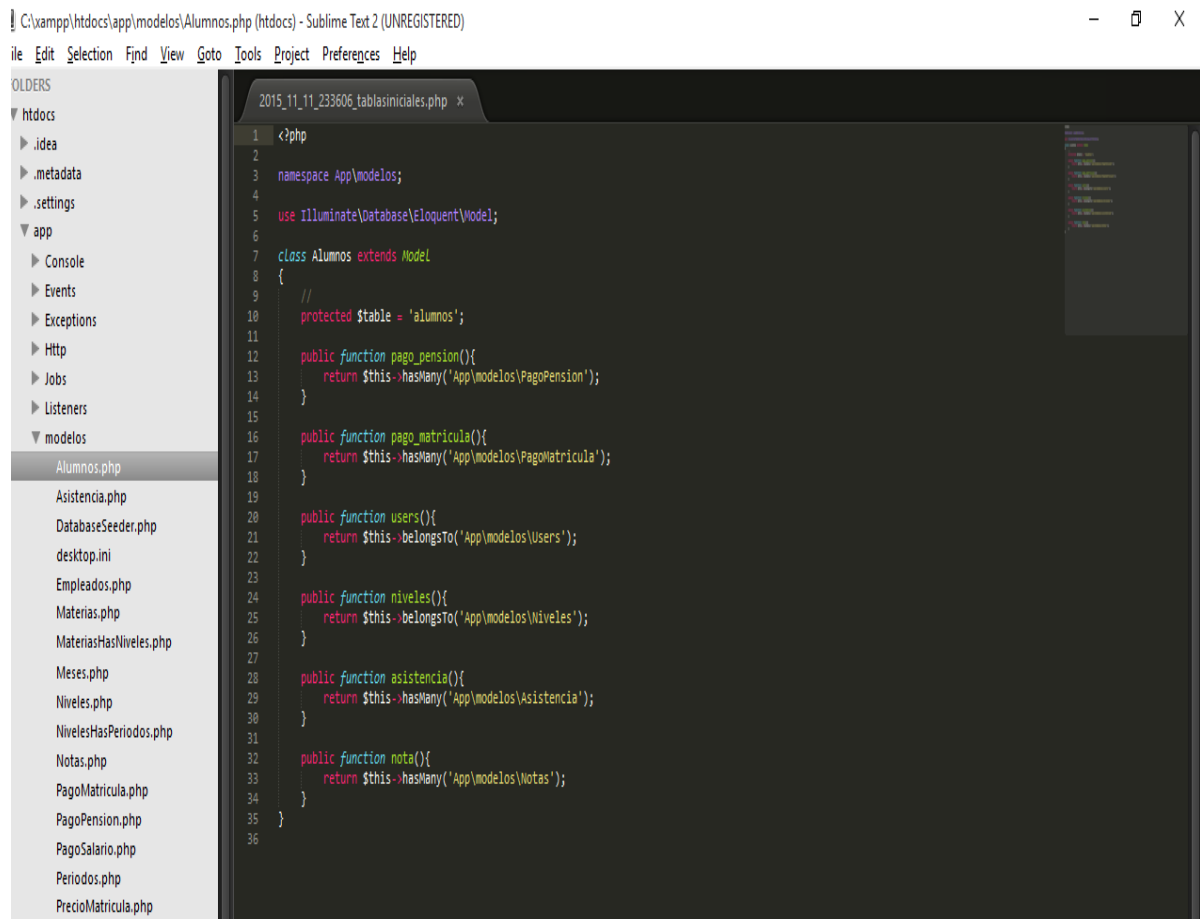
```
_11_11_233606_tablasiniciales.php (htdocs) - Sublime Text 2 (UNREGISTERED)
Tools Project Preferences Help

2015_11_11_233606_tablasiniciales.php x
1 <?php
2
3 use Illuminate\Database\Schema\Blueprint;
4 use Illuminate\Database\Migrations\Migration;
5
6 class Tablasiniciales extends Migration
7 {
8     /**
9      * Run the migrations.
10     */
11     * @return void
12     */
13     public function up()
14     {
15         //
16         if (!Schema::hasTable('users')) {
17             Schema::create('users', function (Blueprint $table) {
18                 //Creacion de columnas
19                 $table->increments('id')->unique();
20                 $table->string('name');
21                 $table->string('lastname');
22                 $table->bigInteger('identificacion');
23                 $table->date('birthday');
24                 $table->string('email');
25                 $table->string('password', 60);
26                 $table->string('telefono');
27                 $table->string('direccion');
28                 $table->string('acudiente');
29                 $table->string('tipo_sangre');
30                 $table->integer('tipo_usuario_id');
31                 $table->rememberToken();
32                 $table->timestamps();
33             });
34         }
35         if (!Schema::hasTable('tipo_usuario')) {
36             Schema::create('tipo_usuario', function (Blueprint $table) {
37                 $table->increments('id')->unique();
38                 $table->string('nombre_tipo');
39                 $table->timestamps();
40             });
41         }
42         if (!Schema::hasTable('niveles')) {
43             Schema::create('niveles', function (Blueprint $table) {
44                 $table->increments('id')->unique();
45                 $table->string('nombre_nivel');
46                 $table->mediumText('descripcion');
47                 $table->timestamps();
48             });
49         }
50         if (!Schema::hasTable('materias')) {
51             Schema::create('materias', function (Blueprint $table) {
52                 $table->increments('id')->unique();
53                 $table->string('nombre_materia');
54                 $table->mediumText('descripcion');
55                 $table->timestamps();
56             });
57         }
58         if (!Schema::hasTable('periodos')) {
59             Schema::create('periodos', function (Blueprint $table) {
60                 $table->increments('id')->unique();
61                 $table->string('nombre_periodo');
62                 $table->mediumText('descripcion');
63                 $table->date('fecha_ini');
64                 $table->date('fecha_fin');
65                 $table->timestamps();
66             });
67         }
68     }
69 }
```

Figura 4.7. Definición de Tablas de Base de Datos.

