

UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



**DESARROLLO DE UN SISTEMA DE
INFORMACIÓN PARA GENERAR CERTIFICADO
DE ESTUDIOS DEL NIVEL BÁSICO REGULAR DE
LA UGEL ANDAHUAYLAS**

Presentador por:

BACH. MILAGROS CANDI PÉREZ TORRES

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO DE SISTEMAS**

ANDAHUAYLAS – APURÍMAC – PERÚ

2017

UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS
FACULTAD DE INGENIERÍA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Presentador por:

BACH. MILAGROS CANDI PÉREZ TORRES

**DESARROLLO DE UN SISTEMA DE
INFORMACIÓN PARA GENERAR CERTIFICADO
DE ESTUDIOS DEL NIVEL BÁSICO REGULAR DE
LA UGEL ANDAHUAYLAS**

Asesor:

ING. EDWING ALCIDES MAQUERA FLORES

ANDAHUAYLAS – APURÍMAC – PERÚ

2017

DEDICATORIA

A mis **padres**: por concederme la oportunidad de estudiar, por su esfuerzo y amor incondicional a lo largo de mi vida.

A mi familia, parientes y amigos: por sus palabras de aliento y por estar siempre presentes en momentos duros de la vida.

A mis profesores: por sus enseñanzas, consejos y ejemplo que llevo presente en el transcurso de mi vida profesional y personal.

AGRADECIMIENTO

A Dios Todopoderoso y mi voz de gratitud a la Universidad Nacional José María Arguedas, institución en la que me forme por haberme permitido culminar esta investigación.

Mis más profundos sentimientos de fraternal gratitud a mis padres y hermanos, toda mi familia quienes sostuvieron mi educación tanto económica como moralmente de forma sincera en cada momento de mi vida para lograr una meta más.

Agradezco a los profesores que, dirigidos por su alto espíritu de la docencia, están sembrando el futuro de la juventud para el perfeccionamiento y culminación a sus metas.

A Manuel Cuva Paredez, profesional de la UGEL Andahuaylas, por su apoyo, comprensión, amabilidad, facilidad y voluntad brindada para el desarrollo y normal ejecución de este proyecto.

ÍNDICE GENERAL

DEDICATORIA	iii
AGRADECIMIENTO	iv
ÍNDICE GENERAL	v
LISTA DE IMAGENES	vii
LISTA DE TABLAS	ix
RESUMEN Y PALABRAS CLAVES	xii
ABSTRACT.....	xiii
CAPÍTULO I	14
INTRODUCCIÓN	15
1.1. DATOS GENERALES	16
CAPÍTULO II.....	17
PLANTEAMIENTO DEL PROBLEMA	17
2.1. Realidad problemática	17
2.2. Formulación del problema	19
2.2.1. Problema General.....	19
2.2.2. Problemas Específicos	19
2.3. Objetivos.....	19
2.3.1. Objetivo General.....	19
2.3.2. Objetivos Específicos.....	19
2.4. Justificación.....	20
2.5. Viabilidad de la investigación.....	21
2.5.1. Viabilidad Técnica	21
2.5.2. Viabilidad Económica.....	22
2.5.3. Viabilidad Operativa.....	22
2.6. Limitación del estudio.....	22
CAPÍTULO III	23
3.1. Antecedentes de la investigación.....	23
3.1.1. Antecedentes a Nivel Internacional	23
3.1.2. Antecedentes a Nivel Nacional.....	23
3.2. Base Teórica	25
3.2.1. Sistema de Información	25
3.2.2. Certificado de Estudios	27

3.2.3. Acta de Notas	27
3.2.4. Procedimiento para Visación	27
3.2.5. Visación de Certificado de Estudios	30
3.2.6. Sistema Web	30
3.2.7. Aplicaciones Web	30
3.2.8. Base de datos.....	31
3.2.9. Sistemas gestores de bases de datos.....	32
3.2.10. Servidor.....	36
3.2.11. MySQL	37
3.2.12. LENGUAJES DE PROGRAMACIÓN	39
3.2.13. XAMPP.....	41
3.2.14. Metodología eXtreme Programming (XP).....	41
3.2.15. Framework	48
CAPÍTULO IV	52
PROPUESTA DE SOLUCIÓN.....	52
4.1. Tecnologías para el entorno de desarrollo.....	52
4.1.1. Framework Laravel v5.3.....	52
4.1.2. Paquete XAMPP v5.6	56
4.1.3. Herramienta MySQL Workbench v6.3.....	57
4.1.4. Software de control de versiones GIT	59
4.1.5. Editor de Texto ATOM.....	60
4.2. Tecnologías de apoyo	61
4.2.1. Servicio de alojamiento de archivos Google Drive	61
4.2.2. Suit de ofimática Google Docs	62
4.3. DESARROLLO DEL SISTEMA DE INFORMACION VISAssoft	63
4.3.1. Planeación.....	63
4.3.2. Diseño	68
4.3.3. Codificación.....	86
4.3.4. Pruebas	91
CAPÍTULO V.....	98
EVALUACION DE LA SOLUCIÓN	98
5.1. APLICACIÓN DE TÉCNICAS DE INVESTIGACIÓN	98
5.2. CONSTATACIÓN DE OBJETIVOS.....	98
CONCLUSIONES.....	109
RECOMENDACIONES	110
REFERENCIAS BIBLIOGRAFICAS	110
ANEXOS	1153

LISTA DE IMAGENES

Imagen 1 Representación del Flujo del procedimiento Visación	28
Imagen 2: Proceso de Verificación de las Notas del Certificado de Estudios	29
Imagen 3 : Valores XP	42
Imagen 4: Proceso XP	43
Imagen 5: Planificación XP	44
Imagen 6: Diseño en Xp	45
Imagen 7: Codificación en XP	46
Imagen 8: Pruebas en XP	47
Imagen 9 : Estructura VisaSoft en laravel.	53
Imagen 10: TurnoController	54
Imagen 11: Eloquent en el modelo Acta	55
Imagen 12: Sintaxis Blade	56
Imagen 13: Apache del paquete Xampp	57
Imagen 14: MariaDB	57
Imagen 15: Warning del Workbench	58
Imagen 16: Workbench tablas y BD del Sistema VisaSoft	59
Imagen 17: GIT	60
Imagen 18: Clase ActaController.php en Atom	61
Imagen 19: captura Google Drive	62
Imagen 20: Tablas de la BD del Sistema VisaSoft	63
Imagen 21: Modelo de Casos de Uso de Negocio Sistema VisaSoft	69
Imagen 22: Modelo de Casos de Uso del sistema VisaSoft	70
Imagen 23: Modelo de CU de Gestión de certificados VisaSoft	73
Imagen 23: Login de VisaSoft	99
Imagen 24: Tablas de la BD de VISAssoft	100

Imagen 25: Crear Nueva Acta.....	101
Imagen 26: Detalle Acta.....	101
Imagen 27: Reportes Actas	102
Imagen 22: Tablas del Login de VisaSof.....	117

LISTA DE TABLAS

Tabla 1: Padrón Instituciones Educativas	21
Tabla 2: Historia de Usuario Registrar Datos de Acta de Notas.....	64
Tabla 3: Historia de Usuario Registrar Datos de Acta de Subsanación.....	65
Tabla 4: Historia de Usuario Buscar Acta de Subsanación.....	65
Tabla 5: Historia de Usuario Buscar Acta de Notas	66
Tabla 6: Historia de Usuario Guardar Datos de UGEL	66
Tabla 7: Historia de Usuario Guardar Datos de la Institución Educativa	67
Tabla 8 : Estimación de esfuerzos por Historia de Usuario.....	68
Tabla 9: Descripción de los Actores de Negocio	70
Tabla 10: Requerimientos no funcionales.....	71
Tabla 11: Requerimientos gestión de información	72
Tabla 12: Matriz de Trazabilidad de RF vs CU	74
Tabla 13: Especificación de CU-001	75
Tabla 14: Especificación de CU-002	76
Tabla 15: Especificación de CU-003	76
Tabla 16: Especificación de CU-004	76
Tabla 17: Especificación de CU-005	77
Tabla 18: Especificación de CU-006	77
Tabla 19: Registrar Acta	79
Tabla 20: Editar Usuario	79
Tabla 21: Eliminar Usuario	80
Tabla 22: Buscar Estudiante	80
Tabla 23: Registrar Estudiante.....	81

Tabla 25: Eliminar Acta	82
Tabla 26: Buscar Acta	83
Tabla 27: Acta Estudiante.....	83
Tabla 28: Editar Estudiante	84
Tabla 29: Eliminar Estudiante	84
Tabla 30: Buscar Estudiante	85
Tabla 31: Reporte Actas	85
Tabla 32 : Tarea Ingeniería Diseño Básico de la Base de Datos.....	86
Tabla 33: Diseño de la interfaz para las modalidades	87
Tabla 34 Tarea Ingeniería Diseño de la interfaz para los niveles	87
Tabla 35: Tarea Ingeniería Diseño de la interfaz para los grados	87
Tabla 36 Diseño de la interfaz para registrar Acta de Notas.....	88
Tabla 37 : Tarea Ingeniería Diseño de la interfaz para los grados	88
Tabla 38: Tarea Ingeniería Diseño de la interfaz para las asignaturas	88
Tabla 39 : Tarea Ingeniería Diseño de la interfaz para los turnos	89
Tabla 40: Tarea Ingeniería Diseño de la interfaz para registrar Acta de Subsanación	89
Tabla 41 : Tarea Ingeniería Diseño de la interfaz para buscar Acta de Subsanación	89
Tabla 42: Tarea Ingeniería Diseño de la interfaz para buscar Acta de Notas.....	90
Tabla 43: Tarea Ingeniería Diseño de la interfaz para registrar datos de UGEL .	90
Tabla 44: Tarea Ingeniería Diseño de la interfaz para registrar datos de la Institución Educativa.....	91
Tabla 45: Prueba Funcional Registro de Datos de Acta Notas	92
Tabla 46: Prueba Funcional Registro de Acta de Subsanación	93
Tabla 47: Prueba Funcional Búsqueda de Acta de Notas.....	94
Tabla 48: Prueba Funcional Búsqueda de Acta de Sustentación	95
Tabla 49: Prueba Funcional Verificar Registro de datos de UGEL.....	96

Tabla 50: Prueba Funcional Verificar Registro de Institución Educativa.....	97
Tabla 51: Presentación de Solicitud del Certificado de Estudios – PRE.PRUEBA.....	103
Tabla 52: Inicio de Búsqueda de Notas en las Actas – PRE.PRUEBA.....	103
Tabla 53: Finaliza la búsqueda de las notas en las actas –PRE.PRUEBA	104
Tabla 54: Presentación de Solicitud del Certificado de Estudios – POST.PRUEBA	104
Tabla 55: Finaliza la búsqueda de las notas en las actas –POST.PRUEBA.....	105
Tabla 56: Tiempo de verificación –PRE.PRUEBA.....	106
Tabla 57: Tiempo de verificación –POST.PRUEBA	107
Tabla 58: Resultado Pre-Prueba/Post-Prueba	108
Tabla 59: Resultado Pre-Prueba/Post-Prueba	108

RESUMEN Y PALABRAS CLAVES

Desarrollo de un Sistema de Información para Generar Certificado de Estudios del Nivel Básico Regular de la UGEL Andahuaylas. La presente investigación se basó en el desarrollo de un sistema de información para la reducción del tiempo en el proceso de búsqueda de actas, para el procedimiento del visado de los certificados de estudios, en la unidad de "VISACION" en la UGEL Andahuaylas. Específicamente en la búsqueda de las actas correspondientes del interesado, las cuales contienen sus respectivas notas, siendo de vital interés para realizar la constatación del certificado de estudios, además por ser este un proceso el cual demanda un considerable lapso de tiempo. Solución al problema: con la finalidad de lograr el mejor desempeño y agilizar los procesos, garantizando un mejor manejo, custodia a acceso a la información de las actas.

En el desarrollo del proyecto se empleó la metodología XP. Se emplearon una serie de tecnologías y herramientas modernas y robustas como el lenguaje de programación PHP, el sistema manejador de base de datos mariaDB, MySql Workbench, el Framework Laravel, editor Atom y el servidor Web Apache 2.4.23 con el fin de lograr mayor calidad, rendimiento en el testeado, fiabilidad y desenvolvimiento del software. En cuanto a los instrumentos de recolección de requerimientos del usuario se ciñeron a la metodología XP. Y para la recolección y validación de los datos en el cumplimiento de los objetivos fueron; la observación directa, las entrevistas no estructuradas, Pre-prueba/post-prueba

De esta manera se pudo concluir que con la implantación del nuevo sistema se genera un mejor desempeño de las labores en el procedimiento de visación de los certificados de estudios en cuanto a la realización del proceso de búsqueda en las actas.

Palabras claves: Sistema de Información, Proceso de verificación, certificado de estudios, XP.

ABSTRACT

SUMMARY AND KEY WORDS

Development of an Information System to Generate Certificate of Studies of the Basic Level of the UGEL Andahuaylas. The present investigation was based on the development of an information system for the reduction of the time in the process of searching for records, for the procedure of the visa of the certificates of studies, in the area of "VISION" in the UGEL Andahuaylas. Specifically in the search of the corresponding records of the interested party, which contain their respective notes, being of vital interest to carry out the verification of the certificate of studies, also because this is a process which demands a considerable time span. Solution to the problem: in order to achieve the best performance and streamline processes, ensuring better management, custody access and information of the minutes.

In the development of the project the XP methodology was used. A number of modern and robust tools and technologies such as the PHP programming language, the database management system mariaDB, MySql Workbench, the laravel Framework, the atom editor and the Apache 2.4.23 Web server were used in order to achieve greater quality, performance, testing and reliability in the development of the software. As for the instruments of collection of the requirements of the user were adhered to the methodology XP. And for the collection and validation of the data in the fulfillment of the objectives were; direct observation, unstructured interviews, pre-test / post-test.

In this way it was possible to conclude that with the implementation of the new system, a better performance of the work is generated in the procedure for the visas of the study certificates in terms of carrying out the search process in the minutes.

Key words: Information System, Verification Process, Certificate of Studies, XP.

CAPÍTULO I

INTRODUCCIÓN

En el año 1973 el estado peruano empezó a preocuparse por el informático y crea la Comisión Nacional de Informática, organismo dependiente del Instituto Nacional de Planificación (INP). En 1986 recibe su primera revisión y crea la Secretaría Nacional de Informática, quien formula el primer proyecto nacional de informática; luego en 1990 es absorbido por el Instituto Nacional de Estadística (INE), por lo que se transforma en INEI desde el cual se logra, aún con visión asistencial, normar y coordinar el desarrollo informático nacional. Posteriormente en el año 2005 la Oficina Nacional de Gobierno Electrónico añade a su nombre el término “Informática” y se convierte en ONGEI, creado sobre la base de la tendencia mundial de la sociedad de la información y el gobierno electrónico (E-Government), ya en el 2007 decidió ratificar la validez y seguimiento de la agenda planteada por la Comisión Multisectorial para el Desarrollo de la Sociedad de la Información (CODESI). El objetivo general y estratégico de la ONGEI es desarrollar el Sistema Nacional de Informática para consolidarlo de manera que pueda cumplir con su misión, que es la de crear un gobierno más cercano, abierto y eficiente empleando la tecnología de la información.