MODELOS

En los modelos se realizó el enlace de las tablas de la base datos, ya que en esta parte se cumple la función de acceder a los datos modificarlos y demás funciones correspondientes al manejo de información de las tablas.



```
1 <?php
2
3 namespace App\models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class Alumnos extends Model
8 {
9     //
10     protected $table = 'alumnos';
11
12     public function pago_pension(){
13         return $this->hasMany('App\models\PagoPension');
14     }
15
16     public function pago_matricula(){
17         return $this->hasMany('App\models\PagoMatricula');
18     }
19
20     public function users(){
21         return $this->belongsTo('App\models\Users');
22     }
23
24     public function niveles(){
25         return $this->belongsTo('App\models\Niveles');
26     }
27
28     public function asistencia(){
29         return $this->hasMany('App\models\Asistencia');
30     }
31
32     public function nota(){
33         return $this->hasMany('App\models\Notas');
34     }
35 }
36
```

Figura 4.8. Modelo (tabla alumno).

RUTAS

En las Rutas se realizó el direccionamiento de cada una de las ventanas del software según corresponde en el funcionamiento estipulado en los controladores.

```
/*Rutas de logueo y deslogueo*/
Route::get('auth/login', ['as'=>'auth/login','uses'=>'Auth\AuthController@login']);
Route::post('auth/login', ['as'=>'auth/login','uses'=>'logueo\LogueoController@postlogin']);
Route::get('auth/logout', ['as'=>'auth/logout','uses'=>'Auth\AuthController@getLogout']);

//Rutas de acceso restringido.
Route::group(['middleware'=>'auth'], function(){
    Route::get('/autorizado/home', ['as'=>'autorizado/home', 'uses'=>'autorizado\AutorizadoHomeController@index']);
    Route::group(['prefix'=>'usuarios'],function (){
        Route::get('/', ['as'=>'usuarios', 'uses'=>'usuarios\UsuariosController@index']);
        Route::get('/modificar', ['as'=>'usuarios/modificar', 'uses'=>'usuarios\UsuariosController@getBuscar']);
        Route::post('/modificar', ['as'=>'usuarios/modificar', 'uses'=>'usuarios\UsuariosController@postModificar']);
        Route::post('/modificapassword', ['as'=>'usuarios/modificapassword', 'uses'=>'usuarios\UsuariosController@postModificaPassword']);
        Route::get('/crear', ['as'=>'usuarios/crear', 'uses'=>'usuarios\UsuariosController@getNuevo']);
        Route::post('/registrar', ['as'=>'usuarios/registrar', 'uses'=>'usuarios\UsuariosController@postRegistrar']);
        Route::post('/editar', ['as'=>'usuarios/editar', 'uses'=>'usuarios\UsuariosController@gactualizar']);
        Route::post('/actualizar', ['as'=>'usuarios/actualizar', 'uses'=>'usuarios\UsuariosController@pactualizar']);
        Route::post('/eliminar', ['as'=>'usuarios/eliminar', 'uses'=>'usuarios\UsuariosController@eliminarUsuario']);
        Route::get('/carnet', ['as'=>'usuarios/carnet', 'uses'=>'usuarios\UsuariosController@getCarnet']);
    });
});

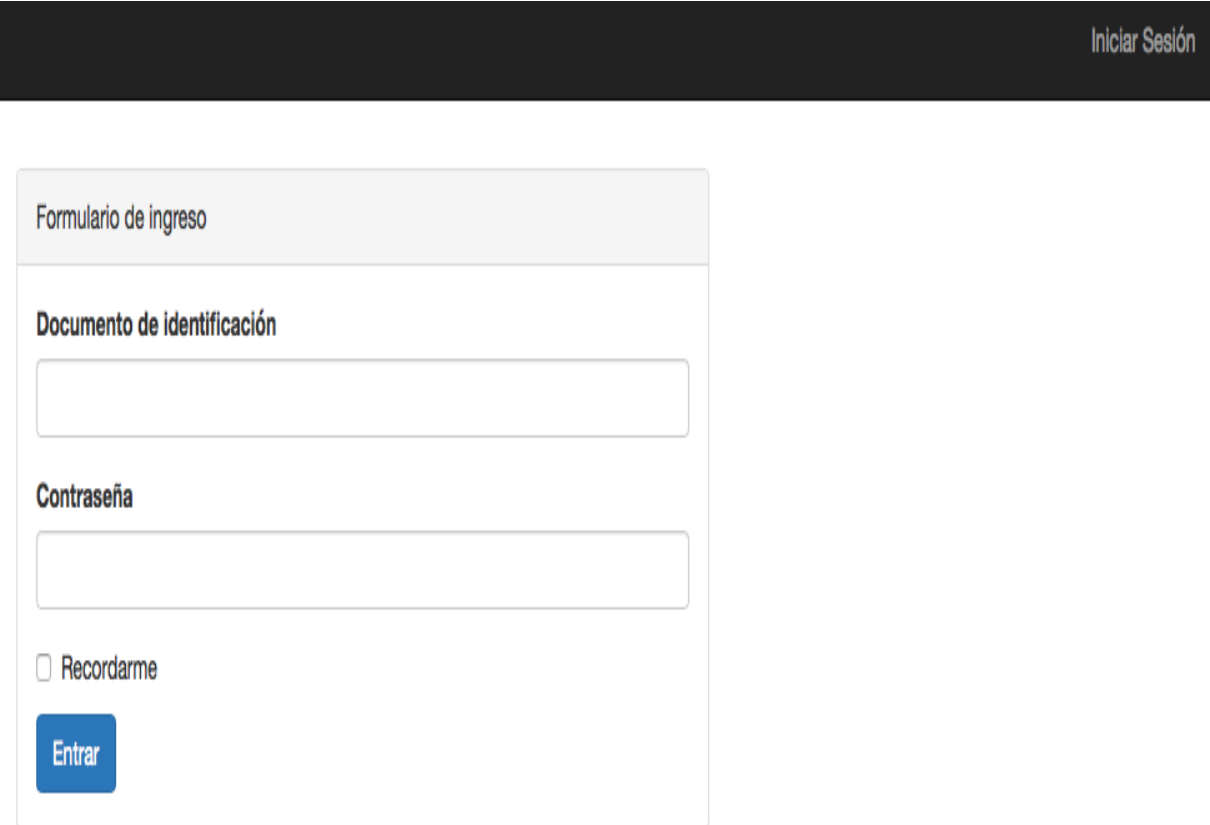
Route::group(['prefix'=>'programas'],function(){
    Route::group(['namespace'=>'programas'],function(){
        Route::get('/', ['as'=>'programas', 'uses'=>'ProgramasController@index']);

        Route::group(['prefix'=>'/nivel'], function(){
            Route::get('/crear', ['as'=>'crear_nivel', 'uses'=>'ProgramasController@getNivelCrear']);
            Route::post('/crear', ['as'=>'crear_nivel', 'uses'=>'ProgramasController@postNivelCrear']);
            Route::post('/editar', ['as'=>'editar_nivel', 'uses'=>'ProgramasController@getNivelEditar']);
            Route::put('/editar', ['as'=>'editar_nivel', 'uses'=>'ProgramasController@putNivelEditar']);
            Route::delete('/editar', ['as'=>'editar_nivel', 'uses'=>'ProgramasController@deleteNivelEditar']);
        });
    });
});
```