No obstante, todos los esfuerzos realizados, en términos de TIC, las instituciones públicas del país siguen actuando y resolviendo sus problemas cotidianos como pueden y cuando pueden, la mayoría se encuentra al margen de los avances tecnológicos y metodológicos que se anuncian, no están integrados o involucrados en los planes de desarrollo propuestos y menos aún tienen con quien llevarlos a cabo (recursos humanos) o con que (presupuesto) llevar a cabo las directivas verticales que les imponen los entes rectores. Adicionalmente, cuando de manera independiente y autónoma cada institución decide optar por tal o cual camino para resolver sus problemas de información, comunicaciones y/o procesos utiliza herramientas y metodologías que mejor le parece para abordar dichos problemas contrata y consume su escaso presupuesto en soluciones con software propietario de diseño tosco y para problemas generales. Además, no es menor la desvinculación y bajo interés por parte de los jefes institucionales y demás personal administrativo a ello se suma la ubicación marginal, desde el punto de vista organizacional, en la que se encuentran ubicadas las áreas de TI en los organigramas de la administración pública. Muchas áreas de TIC aún continúan

siendo secciones dependientes de Logística, Administración o Planificación y su enfoque es básicamente asistencial o de soporte técnico. Gran parte de esta responsabilidad es compartida con los jefes o encargado de TIC quienes no logran plantear con claridad un diagnóstico situacional que les permita sustentar los problemas organizacionales y de gestión que afronta su institución dejando abierta la posibilidad a que las áreas administrativas que conforman la institución.

La Unidad de Gestión Educativa Local (UGEL) Andahuaylas no es ajena a ésta situación, la Unidad de Visación sufre de latencia en la entrega de Certificados de estudios, y cada año se produce un cuello de botella por la elevada cantidad de solicitudes de éste documento, para la entrega de un certificado de estudios se debe pasar por un proceso engorroso y que muchas veces resulta confuso , provocando aún más la demora en la entrega de este documento, éste proceso puede ser optimizado y mejorado mediante un sistema de información que salvaguarde la información necesaria y permita un acceso rápido, seguro y confiable a los datos.

El presente trabajo de investigación consta de cuatro módulos:

En el capítulo 1: Se hace una descripción de la estructura de la investigación, sobre el desarrollo del sistema de información para generar certificado de estudios del nivel básico regular en la UGEL Andahuaylas. Basándose en ello se plantea el problema, los objetivos y la justificación de la investigación.

En el capítulo 2: Se analizó el problema de la investigación describiendo la realidad problemática, el problema general, planteamiento de los objetivos, la justificación, viabilidad y las limitaciones de la investigación.

En el capítulo 3: Se desarrolló el marco teórico, realizando una recopilación de los antecedentes de la investigación y leyes;

En el capítulo 4: Se describen la metodología, tecnologías empleadas para el desarrollo y el logro de los objetivos todo dentro de las especificaciones y recomendaciones ofrecidas por la metodología XP, las historias de usuario.

En el capítulo 5: Se presenta la evaluación de la solución describen el resultado obtenido, y constatación del logro de los objetivos.

También se plantean las conclusiones y recomendaciones respectivas referidas a la implementación del sistema y con el logro de los objetivos planteados en la investigación, y se presentan las referencias bibliográficas utilizadas en la investigación.

1.1. DATOS GENERALES

1.1.1. Título

DESARROLLO DE UN SISTEMA DE INFORMACIÓN PARA
GENERAR CERTIFICADO DE ESTUDIOS DEL NIVEL BÁSICO
REGULAR DE LA UGEL ANDAHUAYLAS

1.1.2. Autor del proyecto

Nombre y Apellido : Bach. Ing. Milagros Candi Pérez Torres.
Escuela Profesional : Ingeniería de sistemas.
E-mail : milash6@gmail.com

1.1.3. Asesor del Proyecto

Nombre y Apellido : ING. EDWING ALCIDES MAQUERA
FLORES
Departamento Académico : Ingeniería y Tecnología de Informática -
DAITI
Categoría de docente : Auxiliar.
Modalidad : Nombrado
E-mail : edwingmaquera@hotmail.com

1.1.4. Línea de investigación

La línea de investigación en el cual se basa el desarrollo del proyecto
es la INGENIERÍA DE SOFTWARE.

1.1.5. Área priorizada del proyecto

Desarrollo de sistema de Gestión (COD. 0403 0502)

1.1.6. Institución y lugar de ejecución del proyecto

UGEL, provincia de Andahuaylas región de Apurímac.

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1. Realidad problemática

La Unidad de Gestión Educativa Local de Andahuaylas es una instancia de ejecución descentralizada que depende de la Dirección Regional de Educación de Apurímac (DRE-Apurímac), que coordina labores y funciones con las entidades educativas asignadas.

Según el informe de (Paredes, 2015) en la UGEL de Andahuaylas se visan Certificado de estudios en un promedio rutinario de 450 Certificados de Estudios al mes. Pero en el periodo de época alta es donde se registra mayor cantidad de solicitudes para este proceso del visado de Certificado de Estudios; en los cuatro primeros meses de cada año, es donde sucede que se llega a registrar en un promedio 1200 a 2000 Visaciones de Certificado de Estudios mensuales.

Los usuarios que solicitan este servicio son todos los que estudiaron en las instituciones educativas pertenecientes a la jurisdicción de la UGEL de Andahuaylas, de los veinte distritos que se cuenta en los siguientes niveles. Inicial; Primaria Menor, CEBA Primaria Secundaria Menor, SEBA Secundaria, PRONOI, CEPED, Tecnológico, Especial, Pedagógico. CEDPRO.

Actualmente el Ministerio de Educación posee un Sistema académico para la administración de las notas denominado SIAGE que soporta las notas de todo el sistema de educación básica regular nivel (inicial, primario y secundario). Este software fue implementado el año 2012 por lo que tiene data a partir de este año los años anteriores no fueron almacenados en la base de datos solo existe en el archivo físico de cada unidad de Gestión educativa a nivel nacional, este Sistema es administrado por la unidad de informática del Ministerio de Educación, todos los miembros de la comunidad educativa peruana solo tienen accesos limitados a nivel de usuario es decir no se tiene permisos para manipular esta base de datos.

El párrafo anterior indica que se tiene información desde el año 2012 de la educación básica regular y el responsable del área de visación necesita tener información digitalizada mínimamente desde el año 2002 en adelante, esta información no se encuentra en el SIAGE, como tampoco ha

considerado este software las notas de los institutos superiores tecnológicos y privados los CEPRO y los PRONOI y CEBA'S.

En la Unidad de Visación es donde el personal responsable, que lleva a cabo el proceso de constatación de las notas de los estudiantes que pertenecen ámbito de la UGEL de Andahuaylas, que corresponde de los 20 Distritos de la provincia de Andahuaylas.

La Unidad de Visación cumple las siguientes funciones:

- Visar los Certificados de Estudios que se expiden en las Instituciones Educativas del ámbito jurisdiccional de la UGEL, de acuerdo a las actas que obran en sus archivos.
- Certificar y autenticar copias de documentos originales relacionados con actas y Certificados que obran en los archivos.
- Clasificar, archivar y custodiar las Actas, Certificados y otros documentos que se tramiten.

El Certificado de Estudios se expide a petición del interesado, quien habitualmente lo utiliza para el reconocimiento de su nivel educativo y para motivos propios (como requisito).

Para este proceso de Visación el plazo de respuesta y entrega de respuesta de dicho solicitud para su Visado del Certificado de Estudios, tiene una duración de un promedio de 3 a 7 días hábiles (como lo establece el manual de organización y funciones), ya que el personal a cargo recibe el Certificado de Estudios que se presenta en dicha solicitud, a partir de ello se procede a la constatación de las notas; con las notas registradas en las actas de dicha Unidad, nota por nota de cada año dando fe de la validez del contenido de las notas de dicho Certificado de Estudios, que pertenece a su respectiva nota original.

El visado del certificado de estudios es un documento que da fe de la legalidad de su contenido de dicho documento. Este documento es de vital importancia para actividades propias que el usuario requiere, pero existe una demora de respuesta, lo cual genera una disconformidad al usuario, provocando atraso en las otras actividades que realizan en el área. Careciendo de una rápida atención, porque el personal a cargo tiene que acudir a sus archivos de acta de notas y contrastar las notas por años, tomándole mucho tiempo y resultándole tedioso este proceso ya que realiza

esta actividad para los solicitantes, pertenecientes de los 20 distritos de la provincia en sus diferentes niveles.

2.2. Formulación del problema

2.2.1. Problema General

¿De qué modo podemos disminuir la demora en el proceso de verificación de notas para generar certificado de estudios del nivel básico regular en la UGEL Andahuaylas?

2.2.2. Problemas Específicos

-¿Cuál es la metodología y los procesos más adecuados para el desarrollo del sistema de Información para Generar Certificado de Estudios del Nivel Básico Regular de la UGEL Andahuaylas?

-¿De qué manera se puede reducir el proceso de verificación de notas del certificado de estudios del nivel básico regular en la UGEL Andahuaylas?

-¿En qué medida se redujo la demora en el proceso de verificación de notas del certificado de estudios del nivel básico regular en la UGEL Andahuaylas?

2.3. Objetivos

2.3.1. Objetivo General

Desarrollar un Sistema de Información para Generar Certificado de Estudios del Nivel Básico Regular de la UGEL Andahuaylas con metodología XP y framework Laravel.

2.3.2. Objetivos Específicos

-Determinar la metodología y sus procesos en el desarrollo del Sistema de Información para Generar Certificado de Estudios del Nivel Básico Regular de la UGEL Andahuaylas

-Implementar el Sistema de Información para reducir el proceso de verificación de notas del certificado de estudios del nivel básico regular en la UGEL Andahuaylas.

-Reducir el tiempo del proceso de verificación de notas para la visación del certificado de estudios del nivel básico regular en la UGEL Andahuaylas.

2.4. Justificación

Esta investigación se realiza, porque existe la necesidad de reducir el tiempo del proceso de verificación de notas, en la Unidad de Visación de la UGEL Andahuaylas. Con el uso del Sistema de Información en la Unidad de Visación, el personal responsable de la Unidad de Visación podrá realizar la constatación de las notas en tiempo real y al momento.

El Sistema de Información beneficiará a los usuarios de las 665 instituciones de Educación Básica Regular del ámbito de la UGEL de Andahuaylas, ya que la respuesta de su solicitud será en menor tiempo, de esa manera continuar con sus actividades.

En la unidad de Visación se atiende las solicitudes de los usuarios, de los 20 distritos de la provincia de Andahuaylas. Los usuarios en el desarrollo de sus actividades académicas necesitan de este requisito, de tener su Certificado de Estudios Visado, pero tienen que esperar entre 3 y 7 días, siendo perjudicial para el flujo de sus actividades. Con el Sistema de información, la búsqueda de las actas será inmediata y el proceso de Visación no excederá, en el peor de los casos, a un día hábil.

El actas de notas es un documento de vital importancia para las Instituciones Educativas y la UGEL Andahuaylas, porque hace parte del historial documentario de los estudiantes de las diferentes Instituciones educativas del ámbito provincial, regional y nacional, el cual da fe de la promoción y certificación de cada estudiante. A partir de este documento se generan los Certificados de Estudios, lo cual es para los intereses personales de cada estudiante/ egresado. El Sistema de Información salvaguardará los registros y disminuirá la manipulación de las actas físicas, evitando así el deterioro de estos documentos.

Además, Esta investigación contribuye al conocimiento existente sobre el uso de la metodología programación extrema (XP). Como instrumento de Análisis, desarrollo y evaluación de sistema de información.

Tabla 1: Padrón Instituciones Educativas

	N° I.E	Nivel	Matriculados
	266	Inicial	7219
	100	Pronoi	660
	205	Primaria	17475
	72	Secundaria	14684
	3	CEBA Primaria	156
	4	CEBA Secundaria	479
	3	Especial	51
	2	CEPED	95
	1	Tecnológico	230
	1	Pedagógico	331
	8	CETPRO	1136
Total	665		42516

Fuente: Padrón Instituciones UGEL 2016-Andahuaylas

2.5. Viabilidad de la investigación

2.5.1. Viabilidad Técnica

Se cuenta con los conocimientos necesarios para el desarrollo de la investigación, así como los conocimientos básicos de las tecnologías a usarse. Para resolver problemas o inconvenientes se recurrirá a la documentación de las tecnologías, a foros y grupos de discusión. El acceso a internet está garantizado, porque brindará acceso a todas las tecnologías que de ser obtenidas de otra manera no tendrían el mismo grado de seguridad y confiabilidad.

El personal de la institución estará involucrado en todas las fases del desarrollo del software, por el área interesada y usuaria del sistema. Además, por ser poseedores de todo el conocimiento técnico y la experiencia suficiente en el procedimiento de visado de Certificado de Notas, manifestaron su total conformidad y disponibilidad para resolver las dudas e inconvenientes durante el desenvolvimiento del proyecto. Ciertos materiales serán proporcionados por la UGEL-Andahuaylas como un Scanner de documentos, una computadora personal, las

actas de notas y todos los documentos involucrados en el procedimiento de Visación.

2.5.2. Viabilidad Económica

Se cuenta con el presupuesto adecuado para desarrollar el software y para todos los gastos pertinentes a este, como desplazamiento local, intercomunicación, trámites documentarios y tecnologías de almacenamiento. Los equipos para el desarrollo del software son de fácil obtención y los que no lo fuesen serán gestionados con ayuda de la UGEL-Andahuaylas, de igual manera cualquier inconveniente en el alcance del presupuesto será resuelto por la UGEL-Andahuaylas.

El framework Laravel, el gestor de dependencias Composer, el lenguaje PHP5, el paquete XAMPP, el servidor Artisan y los servicios en la nube son totalmente gratuitos, para su obtención solo es requerido el acceso a internet.

2.5.3. Viabilidad Operativa

El director de la UGEL-Andahuaylas reconoce el proyecto y brinda todas las facilidades para el desarrollo de este, se cuenta con acceso a las oficinas correspondientes, el apoyo por parte del personal que labora en dicha institución, así como acceso a los documentos, registros y actas. Todo inconveniente será resuelto con el representante del área quién ya manifestó el interés porque se desarrolle el sistema. No se esperan inconvenientes operativos durante el desarrollo del proyecto.

2.6. Limitación del estudio

En el desarrollo de la investigación se presentaron las siguientes limitaciones:

El sistema estará desarrollado con tecnología web laravel, con diseño responsive. Mediante bootstrap framewok.

El sistema funcionará en una única máquina no estará disponible en internet y no será accesible por ninguna otra vía que no sea a través de dicha máquina.

Los escaneados de actas deberán estar en formato pdf.

El sistema solo permitirá el acceso a personal autenticado a través de una pantalla login.

CAPÍTULO III

MARCO TEORICO

3.1. Antecedentes de la investigación

Para fortalecer la investigación fue necesaria revisar estudios relacionados con el tema, a fin de buscar un aporte al mismo, a continuación, se mencionan los más relevantes:

3.1.1. Antecedentes a Nivel Internacional

Al realizar las investigaciones de Diseño Implementación de sistema para generar certificado de estudios, se encontró a nivel internacional el siguiente trabajo.

En (Chile, 2002), se implementa este aplicativo que lleva como nombre “Portal de Certificados en Línea”, este aplicativo en línea permite obtener este documento a quienes cursaron sus estudios durante o después del 2002, del ministerio de educación CHILE.

Relación: esta investigación emplea tecnología Web como alternativa para obtener el documento en línea, la realización de la aplicación, beneficia a los interesados de dicho documento.

El gobernador Rolando Zapata Bello, como parte de una estrategia digital para la educación, presento el servicio de impresión en línea de Certificado de Estudios, que lleva como nombre “Portal de Certificados en Línea”. Este servicio contribuye a alcanzar mejores niveles en la atención al público usuario de obtener este documento a quienes cursaron sus estudios ministerio de educación México- Yucatán.

Relación: esta implementación del servicio ya es un ahorro de tiempo y satisfacción de los usuarios de toda la región de Yucatán para a obtener el certificado en línea.

3.1.2. Antecedentes a Nivel Nacional

Sistema de intermediación digital de la SUNARP, es una plataforma de servicios que le permite realizar el procedimiento de inscripción

registral a través de la presentación electrónica de documentos con firma digital del notario. Esta plataforma de SID SUNARP se encuentra disponible para presentar electrónicamente partes notariales de los siguientes actos:

- Consulta de empresas (MYPES) - (Servicio Registral disponible a nivel nacional).
- Otorgamiento de poderes en el Registro de Personas Naturales (Servicio registral disponible a nivel nacional).
- Compraventa de vehículo en el Registro de Propiedad Vehicular (Servicio registral disponible solo para Vehículos registrados en la Oficina Registral de Lima).
- Compraventa de predios - (Servicio registral disponible solo para predios registrados en la Oficina Registral de Lima).

Los tramites y solicitud a este sistema se realizan con documento escaneados y de esta manera no pueden ser modificados.

Leyes:

- Ley N° 27444, Ley del Procedimiento Administrativo General y D.S. N° 027-2007- PCM.
- D.S. N° 010-2016-MINEDU, artículo 1°.
- D.S. N° 001-2015- MINEDU, artículo 102°, literal j.
- D.S. N° 040-84-ED.
- RVM N° 047-84-ED.
- RVM N° 077-84-ED.
- Texto único de procedimientos administrativos (TUPA).
- D.S N°009-2005-ED Reglamento de la Gestión del Sistema Educativo.

3.2. Base Teórica.

3.2.1.Sistema de Información

Para Mason y Mitroff (1973), “Un sistema de información consiste de por lo menos una persona, con un cierto tipo psicológico en fase de un problema dentro de un contexto organizacional que necesita de un sistema generador de evidencia para llegar a una solución (esto es, para seleccionar algún tipo de acción), y que esa evidencia es disponible para él a través de algún modo de presentación”. De esta manera los autores muestran las palabras claves que comprenden un sistema de información.

Para Telchroew (1976) citado por Silva Fernández (2009), “Un sistema de información es una colección de personas, procedimientos y equipos; diseñado, construido, operado y mantenido para colecciones, registros, procesar, almacenar, recuperar y mostrar información”.

Samuelson (1977) dice: “Sistema de información es la combinación de recursos humanos y materiales que resultan de las operaciones de almacenar, recuperar y usar datos con el propósito de una gestión eficiente en las operaciones de las organizaciones”.

Lucey (1987) define un sistema de información como: “Un sistema para convertir datos procedentes del exterior o del interior del mismo en información y para comunicar ésta en forma apropiada a los directivos de todos los niveles y de todas las funciones para facilitar la toma de decisiones efectivas y oportunas para la planeación, dirección y control de las actividades de las cuales ellos son responsables”.

Senn (1993) menciona: “El sistema de información es un conjunto de personas, datos y procedimientos que funcionan en conjunto. El énfasis en sistema significa que los variados componentes buscan un objetivo común para apoyar las actividades de la organización. Éstas incluyen operaciones diarias de la empresa, la comunicación de los datos e informes, la administración de actividades y la toma de decisiones”.

Para Debons (1991) citado por Muñoz Cañavate (2003), “Sistema de información es un conjunto de personas, maquinaria y

procedimientos que integrados hacen posible a los individuos trabajar con inputs y demandas que aparecen en el trabajo cotidiano”.

Comas Rodríguez (2010) claramente define, “Un sistema de información es un sistema para convertir datos en información y para comunicar ésta adecuadamente a quien la necesite dentro o fuera de una organización facilitando la toma de decisiones acertadas y oportunas para la planeación, organización, dirección y control de las actividades a realizar”.

3.2.1.1. Componentes de un sistema de información

Datos: Hechos concretos, en su estado primario, que el sistema toma para producir información.

Hardware: Una computadora y su equipo periférico: dispositivos de entrada, salida y almacenamiento. El hardware también incluye equipos de comunicación de datos.

Software: Conjunto de instrucciones que indican a la computadora como obtener datos, procesarlos, desplegar información y almacenar datos e información.

Telecomunicaciones: Hardware y software que facilita la transmisión y recepción rápida de texto, imágenes, sonido y animación en forma de datos electrónicos.

Personal: Profesionales y usuarios de los sistemas de información que analizan las necesidades de información de una organización, diseñan y construyen sistemas de información, describen programas de computadora, operan el Hardware y mantienen el software.

Procedimientos diferentes: Reglas para lograr operaciones óptimas y seguras en el procesamiento de datos; los procedimientos incluyen prioridades al ejecutar aplicaciones en la computadora, además de medidas de seguridad.

En conclusión, los sistemas de información permiten almacenar y manipular la información de una organización, para luego presentarla de manera adecuada y que ésta pueda ser aprovechada por los miembros de la misma.

3.2.2. Certificado de Estudios

Es un documento oficial que expide una autoridad de la educación, en el cual se hacen constar las asignaturas o materias cursadas por un estudiante, las calificaciones y/o el número de créditos que corresponden a cada una de dichas asignaturas, todo lo cual se obtiene del expediente académico individual.

Barité, Mario (2001), acerca del Certificado de Estudios, documento otorgado por instituciones educativas que acredita la realización de un proceso formativo según normativas establecidas.

3.2.3. Acta de Notas

Documento oficial que expide un docente de la institución educativa, en el cual se hacen constar las notas obtenidas por los estudiantes del aula, las calificaciones y/o el número de créditos que corresponden a cada una de dicha aula, todo lo cual se obtiene del expediente académico del aula.

3.2.4. Procedimiento para Visación

Consta en Verificar la conformidad del Certificado de Estudio con las Actas de Evaluación Final que emiten las instituciones Educativas Públicas y Privadas.

BASE LEGAL:

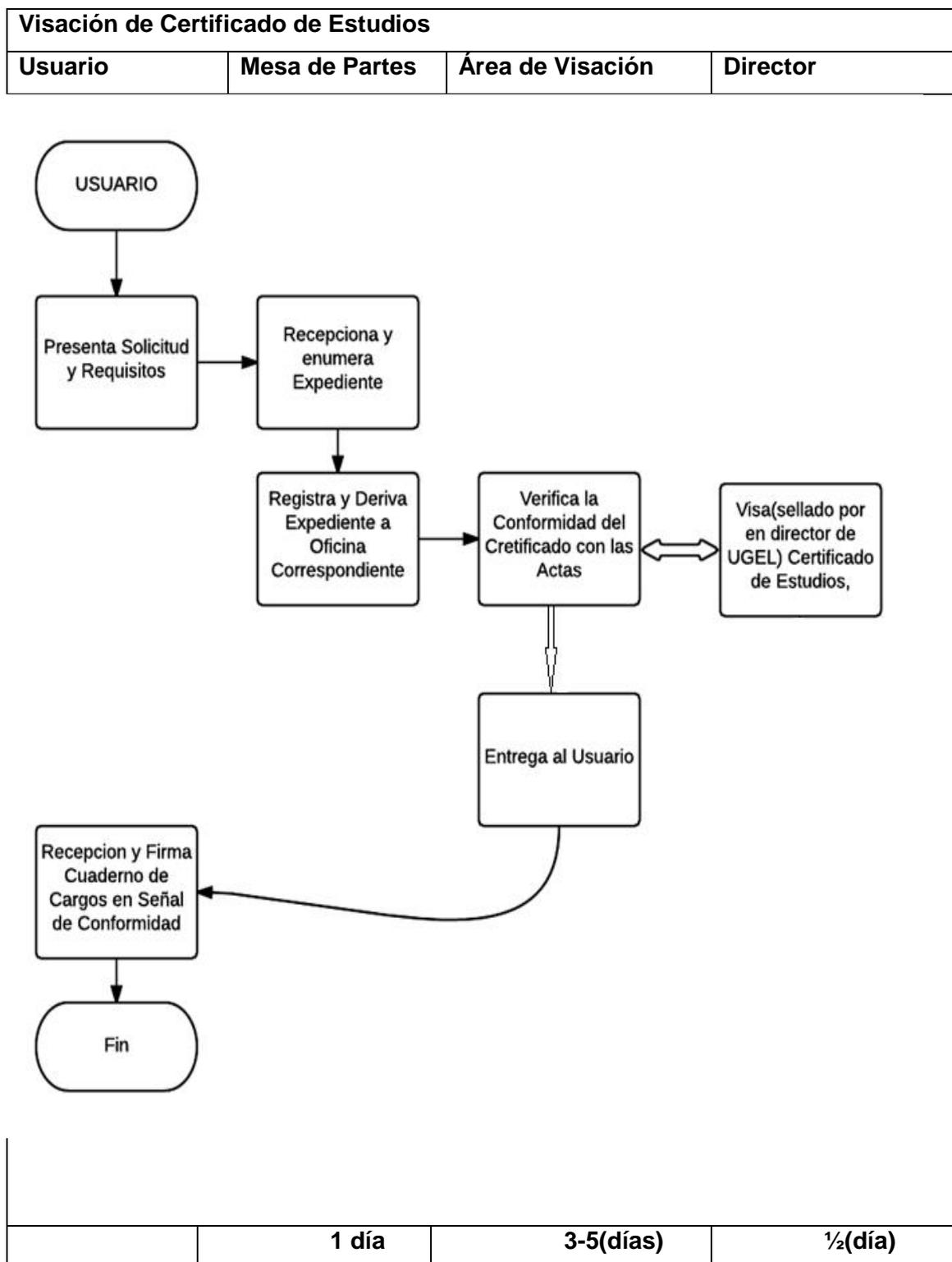
- Ley N° 27444, Ley del Procedimiento Administrativo General y D.S. N° 027-2007-PCM.
- Ley 27815 Ley de Código de Ética de la Función Pública.
- D.S. N° 03-83-ED, N° 04-83-ED, N° 040-84-ED,
- RVM N° 047-84-ED; RVM N° 077-84-ED.
- TUPA.
- D.S N°009-2005-ED Reglamento de la Gestión del Sistema Educativo.

REQUISITOS:

- Certificado de Estudio original.
- Solicitud de Visación.
- Recibo por derecho de pago en Caja de acuerdo al TUPA.

Flujograma del procedimiento de Visación de Certificado de Estudios

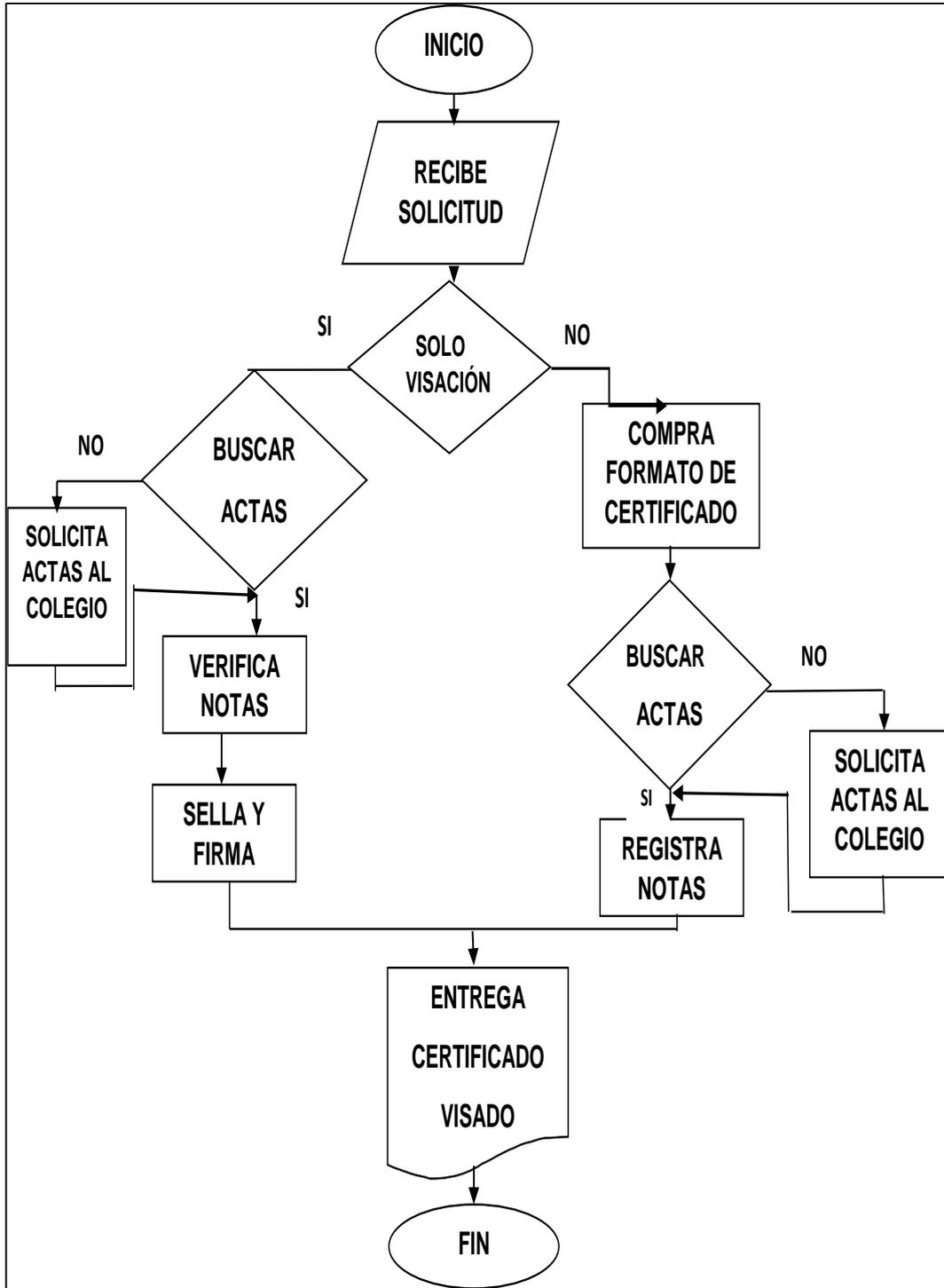
Imagen 1 Representación del Flujo del procedimiento Visación



Fuente: Proceso visar UGEL

Flujograma del proceso de Verificación de Notas del Certificado de Estudios

Imagen 2: Proceso de Verificación de las Notas del Certificado de Estudios



Fuente: Elaboración propia

3.2.5. Visación de Certificado de Estudios

Documento oficial que expide la UGEL, en el cual se hace constar la veracidad del contenido de las notas obtenidas por los estudiantes, las calificaciones y/o el número de créditos que corresponden a cada una del estudiante.