Figura 4.9. Definición de rutas de acceso en el software

4.4. AUTENTICACION DE USUARIO EN EL SOFTWARE

Después de haber realizado el registro en la base de datos es asignado un usuario y contraseña permitiendo el uso del software dependiendo su perfil registrado.



The image shows a login interface. At the top, there is a dark horizontal bar with the text "Iniciar Sesión" in white. Below this, there is a light gray box titled "Formulario de ingreso". Inside this box, there are two input fields: the first is labeled "Documento de identificación" and the second is labeled "Contraseña". Below the password field, there is a checkbox labeled "Recordarme". At the bottom of the form, there is a blue button labeled "Entrar".

Figura 4.10 ingreso al software requiere: usuario y contraseña.

4.5. REGISTRO PERSONAL

Para darle inicio al software fue necesario crear una base de datos la cual tiene como objetivo obtener los datos necesarios para el funcionamiento de este, para este fueron creados diferentes perfiles que se presentan en un plantel educativo.

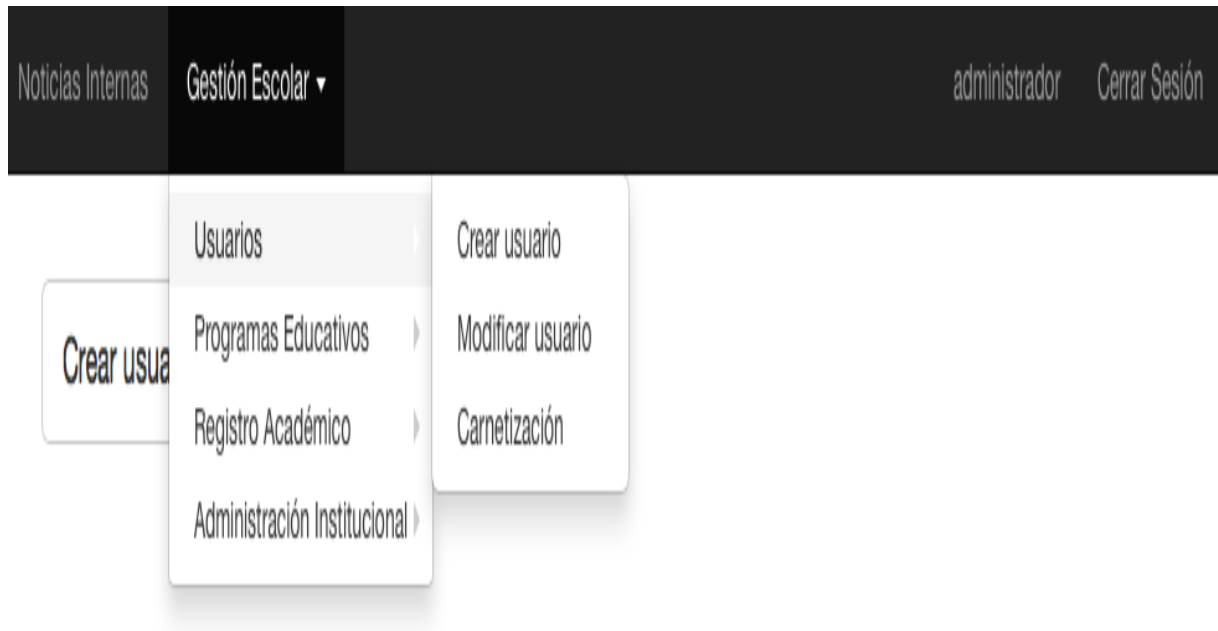


Figura 4.11 panel de selección entre crear docente y alumno (opción: registro, búsqueda o modificación).

4.6. REGISTRO POR PERFILES

PERFIL DEL ESTUDIANTE Y ADMINISTRATIVO

En el formato de perfiles de estudiantes y administrativo, la única diferencia que se presenta es el tipo de permiso de acceso.

Formulario para registrar los usuarios

Nombres

Digite nombres

Apellidos

Digite los apellidos

No. Identificación

Introduzca su numero de identificación

Fecha de nacimiento

dd/mm/aaaa

Correo electrónico

correo@dominio.com

Teléfono

XX X XXXXXXX

Dirección de residencia

CLL 19 Z BIS SUR # 20 A BIS ESTE 30

Nombre de Acudiente

Nombre de persona a cargo

Tipo de permiso de acceso

Alumno

Contraseña

Introduzca o cambie la Contraseña

Verificación de contraseña

Vuelva a ingresar contraseña

Registrar

Figura 4.12 Registro del estudiante (datos requeridos por el software).

En el registro del estudiante se presenta la plantilla de datos requeridos por el software asignando una contraseña y un ID para el uso de este.

PERFIL ADMINISTRATIVO

Formulario para registrar los usuarios

Nombres

Digite nombres

Apellidos

Digite los apellidos

No. Identificación

Introduzca su numero de identificación

Fecha de nacimiento

dd/mm/aaaa

Correo electrónico

correo@dominio.com

Teléfono

XX X XXXXXXX

Dirección de residencia

CLL 19 Z BIS SUR # 20 A BIS ESTE 30

Nombre de Acudiente

Nombre de persona a cargo

Tipo de permiso de acceso

Profesor

Contraseña

Introduzca o cambie la Contraseña

Verificación de contraseña

Vuelva a ingresar contraseña

Registrar

Figura 4.13 Registro administrativo (datos requeridos por el software).