3.2.6. Sistema Web

El sistema Web surge en 1998 como una nueva disciplina orientada a solucionar los problemas derivados de una proliferación de sistemas web de baja calidad, realizados con una carencia completa de proceso. Esta nueva disciplina identifica nuevos elementos propios de las aplicaciones web que no se cubren en las Ciencias de la Computación, en la Ingeniería del Software o en los Sistemas de Información. La Ingeniería Web como disciplina no es una copia de la Ingeniería del Software, aunque parte de un conjunto de principios bien arraigados en ésta últimas, adaptándolos a la naturaleza más abierta y flexible de la Web.

3.2.7. Aplicaciones Web

Las aplicaciones web son soluciones informáticas que los usuarios utilizan accediendo a un servidor web a través de Internet o su red interna (intranet). Como interfaz con la aplicación se utiliza un navegador de Internet. Dicho de otra forma, es un tipo de software diseñado para funcionar sobre un servidor web y ser visualizado mediante un navegador (cliente).

Es importante mencionar que una página web puede contener elementos que permiten una comunicación activa entre el usuario y la información. Esto permite que el usuario acceda a los datos de modo interactivo, gracias a que la página responderá a cada una de sus acciones, como por ejemplo rellenar y enviar respuesta.

Características de la web

Según su propio creador, Berners - Lee, la Web es un sistema que presenta las siguientes características:

Hipermedial: En la Web podemos manejar información multimedia y navegar a través de ella.

Distribuido: A diferencia de las antiguas y enormes bases de datos que concentraban la información físicamente en un único lugar, la Web es un sistema compuesto por miles de servidores localizados en cientos de ciudades del mundo que están interconectadas entre sí.

Heterogéneo: Por ser un servicio relativamente nuevo, la Web tiene la ventaja de poder reunir servicios y protocolos más antiguos (como Gopher, los News, FTP, e inclusive el correo electrónico), de modo tal de presentar la información desde un único programa cliente.

Colaborativo: Ésta es una característica sustancial y la que posiblemente le haya dado el mayor empuje a su crecimiento, ya que cualquier persona, en cualquier parte del mundo, puede agregar información a la Web para que luego pueda ser consultada por el resto de los usuarios.

3.2.8. Base de datos

Para establecer una definición de lo que es una base de datos se debe hacer referencia a diversos autores como por ejemplo: Michael V. (2007), define una base de datos como: Una colección de datos persistentes que pueden compartirse e interrelacionarse, Esta visones muy general y enfatiza en la persistencia de los datos (es decir mantener los datos almacenados de manera estable), además es importante en este concepto la idea de interrelación porque veremos luego que es una de las principales características del modelo relacional de bases de datos. Partiendo del criterio de Piattini, (2006), se define la base de datos como: Colección o depósito de datos integrados, almacenados en soporte secundario (no volátil) y con redundancia controlada. Los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de ellos, y su definición (estructura de la base de datos) única y almacenada junto con los datos, se ha de apoyar en un modelo de datos, el cual ha de permitir captar las interrelaciones y restricciones existentes en el mundo real. Los procedimientos de

actualización y recuperación, comunes y bien determinados, facilitarán la seguridad del conjunto de los datos. Igualmente, la Conference de Statisticiens Européens (1977) define la base de datos como: Colección o depósito de datos, donde los datos están lógicamente relacionados entre sí, tienen una definición y descripción comunes y están estructurados de una forma articular. Una base de datos es también un modelo del mundo real y, como tal, debe poder servir para toda una gama de usos y aplicaciones. Partiendo de todos estos criterios se denomina "Base de Datos" a la colección de datos lógicamente coherente con algún tipo de significado inherente, que permite el almacenamiento de datos de forma segura y confiable. Piatinni, Mario y otros (1996) ha indicado un conjunto de características de las bases de datos las cuales son las siguientes:

- Control centralizado de los datos.
- Integridad de los datos.
- Minimización de las redundancias.
- Independencia de los datos y las aplicaciones.
- Acceso concurrente a los datos.
- Costo mínimo de almacenamiento y mantenimiento.
- Versatilidad para la representación de relaciones.
- Establecimiento de medidas de seguridad.
- Facilidad para el cambio (hardware y software).

3.2.9. Sistemas gestores de bases de datos

Ramos, María (2006), definen un Sistema Gestor de Bases de Datos (SGBD), también llamado Data Base Management System (DBMS) como una colección de datos relacionados entre sí, estructurados y organizados, y un conjunto de programas que acceden y gestionan esos datos. La colección de esos datos se denomina Base de Datos.

Además, se puede decir que: Un Sistema Gestor de base de datos (SGBD) es un conjunto de programas que permiten crear y mantener una Base de datos, asegurando su integridad, confidencialidad y seguridad.

Los Sistemas Gestores de Base de Datos son paquetes de software muy complejos que deben proporcionar una serie de servicios que van a permitir almacenar y explotar los datos de forma eficiente.

Componentes de los sistemas gestores de bases de datos (SGBD) Según (Ramos, María2006), los componentes principales son los siguientes:

Lenguajes de los Sistemas Gestores de Base de Datos

Todos los SGBD ofrecen lenguajes e interfaces apropiadas para cada tipo de usuario: administradores, diseñadores, programadores de aplicaciones y usuarios finales.

Los lenguajes van a permitir al administrador de la base de datos especificar los datos que componen la base de datos, su estructura, las relaciones que existen entre ellos, las reglas de integridad, los controles de acceso, las características de tipo físico y las vistas externas de los usuarios. Los lenguajes del Sistema Gestor de Base de datos se clasifican en:

- Lenguaje de definición de datos (LDD o DDL): se utiliza para especificar el esquema de la base de datos, las vistas de los usuarios y las estructuras de almacenamiento.

Es el que define el esquema conceptual y el esquema interno. Lo utilizan los diseñadores y los administradores de la base de datos.

- Lenguaje de manipulación de datos (LMD o DML): se utilizan para leer y actualizar los datos de la base de datos. Es el utilizado por los usuarios para realizar consultas, inserciones, eliminaciones y modificaciones. Los hay procedurales, en los que el usuario será normalmente un programador y especifica las operaciones de acceso a los datos llamando a los procedimientos necesarios. Estos lenguajes acceden a un registro y lo procesan.

La mayoría de los Sistemas Gestores de Base de Datos comerciales incluyen lenguajes de cuarta generación (4GL) que permiten al usuario desarrollar aplicaciones de forma fácil y rápida, también se les llama herramientas de desarrollo.

Diccionario de datos

El diccionario de datos es el lugar donde se deposita información acerca de todos los datos que forman la base de datos. Es una guía en la que se describe la base de datos y los objetos que la forman.

El diccionario contiene las características lógicas de los sitios donde se almacenan los datos del sistema, incluyendo nombre, descripción, alias, contenido y organización. Identifica los procesos donde se emplean los datos y los sitios donde se necesita el acceso inmediato a la información.

Seguridad e integridad de datos

Un SGBD proporciona los siguientes mecanismos para garantizar la seguridad e integridad de los datos:

- Debe garantizar la protección de los datos contra accesos no autorizados, tanto intencionados como accidentales. Debe controlar que sólo los usuarios autorizados accedan a la base de datos.
- Los SGBD ofrecen mecanismos para implantar restricciones de integridad en la base de datos. Estas restricciones van a proteger la base de datos contra daños accidentales. Los valores de los datos que se almacenan deben satisfacer ciertos tipos de restricciones de consistencia y reglas de integridad, que especificará el administrador de la base de datos. El SGBD puede determinar si se produce una violación de la restricción.
- Proporciona herramientas y mecanismos para la planificación y realización de copias de seguridad y restauración.
- Debe ser capaz de recuperar la base de datos llevándola a un estado consistente en caso de ocurrir algún suceso que la dañe.
- Debe asegurar el acceso concurrente y ofrecer mecanismos para conservar la consistencia de los datos en el caso de que varios usuarios actualicen la base de datos de forma concurrente.

El administrador de la base de datos

En los sistemas de gestión de base de datos actuales existen diferentes categorías de usuarios. Estas categorías se caracterizan

porque cada una de ellas tiene una serie de privilegios o permisos sobre los objetos que forman la base de datos.

El administrador de base de datos tiene una gran responsabilidad ya que posee el máximo nivel de privilegios. Será el encargado de crear los usuarios que se conectarán a la base de datos. En la administración de una base de datos siempre hay que procurar que haya el menor número de administradores, a ser posible una sola persona.

El objetivo principal de un administrador de base de datos es garantizar que la base de datos cumple los fines previstos por organización, lo que incluye una serie de tareas como:

- Instalar el Sistema Gestor Base de Datos en el sistema informático.
- Crear las bases de datos que se vayan a gestionar.
- Crear y mantener el esquema de la base de datos.
- Crear y mantener las cuentas de usuario de la base de datos.
- Arrancar y parar los Sistemas Gestores de base de datos, y cargar las bases de datos con las que se ha de trabajar.
- Colaborar con el administrador del Sistema Operativo en las tareas de ubicación, dimensionado y control de los archivos y espacios de disco ocupados por el Sistema Gestor Base de Datos.
- Colaborar en las tareas de formación de usuarios.
- Establecer estándares de uso, políticas de acceso y protocolos de trabajo diario para los usuarios de la base de datos.
- Suministrar la información necesaria sobre la base de datos a los equipos de análisis y programación de aplicaciones.
- Efectuar tareas de explotación

Algunos ejemplos de Sistemas Gestores de Base de Datos son los siguientes Oracle, PostgreS SQL, SQL Server, MySQL, Access, Base (Open Office).

3.2.10. Servidor

Se puede definir como servidor a un ordenador o máquina informática que está al “servicio” de otras máquinas, ordenadores o personas llamadas clientes y que le suministran a estos, todo tipo de información.

Según Marchionni, Enzo (2011), 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.

Tipos de servidores:

Según Marchionni, Enzo A. (2011), clasifica diferentes tipos de servidores, y pueden ser virtuales o físicos. Podemos clasificarlos según sus capacidades, fabricantes y servicios prestados.

A continuación, describiremos esta última categorización:

- **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 redireccionan a clientes y servidores de seguridad, analizadores y replicadores.

Algunos también brindan opciones de seguridad, como anticipan, 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.
- **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.
- **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 redireccionar 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. Sólo 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.

3.2.11. MySQL

MySQL es un sistema de gestión de bases de datos relacional, fue creada por la empresa sueca MySQL AB, la cual tiene el copyright del código fuente del servidor SQL, así como también de la marca.

Según Welling, Luke, (2005), MySQL es un sistema para la administración de base de datos relacionales (RDBMS) rápido y

sólido. Las bases de datos permiten almacenar, buscar, ordenar y recuperar datos de forma eficiente. El servidor MySQL controla el acceso a los datos para garantizar el uso simultáneo de varios usuarios para proporcionar acceso a dichos datos y para asegurar de que solo obtienen acceso a ellos los usuarios con autorización. Por lo tanto, MySQL es un servidor multiusuario y de subprocesamiento múltiple. MySQL se distribuye bajo un sistema de licencia dual. Puede utilizarse bajo una licencia de código abierto (GPL), que es gratuita mientras cumpla las condiciones de la misma. Si desea distribuir una aplicación que no se GPL y que incluya MySQL, puede adquirir una licencia comercial.

Ventajas de MySQL:

Según Welling, Luke y otros (2005), hace referencia a las siguientes ventajas que tiene el MySQL que son:

- Rendimiento rápido.
- Bajo coste.
- Facilidad de uso.
- Portabilidad.
- Código fuente.
- Disponibilidad de asistencia técnica.

Motores de almacenamiento de MySQL

El motor de almacenamiento (storage-engine) se encarga de almacenar, manejar y recuperar información de una tabla. Los motores más conocidos son MyISAM e InnoDB. La elección de uno u otro dependerá mucho del escenario donde se aplique.

En la elección se pretende conseguir la mejor relación de calidad acorde con nuestra aplicación. Si necesitamos transacciones, claves foráneas y bloqueos, tendremos que escoger InnoDB. Por el contrario, escogeremos MyISAM en aquellos casos en los que predominen las consultas SELECT a la base de datos.

InnoDB dota a MySQL de un motor de almacenamiento transaccional (conforme a ACID) con capacidades de commit (confirmación), rollback (cancelación) y recuperación de fallos. InnoDB realiza bloqueos a nivel de fila y también proporciona funciones de lectura consistente sin bloqueo al estilo Oracle en sentencias SELECT.

Estas características incrementan el rendimiento y la capacidad de gestionar múltiples usuarios simultáneos. No se necesita un bloqueo escalado en InnoDB porque los bloqueos a nivel de fila ocupan muy poco espacio. InnoDB también soporta restricciones FOREIGN KEY. En consultas SQL, aún dentro de la misma consulta, pueden incluirse libremente tablas del tipo InnoDB con tablas de otros tipos.

MyISAM trata tablas no transaccionales.

Proporciona almacenamiento y recuperación de datos rápida, así como posibilidad de búsquedas fulltext. MyISAM se soporta en todas las configuraciones MySQL, y es el motor de almacenamiento por defecto a no ser que tenga una configuración distinta a la que viene por defecto con MySQL.

3.2.12. LENGUAJES DE PROGRAMACIÓN

Según López, Joaquín J. (2011), los lenguajes de programación son como idiomas que constituyen el sistema de comunicación entre el 'hombre y el ordenador, mediante el cual se transmiten a este las instrucciones e información en un formato comprensible para la máquina. Además, podemos definir como lenguaje de programación, es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. Por lo tanto, un lenguaje de programación es un modo práctico para que los seres humanos puedan dar instrucciones a un equipo.

Clasificación de los lenguajes de programación

Según Quero & Enrique, (2003) clasifica a los lenguajes de programación de la siguiente manera:

Lenguajes de bajo nivel

Son lenguajes totalmente dependientes de la máquina, es decir que el programa que se realiza con este tipo de lenguajes no se puede migrar o utilizar en otras máquinas.

Al estar prácticamente diseñados a medida del hardware, aprovechan al máximo las características del mismo.

Lenguajes de alto nivel

Son aquellos que se encuentran más cercanos al lenguaje natural que al lenguaje máquina. Están dirigidos a solucionar problemas mediante el uso de EDD's.

Un lenguaje de programación es un lenguaje artificial que puede ser usado para controlar el comportamiento de una máquina, especialmente una computadora. Estos se componen de un conjunto de reglas sintácticas y semánticas que permiten expresar instrucciones que luego serán interpretadas.

Lenguaje de programación php

Según González, Enrique (2012) PHP es un lenguaje de código abierto muy popular, adecuado para desarrollo web y que puede ser incrustado en HTML. Es popular porque un gran número de páginas y portales web están creadas con PHP. Código abierto significa que es de uso libre y gratuito para todos los programadores que quieran usarlo. Incrustado en HTML significa que en un mismo archivo vamos a poder combinar código PHP con código HTML, siguiendo unas reglas.

También Álvarez, Migel (2007) en su artículo define al lenguaje PHP como un lenguaje para programar scripts del lado del servidor, que se incrustan dentro del código HTML. Este lenguaje es gratuito y multiplataforma.

Además podemos decir que PHP es el acrónimo de Hipertexto Preprocesor, el cual es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Características del Lenguaje de Programación PHP

Según (Vaswani, 2014), manifiesta que el lenguaje de programación PHP tiene las siguientes características únicas:

- Rendimiento.
- Portabilidad.
- Fácil de usar.
- Código libre.
- Soporte comunitario.
- Soporte de aplicaciones de terceros.

3.2.13. XAMPP

Xampp es un servidor independiente de plataforma software libre y es de distribución GNU, que consiste principalmente en la base de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP, MySQL, PHP y PERL. La instalación de estos componentes, de forma independiente e individual, es en ocasiones complicada. XAMPP es una forma fácil de instalar estos módulos; solo se requiere descargar y ejecutar como archivo zip, rar o exe. Actualmente, hay cuatro versiones de XAMPP: para Linux, Windows, Solaris y Mac OS X. (Apache, 2015).

3.2.14. Metodología eXtreme Programming (XP)

La programación extrema o eXtreme Programming (XP) es un enfoque de la ingeniería de software formulado por Kent Beck, autor del primer libro sobre la materia, Extreme Programming Explained: Embrace Change (1999). Es el más destacado de los procesos ágiles de desarrollo de software. Al igual que éstos, la programación extrema se diferencia de las metodologías tradicionales principalmente en que pone más énfasis en la adaptabilidad que en la previsibilidad. Los defensores de XP consideran que los cambios de requisitos sobre la marcha son un aspecto natural, inevitable e incluso deseable del desarrollo de proyectos. Creen que ser capaz de adaptarse a los cambios de requisitos en cualquier punto de la vida del proyecto es una

aproximación mejor y más realista que intentar definir todos los requisitos al comienzo del proyecto e invertir esfuerzos después en controlar los cambios en los requisitos.

VALORES XP

La programación extrema define 5 valores primordiales que son un motor para la realización de actividades que simplifiquen realizar el software, en el siguiente gráfico se definen los valores de XP.

Imagen 3 : Valores XP

The infographic consists of five horizontal colored boxes, each representing a value of XP. Each box contains an icon, a title, and a description.

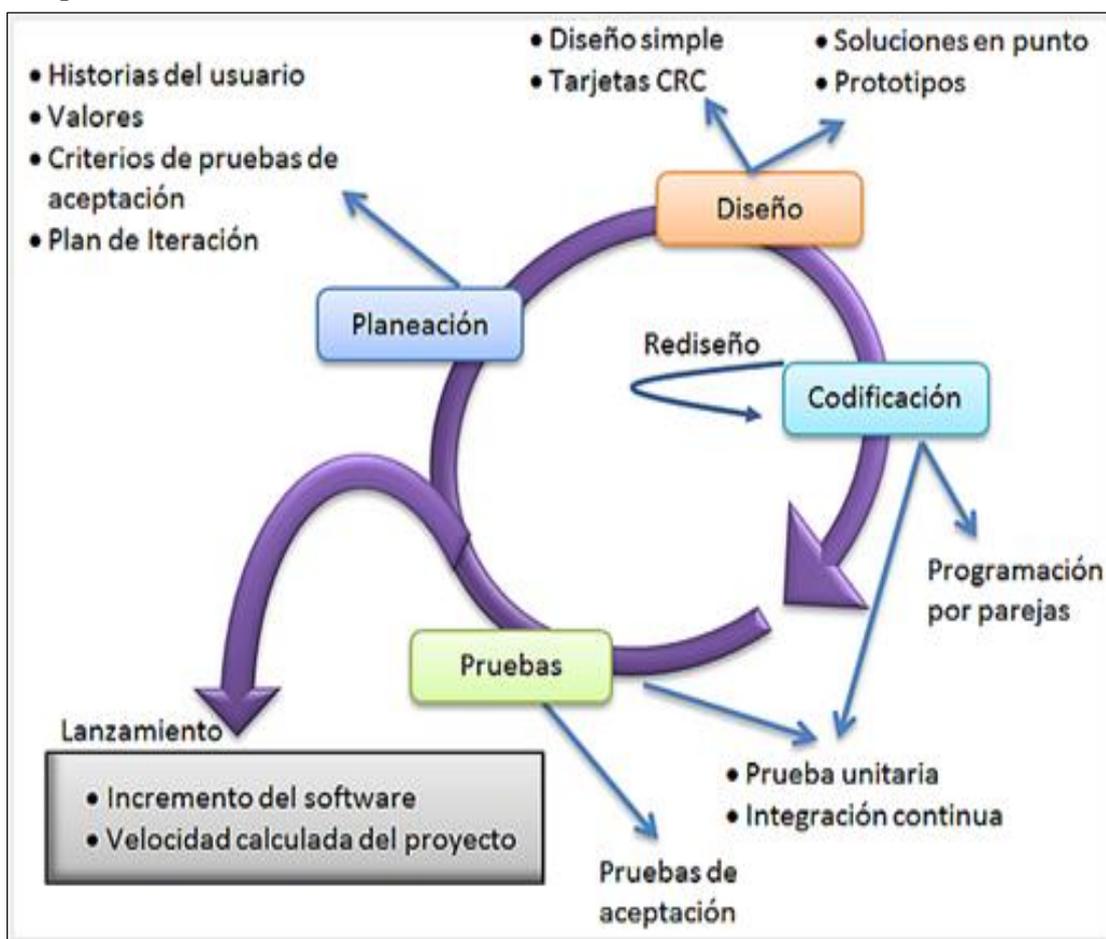
- Comunicación** (Yellow box): Icon of four people. Description: "Es aquí en donde los clientes y desarrolladores establecen ciertos aspectos del software, XP hace énfasis en la estrecha comunicación que debe haber en el equipo de software."
- Simplicidad** (Light Green box): Icon of the equation $1+1=2$. Description: "XP intenta que los desarrolladores diseñen sólo lo más importante del software, de esta forma se termina más rápido con la arquitectura del mismo."
- Retroalimentación** (Green box): Icon of a woman pointing to a whiteboard with letters. Description: "El cliente debe ir familiarizándose con el software, por lo cual cada vez que se realicen módulos son presentados a este."
- Prueba Unitaria** (Cyan box): Icon of a pencil checking off boxes on a list. Description: "La prueba unitaria es la táctica de pruebas, es decir, cada vez que se agregue una funcionalidad, se hacen pruebas"
- Historias del usuario** (Blue box): Icon of a person standing next to several binders. Description: "Son también llamados casos de uso y hacen una documentación de las pruebas"

Fuente: <http://www.marblestation.com/>

PROCESO XP

La forma en la que se establecen las reglas de desarrollo XP se dan en base a la programación orientada a objetos, en este tipo de desarrollo se establecen 4 actividades estructurales que son expuestas en el siguiente gráfico.

Imagen 4: Proceso XP



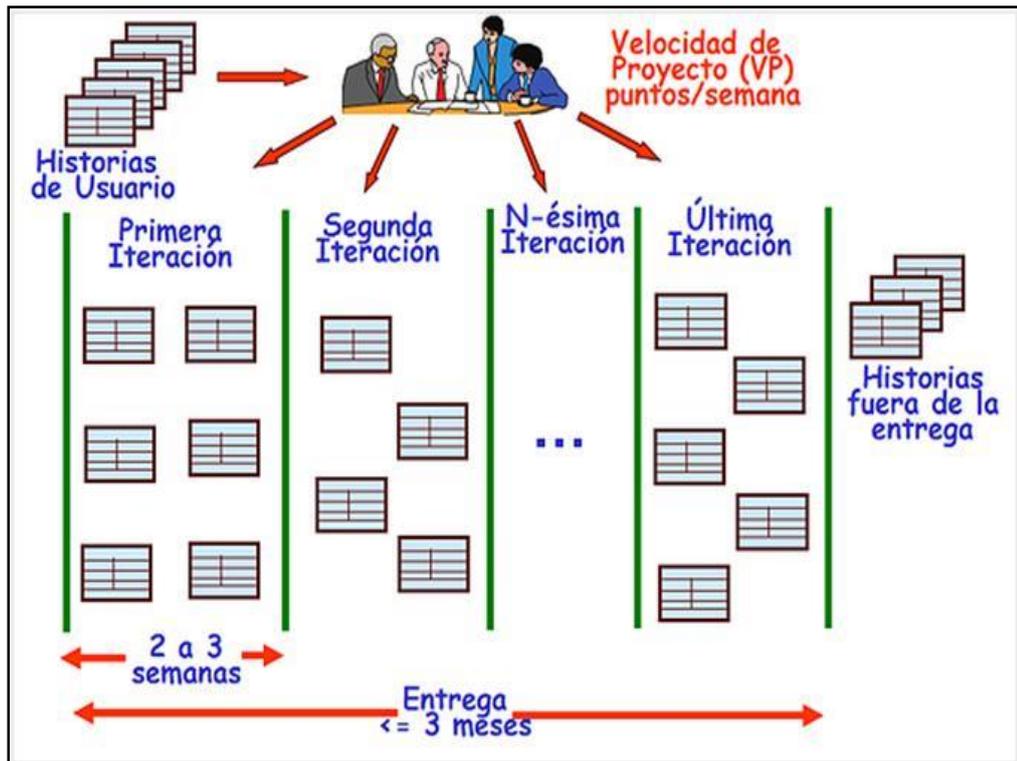
Fuente: <http://www.marblestation.com/>?

PLANEACIÓN

La planificación se realiza por etapas, es decir, que es iterativa, sin embargo, no puede haber una planificación sin que antes haya habido comunicación con el cliente, en esta reunión con el cliente, el establece sus requerimientos lo cual hace que el equipo de software entienda cuáles son las características y funcionalidades que se necesitan.

Cuando el desarrollador entiende lo que escucha del cliente, este puede ser capaz de hacer los casos de uso, en los cuales el cliente puede asignar prioridades entre todas las funciones del software. Tecno Tips.(2012).

Imagen 5: Planificación XP

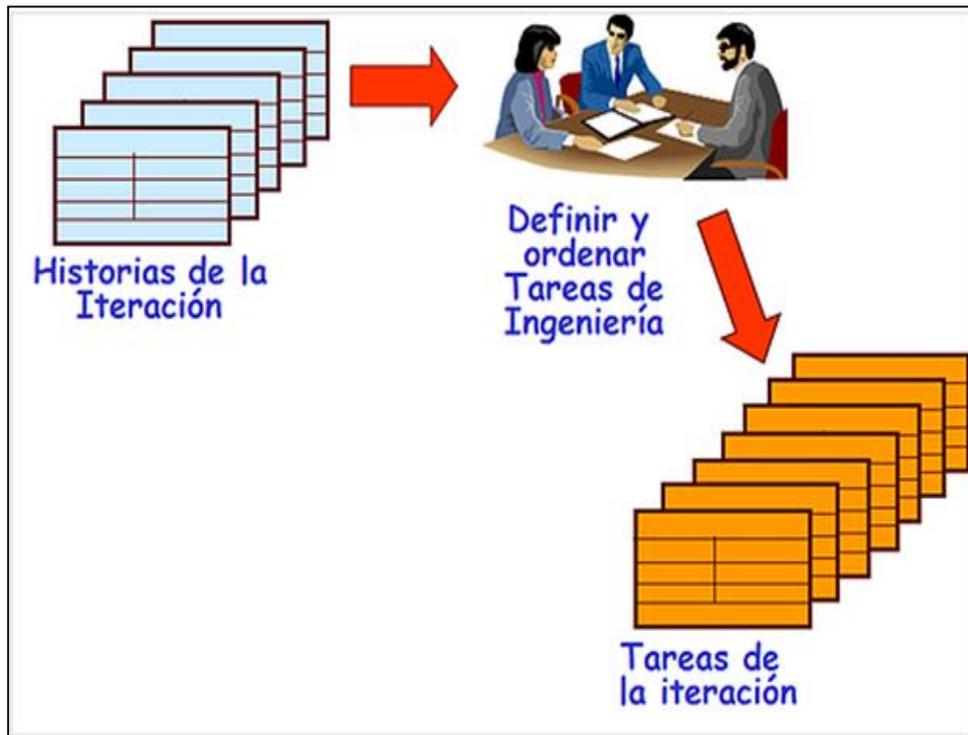


Fuente: <http://www.marblestation.com/>?

DISEÑO

El diseño es la guía para la implementación del sistema, por lo tanto debe ser claro, y para poder ser claro necesita de simplicidad, ya que no sólo será entendido por el programador sino que también en muchas ocasiones, por el usuario. En el diseño se pueden también asignar las responsabilidades y módulos de cada persona en el equipo.

Imagen 6: Diseño en Xp



Fuente: <http://www.marblestation.com/?>

CODIFICACIÓN

Para poder empezar en la codificación, antes se debieron hacer pruebas unitarias de avances en diseño a los clientes, para así, poder establecer los requerimientos primordiales.