63

En el registro administrativo se presenta la plantilla de datos requeridos por el software asignando una contraseña y un ID para el ingreso al software; se requiere seleccionar que perfil se tiene como (docente de planta, coordinador, director o administrador del software).

4.7. BÚSQUEDA Y MODIFICACIÓN DEL PERFIL

Formulario de vista/modificación de usuarios

Criterio de Búsqueda

Digite búsqueda

Buscar

Resultados de la búsqueda

Nombres	Apellidos	Identificación	Fecha de nacimiento	Correo	Teléfono	Dirección	Acudiente	Nivel de acceso	Acciones
juan	rodriguez	1026279934	1992-06-15	juan.pbooks@hotmail.com	3172438244	cil 67 a bis	alfredo	4	<div>Modificar</div>
sebastian	perez					a			<div>Eliminar</div>

Figura 4.14 Búsqueda y Modificación del perfil.

- Se presenta una vista previa de la base de datos existente.
- En el cuadro de texto es habilitado para ingresar un valor de búsqueda.
- La modificación se habilita en caso de actualización de datos del usuario, (las modificaciones solo las puede realizar el administrador del software).

4.8. MATRÍCULA DEL ALUMNO

Datos de ingreso:

Usuario

Nombre: juan perez -- Identificación: 1026279934

Nivel

once

Crear alumno

Editar alumno

Busque el nombre o palabras clave para el alumno. Se extenderá una tabla donde podrá elegir la acción y el elemento donde desea ejecutarlo.

Busqueda de palabras clave:

Texto a buscar

Buscar

Figura 4.15 Matricula del alumno.

Se realiza la actualización de la matrícula por año de cada estudiante teniendo en cuenta el costo de su matrícula y pensión para el año como apoyo al sistema de pagos.

4.9. REGISTRO ACADÉMICO

PROCEDIMIENTO DE CALIFICACIÓN EN LA INSTITUCIÓN

El año académico en la Institución se divide en 4 periodos o cortes, a su vez cada materia tiene 3 notas por periodo, las cuales se promedian y se calcula la nota final al culminar cada periodo académico; al final del año el promedio de las notas definitivas en cada periodo se promedia y se calcula la nota definitiva de la materia, a continuación se presenta un cuadro explicativo del procedimiento de calificación en la Institución.

1er periodo			
nota 1	nota 2	nota3	def1
2do periodo			
nota 1	nota 2	nota 3	def2
3er periodo			
nota 1	nota 2	nota 3	def3
4to periodo			
nota 1	nota 2	nota 3	def4
Definitiva			
promedio	definitivas	de	cada periodo

Tabla 4.1 Procedimiento de calificación en la institución

FUNCIONAMIENTO SEGÚN TIPO DE USUARIO

El registro de notas se lleva de la siguiente manera:

- Usuario Administrador: este tipo de usuario tiene acceso completo a toda la aplicación desde visualización de las notas, hasta registro y modificación de las mismas.
- Usuario Docente: este tipo de usuario tiene acceso limitado es decir que puede visualizar, registrar y modificar las notas en las materias que se le han asignado con sus respectivos cursos, pero no tiene ningún tipo de acceso a otras materias y cursos que no se le han asignado.

- Usuario Estudiante: este tipo de usuario solo puede tener acceso a la visualización de las notas que se han registrado previamente en su área de notas, por lo cual no puede hacer ningún tipo de modificación en ninguna materia.

Asignar calificación a alumno por materia

Seleccione el nivel:

Seleccione nivel

once

⌵

Seleccionar nivel

Nivel Seleccionado:

once

Seleccionar materia:

Seleccione materia

matematicas

⌵

Seleccionar materia

Materia Seleccionada:

matematicas

Seleccionar periodo de la materia:

Seleccione periodo

primer periodo

⌵

Seleccionar periodo

Periodo seleccionado:

primer periodo

Figura 4.16 Pantalla de ingreso al área de notas según tipo de usuario.

PROCEDIMIENTO DE ASIGNACIÓN DE MATERIAS EN EL SOFTWARE

Antes de poder hacer cualquier registro de notas es necesario crear las materias por cursos, se hace de esta forma ya que crear las materias por estudiante sería bastante dispendioso y poco práctico sabiendo que los estudiantes de un mismo curso ven las mismas materias, por lo cual los pasos para crear las materias son los siguientes:

- Dirigirse al cuadro de creación de plan de estudios
- Seleccionar curso y darle el nombre a la materia

The image shows a web interface for creating course materials. It is divided into four main sections:

- Top Left:** A section titled "Seleccione el nivel del plan de estudios:". It contains a dropdown menu labeled "Seleccione nivel" with "once" selected, and a blue button labeled "Seleccionar nivel".
- Top Right:** A section titled "Nivel Seleccionado:". It displays the selected level "once" and its description "once a".
- Bottom Left:** A section titled "Seleccione la materia a incluir al plan de estudios:". It contains a dropdown menu labeled "Seleccione materia" with "matematicas" selected, and a blue button labeled "Seleccionar materia".
- Bottom Right:** A section titled "Materias incluidas al plan de estudios, seleccione la que desea quitar (También eliminará los periodos asociados):". It contains a dropdown menu labeled "Seleccione materia" with "matematicas" selected, and a blue button labeled "Borrar materia del nivel".

Figura 4.17 Ventana de creación de materias por curso.

PROCEDIMIENTO PARA EL REGISTRO DE NOTAS EN EL SOFTWARE

Sabiendo que los únicos que pueden registrar y modificar materias son los usuarios docente y administrador, se debe ingresar al programa con alguno de estos usuarios y seleccionar el área de registro académico y se siguen los siguientes pasos:

- Se debe seleccionar el curso al cual se va a registrar notas
- Seleccionar la materia a registrar y el periodo académico
- Por último se ingresan cada una de las notas del periodo a los estudiantes y se le dan en el botón registrar y quedara todo actualizado.

Ingresar nota a alumnos:

Nombre de nota

Descripción

Alumno: juan perez

Guardar Calificación

Alumno: fernando vasquez

Guardar Calificación

Alumno: administrador

Guardar Calificación

Figura 4.18 Ventana de registro de notas.

4.10. ADMINISTRACION DE NÓMINA

Editar y exportar nomina

Seleccione el año y mes, se desplegará la tabla de nómina correspondiente.