Uno de los mejores mecanismos para hacer que la codificación funcione de manera correcta es la unión de dos personas del equipo, es decir, la programación en parejas, cada una de estas personas, con características distintas y especializadas en distintas áreas puede encargarse de tareas distintas dentro de un mismo código.

Imagen 7: Codificación en XP

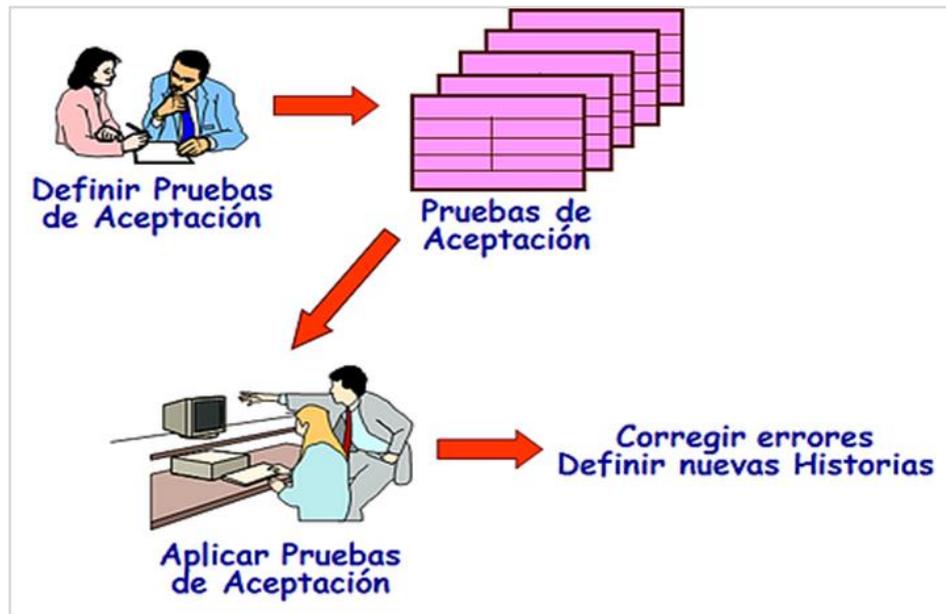


Fuente: <http://www.marblestation.com/>

PRUEBAS

Las pruebas unitarias son la medida de comprobación de la funcionalidad de cada uno de los módulos o componentes del sistema, estas pueden ser ejecutadas a diario y brindan una información del avance que tiene el proyecto.

Imagen 8: Pruebas en XP



Fuente: <http://www.marblestation.com/?>

Hacer pruebas de funcionamiento es primordial si se requiere un software sin errores, y realizar pruebas a diario y por separado ayuda a que lo que se entregue al cliente sea un producto final confiable. Las pruebas no solamente las realiza el desarrollador con su equipo, sino que también interviene el cliente, él tiene la última palabra, y si tiene nuevos requerimientos puede darlos en el momento de las pruebas, a esto se le llama pruebas del cliente.

Principios, roles y prácticas de Programación extrema

Se busca:

- Realimentación rápida.
- Asumir la simplicidad.
- Cambio incremental.
- Aceptar el cambio.
- Hacer trabajo de calidad.

Las Prácticas son:

- El juego de la planificación.
- Pequeñas entregas.
- Metáfora.
- Diseño simple.

- Pruebas.
- Refactorización.
- Programación por parejas.
- Propiedad colectiva.
- Integración continúa.
- 40 horas semanales.
- Cliente en casa.
- Estándares de codificación.

3.2.15. Framework

Según el portal web Jordisan (2006), un framework es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. Puede hacer uso del paradigma MVC (Model-View-Controller) separa en tu aplicación la gestión de los datos, las operaciones, y la presentación, también pueden llegar al detalle de definir los nombres de ficheros, su estructura, las convenciones de programación, etc.

Los frameworks no necesariamente están ligados a un lenguaje concreto, aunque sea así en muchas ocasiones. En el cada vez más popular Ruby on Rails, 'Ruby' es el lenguaje de programación y 'Rails' el framework; por otro lado, JavaServer Faces está orientado a desarrollos en Java. Sin embargo, nada impide definir el mismo framework para lenguajes diferentes: por ejemplo, existe un framework llamado Biscuit cuyo objetivo es prácticamente convertirse en un "PHP on Rails". Eso sí, cuanto más detallado es el framework, más necesidad tendrá de ceñirse a un lenguaje concreto. También es posible que el framework defina una estructura para una aplicación completa, o bien sólo se centre en un aspecto de ella. Siguiendo con los ejemplos, Ruby on Rails ofrece un marco para el desarrollo completo de una aplicación web, mientras que JavaServer Faces está más orientado a la interfaz de usuario.

¿Qué ventajas tiene utilizar un 'framework'?

Las que se derivan de utilizar un estándar; entre otras:

- El programador no necesita plantearse una estructura global de la aplicación, sino que el framework le proporciona un esqueleto que hay que “rellenar”.
- Facilita la colaboración. Cualquiera que haya tenido que “pelearse” con el código fuente de otro programador (¡o incluso con el propio, pasado algún tiempo!) sabrá lo difícil que es entenderlo y modificarlo; por tanto, todo lo que sea definir y estandarizar va a ahorrar tiempo y trabajo a los desarrollos colaborativos.
- Es más fácil encontrar herramientas (utilidades, librerías) adaptadas al framework concreto para facilitar el desarrollo.

¿Es necesario utilizar un 'framework'?

Un desarrollador puede crear toda una aplicación sin seguir ningún framework conocido; puede que sea tan pequeña que no lo considere necesario, que no conozca ninguno que se adapte a sus necesidades, o simplemente no desee dedicar tiempo a seleccionar y utilizar uno. Sin embargo, a medida que la aplicación crece, un programador competente procurará seguir unas determinadas pautas que le faciliten su trabajo de desarrollo y mantenimiento: separación de presentación y lógica, una sintaxis coherente, etc. La evolución natural será hacia que, de algún modo, se construirá su propio framework.

Y en vez de definir un estándar, es mejor utilizar uno ya definido y aprovechar el trabajo de otros muchos desarrolladores. El coste inicial (la curva de aprendizaje) de utilizar un framework se compensa probablemente en cuanto el trabajo de desarrollo crezca mínimamente.

¿Qué 'framework' utilizo?

Buscando en la red se encuentra mucha información sobre los frameworks existentes para las diferentes plataformas y lenguajes. Posiblemente uno de sus principales problemas es que haya demasiados: ya se sabe, lo bueno de los estándares es que hay muchos para elegir. Sin embargo, la elección del framework concreto a utilizar vendrá marcada por:

- El tipo de aplicación a desarrollar.
- El lenguaje de programación y otras tecnologías concretas: base de datos, sistema operativo, etc.

En conclusión

La utilización de un framework en el desarrollo de una aplicación implica un cierto coste inicial de aprendizaje, aunque a largo plazo es probable que facilite tanto el desarrollo como el mantenimiento.

Existen multitud de frameworks orientados a diferentes lenguajes, funcionalidades, etc. Aunque la elección de uno de ellos puede ser una tarea complicada, lo más probable que a largo plazo sólo los mejor definidos (o más utilizados, que no siempre coinciden con los primeros) permanezcan. Y si ninguno de ellos se adapta a las necesidades de desarrollo, siempre es mejor definir uno propio que desarrollar "al por mayor".

3.2.16. Laravel

El portal web Arsys explica que Laravel es un nuevo y poderoso **Framework PHP** desarrollado por **Taylor Otwell**, que promete llevar al lenguaje PHP a un nuevo nivel.

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, que veremos enseguida.

Modelo-Vista-Controlador

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

Manejo de los Datos en Laravel

Laravel incluye una valiosa pieza de software, llamada **Eloquent ORM**. Este ORM se funda en patrón **active record** y su funcionamiento es en extremo sencillo.

En Laravel es opcional el uso de **Eloquent**, pues también dispone de otros recursos que nos facilitan interactuar con los datos, o específicamente la **creación de modelos**.

CAPÍTULO IV

PROPUESTA DE SOLUCIÓN

En este capítulo se describen las fases y procesos ejecutados hasta lograr la implementación del Sistema de Información para Generar Certificados de Estudios del Nivel Básico Regular de la UGEL Andahuaylas, todo dentro de las especificaciones y recomendaciones ofrecidas por la metodología XP, los requerimientos hechos por el usuario del proyecto y el criterio del equipo de programación.

4.1. Tecnologías para el entorno de desarrollo

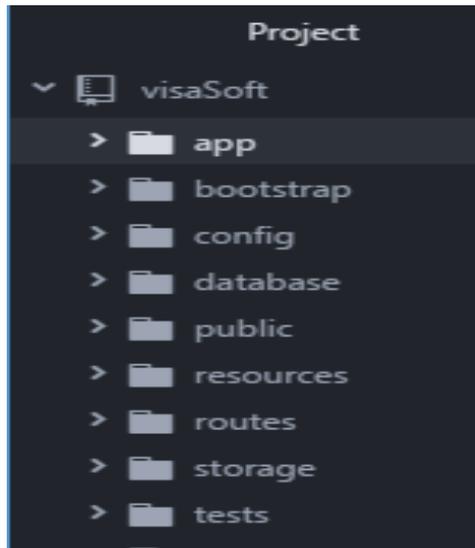
4.1.1. Framework Laravel v5.3

En comparación con otros frameworks, Laravel ha demostrado ser aquél que se mantiene en constante evolución, con una comunidad activa, con gran cantidad de colaboradores que ya supera a otros frameworks y que sigue creciendo. Laravel es la opción a elegir porque muchas de sus características son automatizadas y ha corregido falencias que otros frameworks aún siguen teniendo.

Se ha elegido Laravel 5.3 por ser la última versión liberada al iniciar este proyecto, de este modo garantizar un entorno moderno, ya que a partir de esta elección se determinarán las versiones de las otras tecnologías que se usarán.

Laravel es el framework sobre el cual está programado el sistema, a continuación, se muestra la estructura del proyecto proporcionado por laravel.

Imagen 9 : Estructura VisaSoft en laravel.



Fuente: Elaboración propia

En la imagen se muestra el directorio “app” dentro de él se escribe el código del sistema tales como los controladores, modelos y los archivos requests. En la carpeta “resources” se encuentran los archivos correspondientes a las vistas del sistema.

Laravel disminuye los problemas recurrentes al programar con el lenguaje PHP, permite una sintaxis elegante y expresiva.

A continuación, se mencionan las tecnologías con las que Laravel recomienda trabajar, a pesar de que parezca una imposición por parte del framework se debe aclarar que tienen características robustas y de no usarlas produciría deficiencias en el producto final, además de un desarrollo entorpecido.

PHP 5.6

Es un lenguaje de programación para uso general, del lado del servidor, originalmente diseñado para el desarrollo web. No puede omitirse su uso en este proyecto porque Laravel es un framework php. Los controladores, los modelos, y las clases para manejar los requests del sistema VisaSoft están enteramente escritos con lenguaje PHP. Ejemplo del controlador Turno programado con lenguaje PHP del sistema VisaSoft.

Imagen 10: TurnoController

```
TurnoController.php

class TurnoController extends Controller
{
    public function __construct(){

    }

    public function index(Request $request)
    {

    }

    public function create()
    {

    }

    public function store(TurnoFormRequest $request)
    {

    }

    public function show($id)
    {

    }
}
```

Fuente: Elaboración propia

Además de ser útil para la codificación del sistema, se usó php para correr scripts y ejecutar comandos por consola. Por ejemplo: “**php artisan –versión**” comando que muestra la versión de laravel que se está usando. Entre las tareas que se automatizan con php están: renombrar el proyecto laravel, ejecutar el servidor para correr pruebas del sistema, crear y declarar los modelos, controladores y requests del sistema.

Composer

Es un manejador de dependencias, fue usado para descargar Laravel y todas sus librerías, esta tecnología se encarga, por si sola, de averiguar todos los paquetes necesarios para una instalación correcta de Laravel y descargará automáticamente la versión correcta de cada paquete.

Comando Composer para crear un proyecto Laravel:

```
composer create-project --prefer-dist laravel/laravel  
VisaSoft
```

Artisan

Herramienta que permite ejecutar diferentes tareas por línea de comandos, será usado para la creación de los controladores y modelos del proyecto, entre otras cosas.

Comando para crear un controlador con artisan y php:

php artisan make:controller ActaController

Eloquent:

Es un ORM (Object-Relational Mapping), se encarga de mapear los datos de la Base de Datos y convertirlos en objetos() php y viceversa, esto para evitar usar el lenguaje SQL dentro de nuestras clases php.

En el sistema VisaSoft todas las clases que pretenden ser modelos de la Base de Datos heredan obligatoriamente de la clase Eloquent.

Ejemplo del Modelo Acta del sistema VisaSoft que hereda de la clase Eloquent:

Imagen 11: Eloquent en el modelo Acta

```
Acta.php
1  <?php
2
3  namespace visaSoft;
4
5  use Illuminate\Database\Eloquent\Model;
6
7  class Acta extends Model
8  > {
24
```

Fuente: Elaboración propia

Blade

Es un procesador de templates, para el manejo de vistas en Laravel, si bien se pueden usar otros procesadores más potentes como Twig, Blade satisface todos los requerimientos para el diseño de la interfaz del sistema. Vistas dinámicas manejadoras. Ejemplo de uso de sintaxis Blade en la maquetación de las vistas del

sistema VisaSoft. Las líneas subrayadas corresponden a sintaxis Blade.

Imagen 12: Sintaxis Blade

```
edit.blade.php
2  @extends ('layouts.admin')
3  @section ('contenido')
4      <div class="row">
5          <div class="col-lg-6 col-md-6 col-sm-6 col-xs-12">
6              <h3>Editar Acta: {{ $acta->idacta }}</h3>
7              @if (count($errors)>0)
8                  <div class="alert alert-danger">
9                      <ul>
10                     @foreach ($errors->all() as $error)
11                         <li>{{ $error }}</li>
12                     @endforeach
13                 </ul>
14             </div>
15             @endif
16
```

Fuente: Elaboración propia

4.1.2. Paquete XAMPP v5.6

Es un paquete de instalación, el cual permitirá crear un entorno web en un computador personal, se encargará automáticamente de la instalación y configuración de diferentes tecnologías web como son:

Apache, es un servidor web de código abierto, permite configurar un Hosting Virtual, el cual se encargará de gestionar la entrega o denegación de peticiones de páginas.

Imagen 13: Apache del paquete Xampp



Fuente: Elaboración propia

PHP 5.6 y Perl, lenguajes que serán instalados por el paquete XAMPP, siendo útil para el proyecto la instalación de PHP 5.6 cuya versión es compatible con Laravel.