Seleccione mes y año para verificar la nómina:

Año

2015

Mes

Enero

Buscar

Figura 4.19 Ventana de administración de nómina.

Se realizó este ítem de liquidación de nómina con el fin de generar una planilla de pagos completa, incluyendo en todo el personal de la empresa las normatividades vigentes para el trabajador como parafiscales, horas extras, descuentos y demás.

Datos de ingreso:

Seleccione empleado

Nombre: Juan Perez -- Identificación: 1026279934

Fecha de inicio

15 de abril

Fecha de Final

29 de abril

Días trabajados

60

días

Mes

Abril

Calcular liquidación

Figura 4.20 Visualización de personal de la empresa.

LIQUIDACIÓN DE NÓMINA

PRIMER APELLIDO	SEGUNDO APELLIDO	PRIMER NOMBRE	SEGUNDO NOMBRE	SALARIO	DIAS
Ac	Donec	ante. Maecenas	2500000	2500000	30
Vivamus	Enim	justo faucibus	2600000	2600000	30
Prueba	Prueba	Test	12222111111	12222111111	30
Prueba01	Prueba01	Andres	25000	25000	30
Fsdfsdf	Sdasds	Sdfsdf	11111111	11111111	30
Adasdasd	Sdfsdfs	Asdasdasd	2500122	2500122	30
Fsdfsdf	Adsfsdf	Asqwdqwd	123143243	123143243	30
Sdcscdcs	Asdffw	qwdwqwc	23423432	23423432	30
Yujyujyj	wqfewfrthy	yujyujyjtyrt	12423432	12423432	30

Tabla 4.2 parte 1: liquidación de nómina general.

DEVENGADOS		DEDUCIDOS				
SUB/TRANS	TOTAL DEVENGADO	SALUD	PENSIONES	TOTAL DEDUCIDO	NETO PAGAR	FIRMA
74000	2574000	100000	100000	200000	2374000	
74000	2674000	104000	104000	208000	2466000	
74000	12222185111	488884444,4	488884444,4	977768888,9	11244416222	
74000	99000	1000	1000	2000	97000	
74000	11185111	444444,44	444444,44	888888,88	10296222,12	
74000	2574122	100004,88	100004,88	200009,76	2374112,24	
74000	123217243	4925729,72	4925729,72	9851459,44	113365783,6	
74000	23497432	936937,28	936937,28	1873874,56	21623557,44	
74000	12497432	496937,28	496937,28	993874,56	11503557,44	
APORTE	APORTE TRA	TOTAL APORTE				
30	30	30				

Tabla 4.3 parte 2: liquidación de nómina general.

4.11. CONTROL DE ASISTENCIA

Para llevar a cabo la función de control de asistencia en el software, se debieron realizar otras labores extras como lo son:

Cartelización

Se realizó la toma de fotografías de todo el personal de la Institución para poder desarrollar la carnetización, esto se realizó por cursos iniciando por los cursos de más pequeños hasta los más grandes y por último se realizó la toma de fotos de los profesores y personal administrativo.



Figura 4.21 Imagen de la toma de fotos realizada al personal de la Institución.

GENERACIÓN DE CÓDIGOS DE BARRAS

Posterior a la toma de fotos del personal se debió realizar el diseño del carnet y la generación de código de barras, para esto se usó una fuente que genera estos códigos en Microsoft Word esta fuente fue descargada gratuitamente en la página de internet: <http://www.dafont.com/es/theme.php?cat=711>.

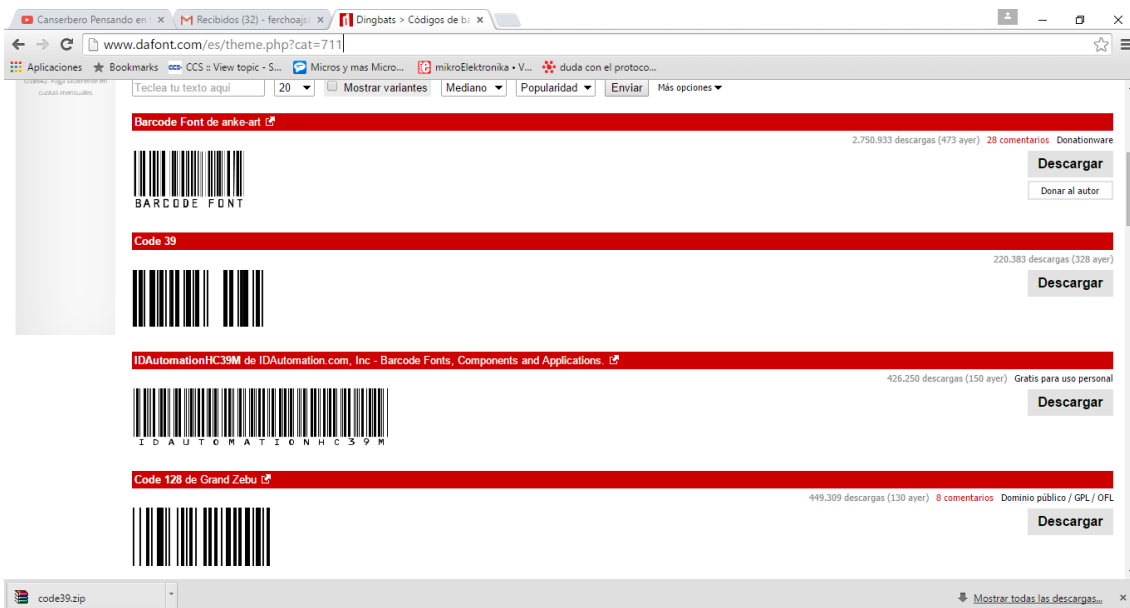


Figura 4.22 Imagen de la página de internet <http://www.dafont.com/es/theme.php?cat=711>.

Se descargó el tipo de código Code 39 ya que es uno de los más usados a nivel mundial por lo cual se garantiza que cualquier lector de códigos será compatible con este, una vez descargada la fuente para generar códigos de barras simplemente se abre el archivo y se instala ya con esto nos debe aparecer la fuente en la lista de Microsoft word.