MariaDB, sistema de gestión de bases de datos, es un fork de MySQL pero con licencia GPL. Viene por defecto a partir de la versión 5.6 de XAMPP.

Imagen 14: MariaDB



Fuente: Elaboración propia

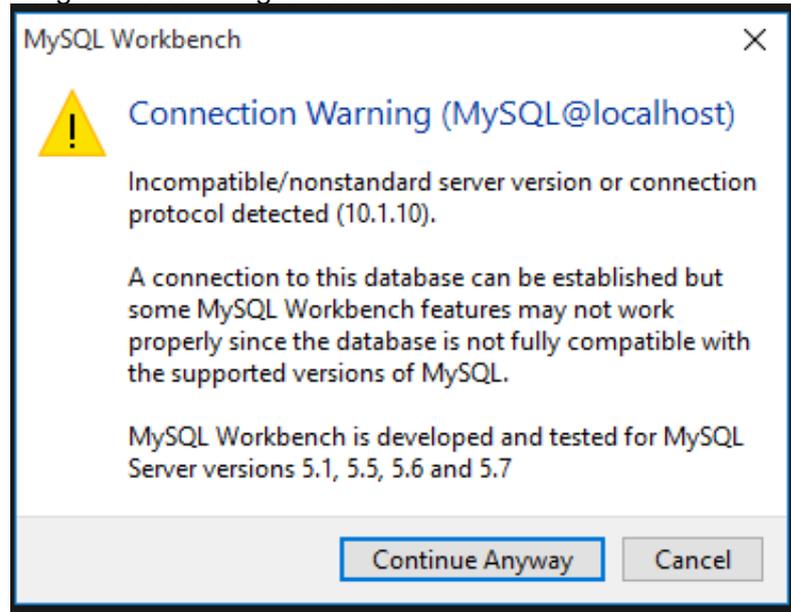
4.1.3.Herramienta MySQL Workbench v6.3

Software que se usará para realizar la administración, diseño, creación y mantenimiento de la Base de datos. Workbench está diseñado para trabajar con MySQL, por lo que se espera tener algunas complicaciones durante el desarrollo al usar MariaDB alerta, a pesar de que ésta, es un derivado de MySQL y sus cambios siempre procuraron tener una alta compatibilidad con este gestor, ya ha tenido

muchas actualizaciones y es probable que Workbench no soporte algunas de sus características.

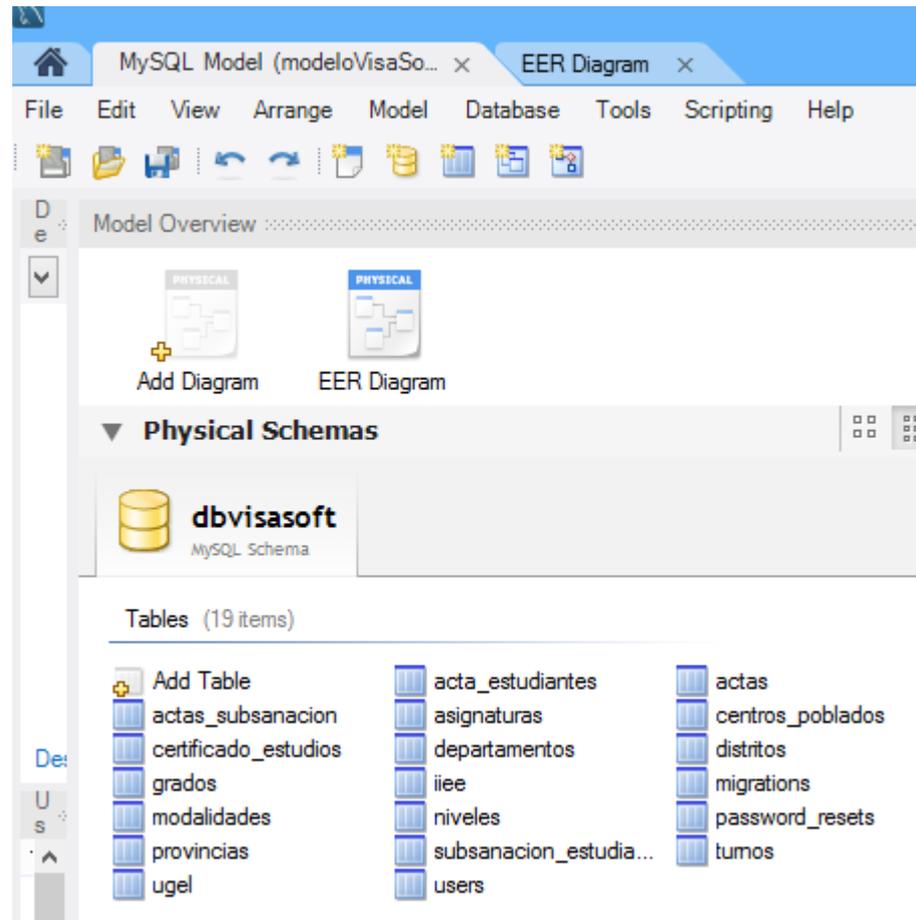
Imagen que muestra un Warning lanzado por Workbench, debido a que VisaSoft trabaja con MaríaDB y workbench con MySQL.

Imagen 15: Warning del Workbench



Fuente: Elaboración propia

Imagen 16: Workbench tablas y BD del Sistema VisaSoft



Fuente: Elaboración propia

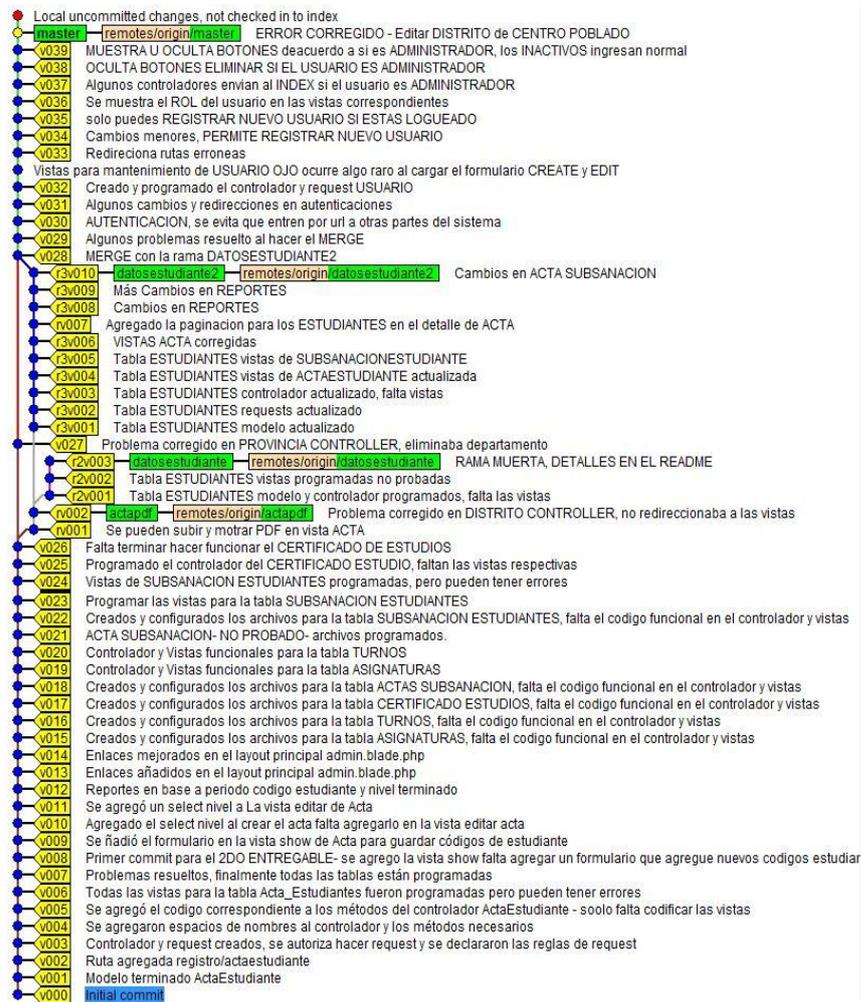
4.1.4. Software de control de versiones GIT

Será usado durante la codificación del sistema, para gestionar los cambios del mismo. Permitirá la administración de las distintas versiones del software y facilitará su mantenimiento. Con GIT se tendrá guardado todos los estados por los que pasó el sistema de manera simple y detallada, se mantendrá un historial del producto para tener una visión general de la evolución de este.

Además, Git permitirá:

- Comparar el código de un archivo y mostrar las diferencias entre versiones.
- Restaurar versiones antiguas.
- Fusionar cambios entre distintas versiones.
- Crear ramas al proyecto y luego fusionarlas.

Imagen 17: GIT



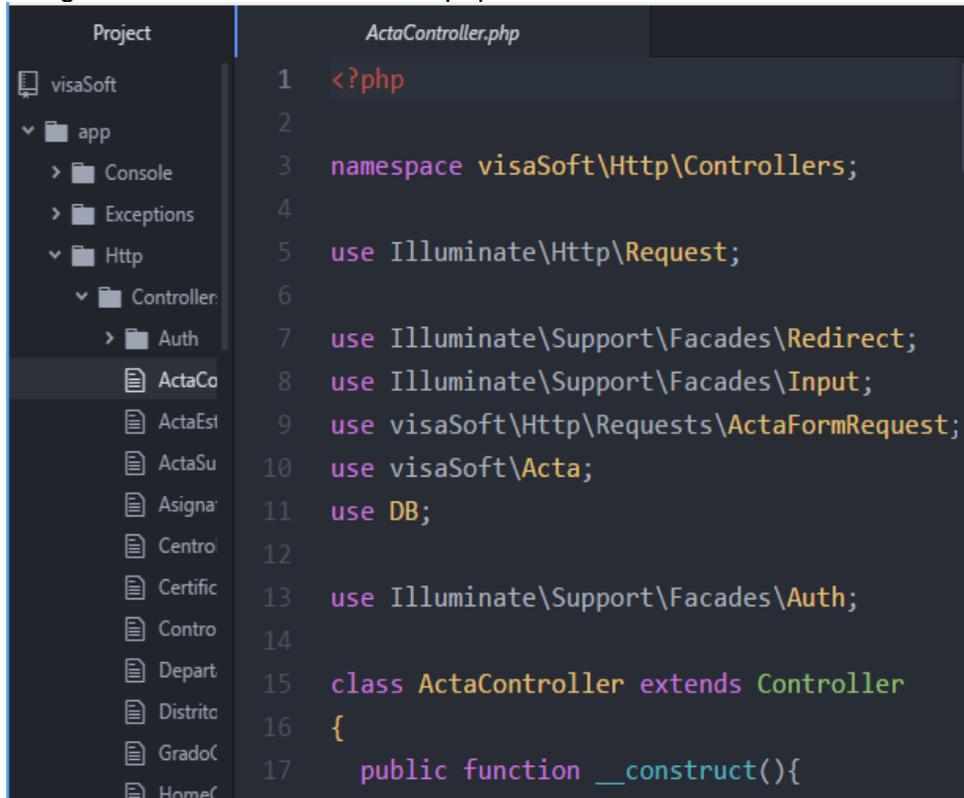
Fuente: Elaboración propia

4.1.5. Editor de Texto ATOM

Es una aplicación de escritorio, construido utilizando tecnologías web, desarrollado por GitHub, viene con Git Control y es compatible con varios lenguajes, entre ellos php, que es de particular interés para el proyecto. Será usado para escribir el código fuente del sistema, ya que Atom resalta la sintaxis del código, funcionalidad que ayudará a mitigar los errores durante la programación. Además, la interfaz de este editor es igual a Sublime Text, editor preferido por los desarrolladores web, y trae muchas funcionalidades de este, con la diferencia de que Atom es completamente gratuito. A su vez Atom

puede extender sus funcionalidades, con la instalación de más de 500 paquetes que tiene a su disposición.

Imagen 18: Clase ActaController.php en Atom



```
1 <?php
2
3 namespace visaSoft\Http\Controllers;
4
5 use Illuminate\Http\Request;
6
7 use Illuminate\Support\Facades\Redirect;
8 use Illuminate\Support\Facades\Input;
9 use visaSoft\Http\Requests\ActaFormRequest;
10 use visaSoft\Acta;
11 use DB;
12
13 use Illuminate\Support\Facades\Auth;
14
15 class ActaController extends Controller
16 {
17     public function __construct(){
```

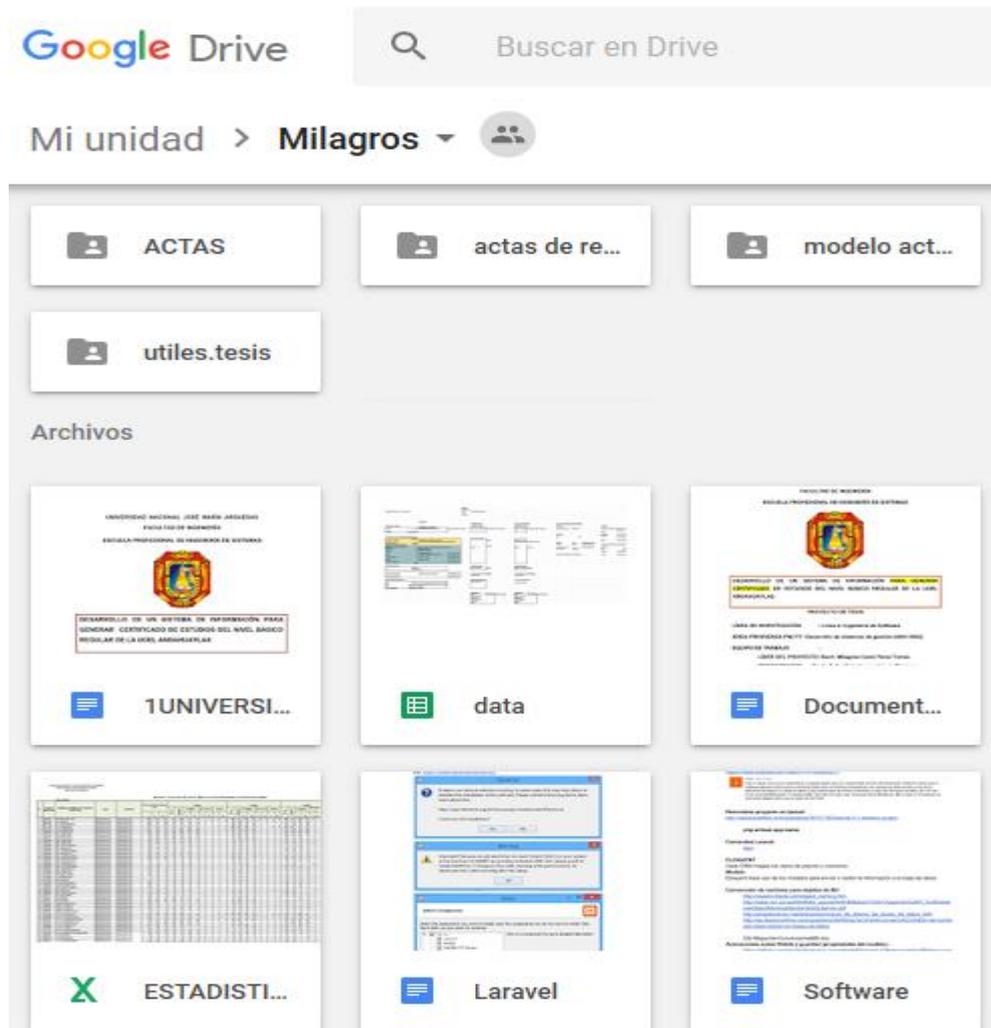
Fuente: Elaboración propia

4.2. Tecnologías de apoyo

4.2.1. Servicio de alojamiento de archivos Google Drive

Ofrece el servicio de almacenamiento en la nube, es gratuito con hasta 15GB de espacio y no necesita configuraciones complejas para funcionar. Será usado para guardar documentos y archivos importantes del proyecto. Si en algún caso los equipos usados para el desarrollo del sistema, se incapacitan y de esta manera se perdiera los documentos y el avance del proyecto, podrían ser recuperados por mantenerse una copia de éstos en Google Drive, ya que estos datos son almacenados en la nube y son protegidos por un equipo especializado. El único impedimento para acceder a estos datos sería no contar con acceso a Internet.