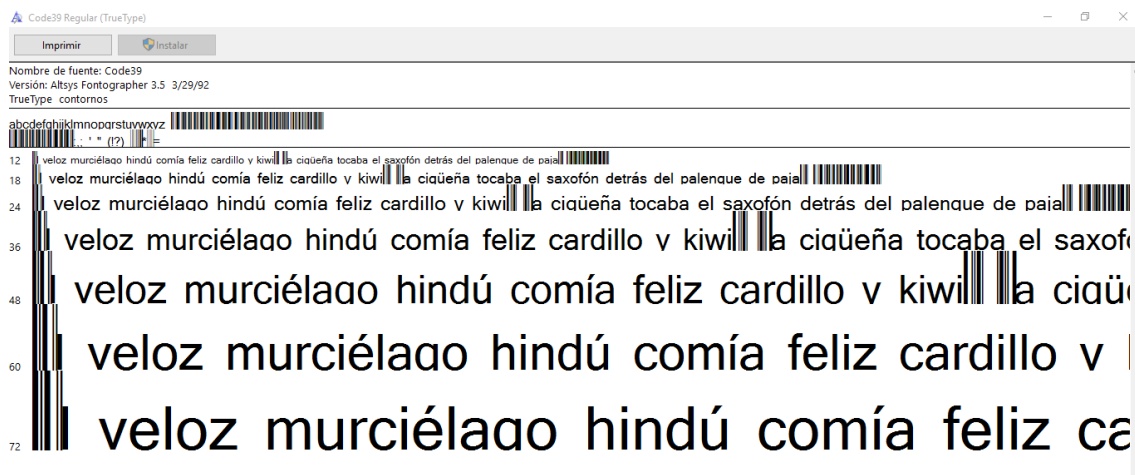


Figura 4.23 Imagen de la instalación de la fuente Code39.

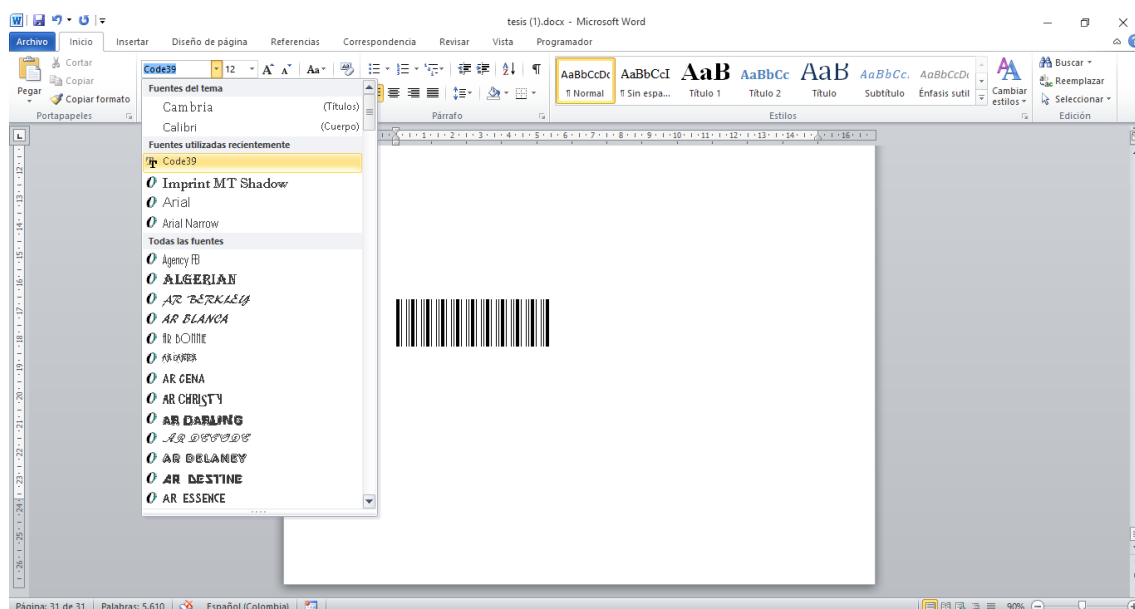


Figura 4.24 Fuente añadida exitosamente en la lista de Microsoft Word

Después de realizadas las labores de toma de fotos y generación de códigos de barras de los carnets se procedió a realizar la impresión de estos, para esto se contrató este servicio de impresión en carnets en pvc para dar mayor calidad a estos.

VERSIÓN FINAL DEL CARNET



Figura 4.25 Ejemplo del carnet terminado y listo para entregar al estudiante.

GENERACION BASE DE DATOS

Se generó la base de datos de todo el personal de la institución incluyendo los siguientes ítems: Nombre, apellido, documento de identificación, dirección, teléfono, fecha de nacimiento, tipo de personal (estudiante, docente o administrativo), tipo de sangre, etc.

Formulario de vista/modificación de usuarios

Criterio de Búsqueda

Resultados de la búsqueda

Nombres	Apellidos	Identificación	Fecha de nacimiento	Correo	Teléfono	Dirección	Acudiente	Nivel de acceso	Acciones
juan	perez	1026279934	0000-00-00	juan_payaso@hotmail.com	7651404	cll 67	juan	2	<input type="button" value="Modificar"/> <input type="button" value="Eliminar"/>

Figura 4.26. Base de datos generada del personal de la Institución.

IMPLEMENTACIÓN DEL CONTROL DE ASISTENCIA EN EL SOFTWARE

Teniendo los carnets y la base de datos listos, se procedió a realizar la programación en el software para el control de asistencia, por lo cual se creó la clase encargada de realizar esta función, básicamente esta clase verifica el ingreso del dato recibido por el lector de código de barras y teniendo en cuenta el día y la hora hace el registro de asistencia de cada integrante de la institución, así se puede llevar control de la asistencia de los estudiantes y el control de horario de los trabajadores

The interface is divided into two main columns. The left column contains three selection sections and one entry section. The right column displays the selected values for the first three sections.

Left Column:

- Seleccione el nivel:** A dropdown menu labeled "Seleccione nivel" with the value "once" selected, and a blue button labeled "Seleccionar nivel".
- Seleccionar materia:** A dropdown menu labeled "Seleccione materia" with the value "matematicas" selected, and a blue button labeled "Seleccionar materia".
- Seleccionar periodo de la materia:** A dropdown menu labeled "Seleccione periodo" with the value "primer periodo" selected, and a blue button labeled "Seleccionar periodo".
- Ingresar asistencia:** A section with a label "Lectura de código" next to an empty text input field, and a blue button labeled "Envío manual del codigo".

Right Column:

- Nivel Seleccionado:** Displays the value "once".
- Materia Seleccionada:** Displays the value "matematicas".
- Periodo seleccionado:** Displays the value "primer periodo".

Bottom Section:

Estos son las ultimas 30 asistencias de la materia y periodo:

Figura 4.27. Ventana del control de asistencia del software

CAPÍTULO 5

RESULTADOS

5.1. ASISTENCIA

SISTEMA DE MANEJO ANTERIOR AL SOFTWARE

La asistencia se controlaba y registraba de forma manual con planillas que contenían el listado de los alumnos organizado por cursos, el proceso tomaba entre 5 y 10 minutos por cada salón de clases, teniendo en cuenta que la Institución cuenta con 18 salones y un promedio por salón de 30 estudiantes.

MANEJO ACTUAL CON EL SOFTWARE

El registro de asistencia en el software se realiza apoyándose con la carnetización de la institución ya que el carnet cuenta con un código de barras que identifica al estudiante, este proceso se lleva a cabo con lectores de códigos de barras ubicados en la entrada de la institución, simplificando notablemente el tiempo de registro puesto que el colegio habilita el ingreso a los estudiantes a las 6am y cierra sus puertas a las 6:30am.

COMPARACIÓN DE MANEJO (ANTES / ACTUAL)

NUMERO DE SALONES	TIEMPO ANTERIOR(Minutos)	TIEMPO ACTUAL(Minutos)
1	10	1.6
18	180	30

TABLA 5.1. Comparación de manejo respecto al tiempo.

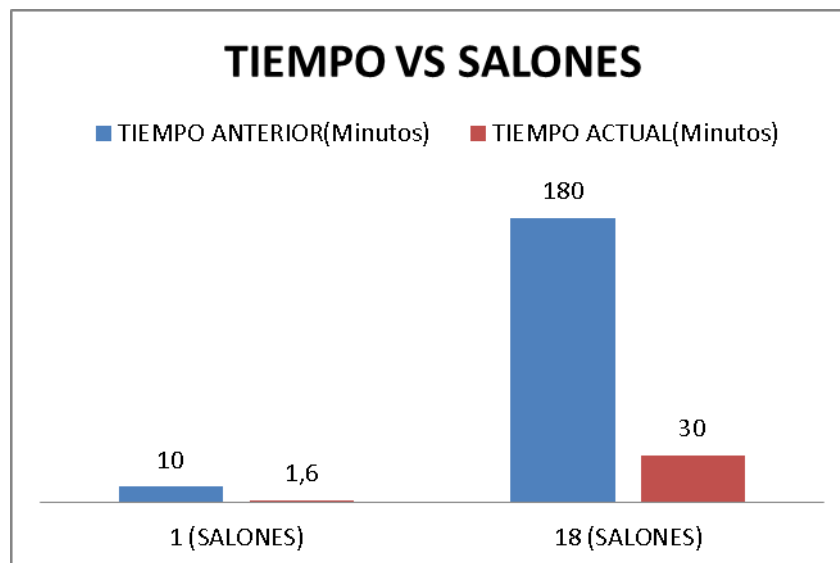


FIGURA 5.1. Grafica Tiempo Vs Salones.

VENTAJAS DEL SOFTWARE

Según los datos anteriores podemos observar y deducir que el software brinda una reducción de aproximadamente 150 minutos diarios en el tiempo que tarda el registro de asistencia, si se multiplica 150×22 que son el número de días en promedio que asisten los estudiantes a la institución mensualmente obtenemos un resultado de 3300 minutos equivalentes a 55 horas de trabajo al mes que se deben invertir para realizar este registro, el promedio del valor de cada hora tomando como referencia el salario mínimo legal mensual vigente para el 2015 es de 2.684 pesos, si lo multiplicamos por 55 horas nos da un valor de 147.620 pesos mensuales que sería el valor promedio que se economiza la institución por este proceso de registro de asistencia.

5.2. REGISTRO DE NOTAS

SISTEMA DE MANEJO ANTERIOR AL SOFTWARE

El colegio anteriormente contaba con un sistema de registro de notas sistematizado similar al que se está realizando actualmente, este proceso comprendía 3 etapas: en la primera cada profesor se debía dirigir a secretaria para que se le cargara el archivo de registro de notas en su memoria USB, en la segunda etapa los profesores disponen de 8 días para actualizar el registro y devolver las memorias y en la tercera etapa la secretaria debía realizar la adecuación o recopilación de información de cada memoria para generar los informes académicos y esta etapa la realizaba en un tiempo promedio de 2 días, por lo cual el proceso completo podía durar entre 10 a 12 días.

MANEJO ACTUAL CON EL SOFTWARE

Actualmente el proceso de registro de notas le brinda al profesor un usuario y contraseña el cual está habilitado para registrar las notas durante su periodo académico o a medida que vaya evaluando en sus áreas asignadas, por lo cual el proceso de recopilación se hace automáticamente, teniendo en cuenta que aún se mantienen los 8 días de plazo para el docente entregue su informe de registro de notas al finalizar el periodo académico.

COMPARACIÓN DE MANEJO (ANTES / ACTUAL)

PERIODO	TIEMPO ANTERIOR(DIAS)	TIEMPO ACTUAL(DIAS)
1	12	8
2	12	8
3	12	8
4	12	8
TOTAL	48	32

TABLA 5.2. Comparación de manejo respecto al tiempo.

VENTAJAS DEL SOFTWARE

Con el sistema actual la Institución se economiza un tiempo de 4 días por cada periodo, lo cual en el año sería un total de 16 días, tomando como referencia el salario mínimo legal mensual vigente en el 2015 es 21.478 pesos diarios por lo cual al año sería un total de 343.648 pesos de economía para Institución.

5.3. PAGOS

SISTEMA DE MANEJO ANTERIOR AL SOFTWARE

En el sistema de pagos anteriormente se generaba un recibo de caja en Excel con copia a la Institución, llevando manualmente una base de datos para el control del ingreso diario de dinero, esto implicaba diariamente revisar las copias del recibo uno a uno y registrarlo en su base de datos correspondiente (pensiones, matrículas y otros) tomando aproximadamente 1 hora.

MANEJO ACTUAL CON EL SOFTWARE

El manejo de pagos actualmente en la Institución por medio del software genera un recibo de caja y automáticamente registra el pago en la base de datos correspondiente (pensión, matrículas y otros), evitando así el registro manual al finalizar el día para el control de ingreso de dinero, dejando únicamente la comparación de los resultados ente el software y el dinero en caja.

COMPARACIÓN DE MANEJO (ANTES / ACTUAL)

DIA DE REGISTRO	TIEMPO ANTERIOR(Minutos)	TIEMPO ACTUAL(Minutos)
1	60	10
22	1320	220

TABLA 5.3. Comparación de manejo respecto al tiempo.

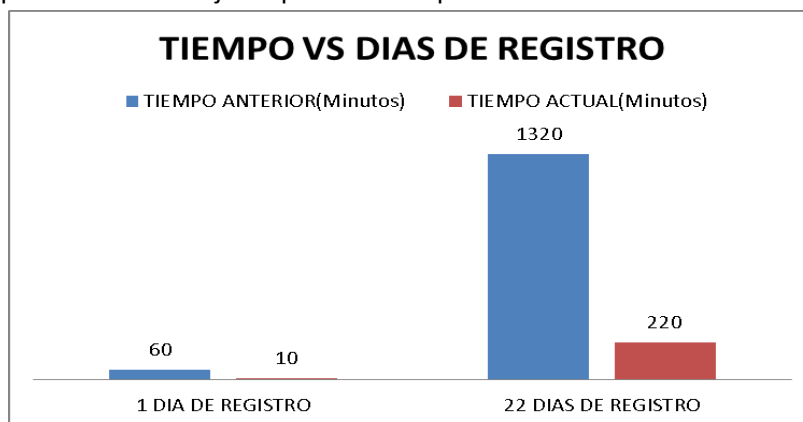


FIGURA 5.2. Grafica Tiempo Vs Días de Registro.

VENTAJAS DEL SOFTWARE

En base a los anteriores datos podemos decir que con el software actual la Institución se economiza en promedio en un mes laboral un total de 48.312 pesos teniendo en cuenta que se ahorra un total de 18 horas y tomando como referencia el salario mínimo legal mensual vigente en 2015 que es de 2684 pesos la hora.

POSIBLE MEJORAS EN EL SOFTWARE

- A futuro a este software se le impletmentara un sistema de control de errores, para no permitir usuarios con doble registro.

CAPÍTULO 6

CONCLUSIONES

CONCLUSIONES

- Java a pesar de ser un lenguaje robusto y frecuentemente usado presenta una debilidad de peso en aplicaciones web en específico en cuanto a su almacenamiento en servidores, ya que requiere que estos tengan una máquina virtual para su funcionamiento. Lo cual hace que los costos de alojamiento de información web en cantidad sean altamente elevados.
- La planeación adecuada de recolección, organización y utilización de la información para el software es fundamental debido a que el no tener en cuenta un mínimo detalle implicaría cambios drásticos en la estructura fundamental de la programación del software.
- La virtualización de las diferentes operaciones de la institución son de gran impacto positivo respecto a la optimización de tiempos en los procesos (registro de notas, pagos, liquidación de nómina y toma de asistencia) lo cual es demostrado en efectividad y beneficios económicos.
- Los padres de familia de la institución se ven altamente beneficiados respecto al interés de mantenerse informados acerca del rendimiento, comportamiento y educación de sus hijos.
- Teniendo en cuenta el módulo de asistencia se optimizo el tiempo en 150 minutos de trabajo mensualmente, se calcula un ahorro promedio de 147.620 pesos.
- En el módulo de notas se calcula que la Institución se economiza en promedio 343.648 pesos anuales además de hacer un proceso mucho más efectivo y amigable con los docentes, evitando el uso de las memorias USB para la recolección de información.
- En el módulo de pagos se calcula que la Institución se economiza en promedio 48.612 pesos mensuales además de entregar un valor exacto del flujo que se maneja diariamente en caja, omitiendo distracciones o errores que pueda tener el empleado que realiza el registro de esta información en las bases de datos de la institución.

CAPÍTULO 7

BIBLIOGRAFÍA

Referencias Bibliograficas

BIBLIOGRAPHY 6-7, c. p.-P. (s.f.).

(s.f.). administracion de servidores- Enzo Augusto Marchionni- Manuales USERS.

(s.f.). Administrador de servidores, por Marchionni, Enzo Augusto , pag 23-30.

como programar en java séptima edición- P.J Deitel pag 6-7. (s.f.).

(s.f.). como programar en java séptima edición- P.J Deitel pag 7-8.

(s.f.). como programar en java séptima edición- P.J Deitel pag 8.

(s.f.). como programar en java séptima edición- P.J Deitel pag 9-10.

(s.f.). como programar en java séptima edición- P.J Deitel pag. 10.

(s.f.). como programar en java séptima edición.

(s.f.). La Biblia de MySQL Ian Giffillan editorial Anaya.

(s.f.). Manual de programación php- Marcelo Maraboli rosselott.

(s.f.). CONTABILIDAD GENERAL DE GUILLERMO GONZÁLEZ SAAVEDRA, 2003, paginas 55,56,57.

(s.f.). Contabilidad: principios y aplicaciones By Horace R Brock, Ch. E Palmer capitulo 15 pag 249.

(s.f.). El gran libro de HTML5, CSS3 y Javascript - Juan Diego Gauchat editorial marcombo.

(s.f.). El gran libro html5-css3 y javascript Juan Diego Gauchat Editorial marcombo.

Referencias Web

(s.f.). <http://desarrollandowebsdinamicas.blogspot.com.co/2013/03/que-es-laravel.html>.

(s.f.). <http://jorgesanchez.net/programacion/manuales/Java.pdf>.

(s.f.). http://librosweb.es/libro/css/capitulo_1.html.

(s.f.). <http://tc2proyectodegradoingdesistemas.blogspot.com.co/2012/05/introduccion.html>.

(s.f.). <http://www.gerencie.com/nomina.html>.

(s.f.). <http://www.ibrugor.com/blog/apache-http-server-que-es-como-funciona-y-para-que-sirve/>.

(s.f.). https://kesquivel.files.wordpress.com/2011/03/clase_no1_quc3a9-es-un-sistema-de-base-de-datos.pdf.