Imagen 19: captura Google Drive



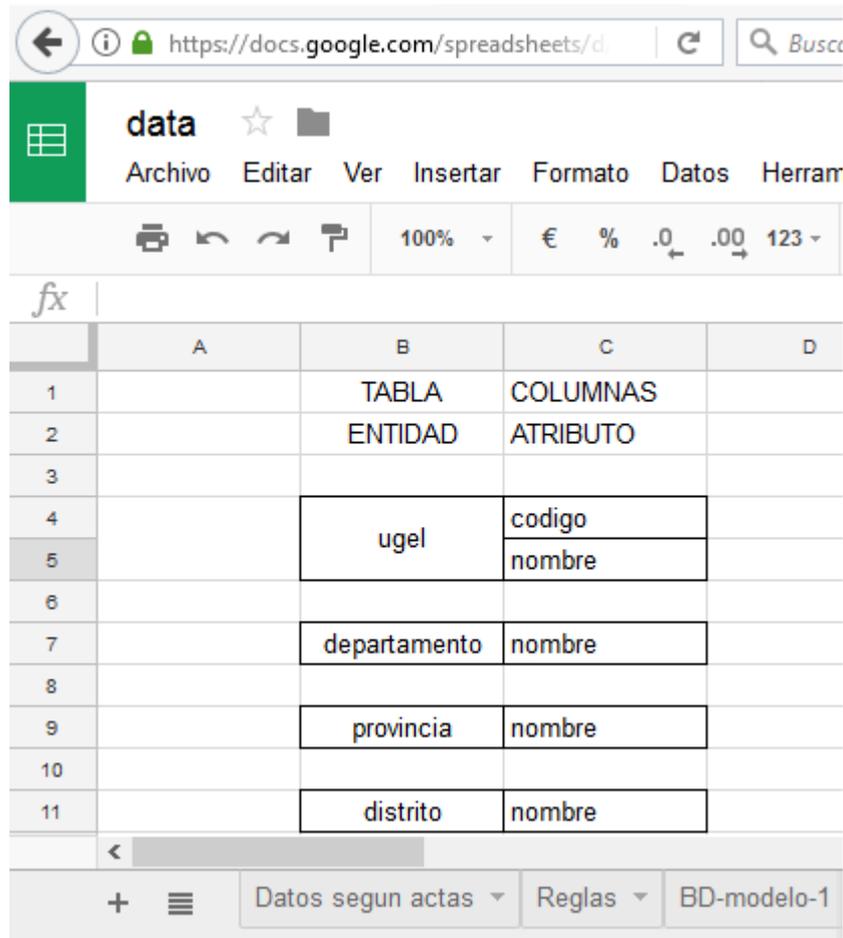
Fuente: Elaboración propia

4.2.2. Suit de ofimática Google Docs

El servicio de ofimática de Google en la nube, permite crear documentos, editarlos y compartirlos en la red con otros usuarios. Para el proyecto se usará el procesador de texto y hojas de cálculo, para las labores de análisis, documentación y diseño del sistema.

Imagen del documento de hojas de cálculo creado y abierto con Google Docs, muestra las tablas de la Base de Datos del sistema VisaSoft

Imagen 20: Tablas de la BD del Sistema VisaSoft



The image shows a Google Sheets spreadsheet titled 'data'. The spreadsheet contains a table with the following structure:

	A	B	C	D
1		TABLA	COLUMNAS	
2		ENTIDAD	ATRIBUTO	
3				
4		ugel	codigo	
5			nombre	
6				
7		departamento	nombre	
8				
9		provincia	nombre	
10				
11		distrito	nombre	

Fuente: Elaboración propia

4.3. DESARROLLO DEL SISTEMA DE INFORMACION VISAssoft

Se desarrolla las fases del sistema de información VISAssoft según la metodología extreme programming(XP).

4.3.1.Planeación

Es el primer paso en el cual se define las historias de usuario con el cliente para la especificación de los requerimientos, permitiendo disminuir la documentación. Para definir las historias de usuario se realizó reuniones con los involucrados a utilizar el sistema de información VISAssoft, con quienes se logró definir las necesidades y problemáticas en la atención al usuario con el proceso actual en la emisión del certificado de Estudios Visado. En función a las reuniones

que se dieron se ha definido el problema a dar solución con prioridad y las limitaciones del sistema a implementar.

Historias de Usuario

Como parte del análisis del sistema, se tienen las Historias de Usuario, que fueron hechas y sugeridas por la parte del responsable de la Unidad de Visación conforme lo señala XP. Son el primer paso para obtener los requerimientos de software, están escritas en palabras propias del cliente y son la base para estimar los tiempos de implementación.

Tabla 2: Historia de Usuario Registrar Datos de Acta de Notas

Historia de Usuario	
Número: 1	Nombre Historia: Registrar Datos de Acta de Notas
Usuario: Responsable de la Unidad de Visación	
Modificación de Historia Número: 1	Iteración Asignada: 1
Prioridad en Negocio: Alta	Puntos Estimados: 6
Riesgo en Desarrollo: Alta	Puntos Reales: 6
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se introducen los datos del Acta de Notas (Ugel, Institución Educativa, Modalidad de educación, Nivel de educación, Periodo Lectivo e Imagen del acta escaneada)	
Observaciones:	

Fuente: Elaboración propia

Tabla 3: Historia de Usuario Registrar Datos de Acta de Subsanación

Historia de Usuario	
Número: 2	Nombre Historia: Registrar Datos de Acta de Sustentación
Usuario: Responsable de la Unidad de Visación	
Modificación de Historia Número: 1	Iteración Asignada: 2
Prioridad en Negocio: Alta	Puntos Estimados: 4
Riesgo en Desarrollo: Alta	Puntos Reales: 4
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se introducen los datos del Acta de Sustentación (Asignatura a la que pertenece el Acta, Institución Educativa, Nivel de educación, turno, año, sección e Imagen del acta escaneada)	
Observaciones: Las asignaturas dependen del Nivel de Educación (Primaria, Secundaria, inicial)	

Fuente: Elaboración propia

Tabla 4: Historia de Usuario Buscar Acta de Subsanación

Historia de Usuario	
Número: 3	Nombre Historia: Buscar Acta de Subsanación
Usuario: Responsable de la Unidad de Visación	
Modificación de Historia Número: 1	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Media	Puntos Reales: 1
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se debe poder buscar las Actas de Subsanación con alguno de los nombres de los estudiantes, registrado en ellas	
Observaciones:	

Fuente: Elaboración propia

Tabla 5: Historia de Usuario Buscar Acta de Notas

Historia de Usuario	
Número: 4	Nombre Historia: Buscar Acta de Notas
Usuario: Responsable de la Unidad de Visación	
Modificación de Historia Número: 1	Iteración Asignada: 3
Prioridad en Negocio: Media	Puntos Estimados: 1
Riesgo en Desarrollo: Media	Puntos Reales: 1
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Encontrar todas las Actas de Notas que contengan determinado Nombre del estudiante, tener la posibilidad de ver los datos de dichas actas encontradas	
Observaciones:	

Fuente: Elaboración propia

Tabla 6: Historia de Usuario Guardar Datos de UGEL

Historia de Usuario	
Número: 5	Nombre Historia: Guardar Datos de UGEL
Usuario: Responsable de la Unidad de Visación	
Modificación de Historia Número: 1	Iteración Asignada: 4
Prioridad en Negocio: Baja	Puntos Estimados: 0.1
Riesgo en Desarrollo: Baja	Puntos Reales: 0.2
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Guardar los datos más importantes de la UGEL (Código de la UGEL y su nombre)	
Observaciones: Solo se guardarán los Datos de la UGEL Andahuaylas	

Fuente: Elaboración propia

Tabla 7: Historia de Usuario Guardar Datos de la Institución Educativa

Historia de Usuario	
Número: 6	Nombre Historia: Guardar Datos de la Institución Educativa
Usuario: Responsable de la Unidad de Visación	
Modificación de Historia Número: 1	Iteración Asignada: 4
Prioridad en Negocio: Media	Puntos Estimados: 0.5
Riesgo en Desarrollo: Media	Puntos Reales: 2
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Guardar los datos más importantes de la Institución Educativa (Código de Institución, Departamento, provincia, distrito y centro poblado al que pertenece, nombre o número y tipo de gestión)	
Observaciones: El tipo de gestión de la Institución puede ser de dos tipos: Público o privado	

Fuente: Elaboración propia

Estimación de esfuerzo por Historia de Usuario

Se realiza a partir de las Historias de Usuario, en la parte inicial del proyecto, bajo el criterio del equipo de desarrollo y con conformidad del cliente.

Una vez declaradas todas las historias de usuario, se realizó una reunión con el equipo de desarrollo quien estimó el tiempo que tomará implementar cada historia de usuario, cabe resaltar que los tiempos declarados en esta primera parte pueden diferir de los realmente requeridos, al tratarse de una aproximación y un primer acercamiento a la duración real del proyecto. XP señala que es buena práctica realizar estas estimaciones, durante la etapa de análisis del sistema, mucho antes de iniciar el desarrollo del software, puesto que servirán para medir el progreso del proyecto y como guía a la cual ajustarse, por lo que, una vez terminado este proceso, solo se volverán a realizar nuevas estimaciones de esfuerzo si se identifican nuevas historias de usuario.

Para ver las estimaciones de esfuerzo, en el orden en el que serán desarrolladas las historias de usuario ver la Tabla 9. El equipo de desarrollo determinó otorgar más puntos de esfuerzo a la primera historia

de usuario, ya que en esa se preparará el entorno de desarrollo y se realizarán las configuraciones pertinentes, además se elegirá una primera interfaz para el sistema y se procederá con diseños clave como el de la base de datos que ha de ser diseñada de tal manera que permita cambios menores para cumplir con los requerimientos de las siguientes historias de usuario.

Tabla 8 : Estimación de esfuerzos por Historia de Usuario

Historias de Usuario	Puntos Estimados (semana/hombre)
Registrar Datos de Acta de Notas	6
Registrar Datos de Acta de Subsanación	4
Buscar Acta de Subsanación	1
Buscar Actas de Notas	1
Guardar Datos de UGEL	0.1
Guardar Datos de la Institución Educativa	0.5

Fuente: Elaboración propia

4.3.2. Diseño

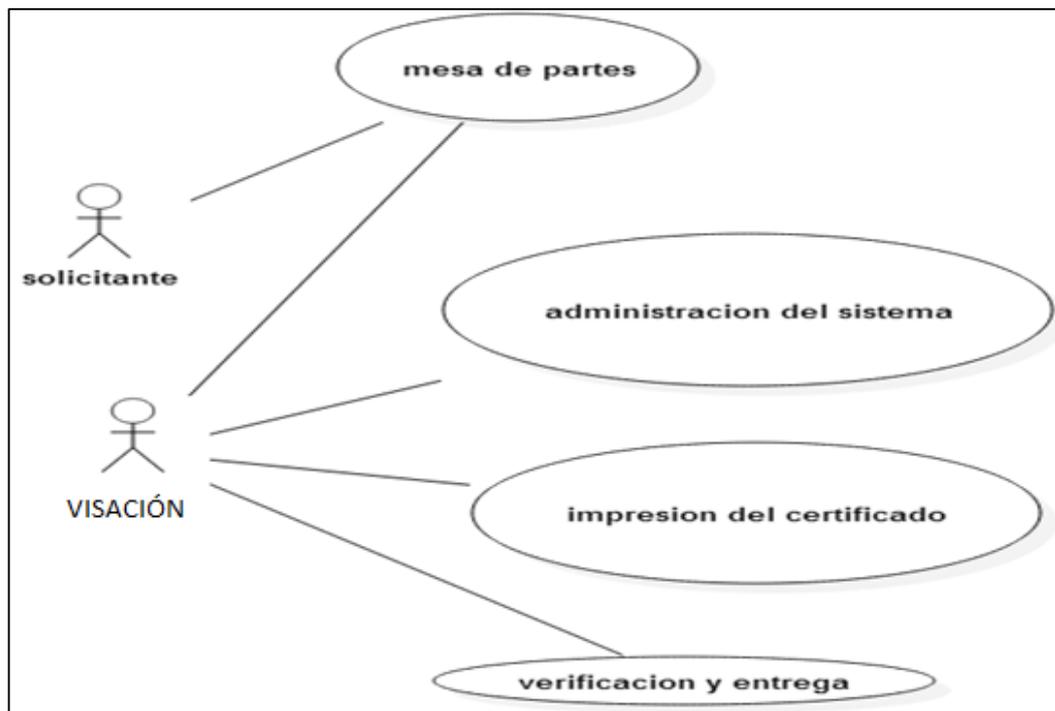
El diseño se realizó durante el desarrollo y programación del sistema de información VisaSoft, siendo constantemente revisado y modificado debido a las solicitudes del usuario presentados durante el desarrollo.

Entre los elementos más importantes que menciona XP referentes al diseño es la simplicidad. Se elaboran diseños simples ya que sirven de referencia para la implementación y empleamos el modelado UML optando la realización de algunos diagramas relevantes, también están las tarjetas CRC y el modelo Entidad Relación de la Base de Datos DBVISAssoft.

4.3.2.1. Modelo de casos de Uso del Negocio

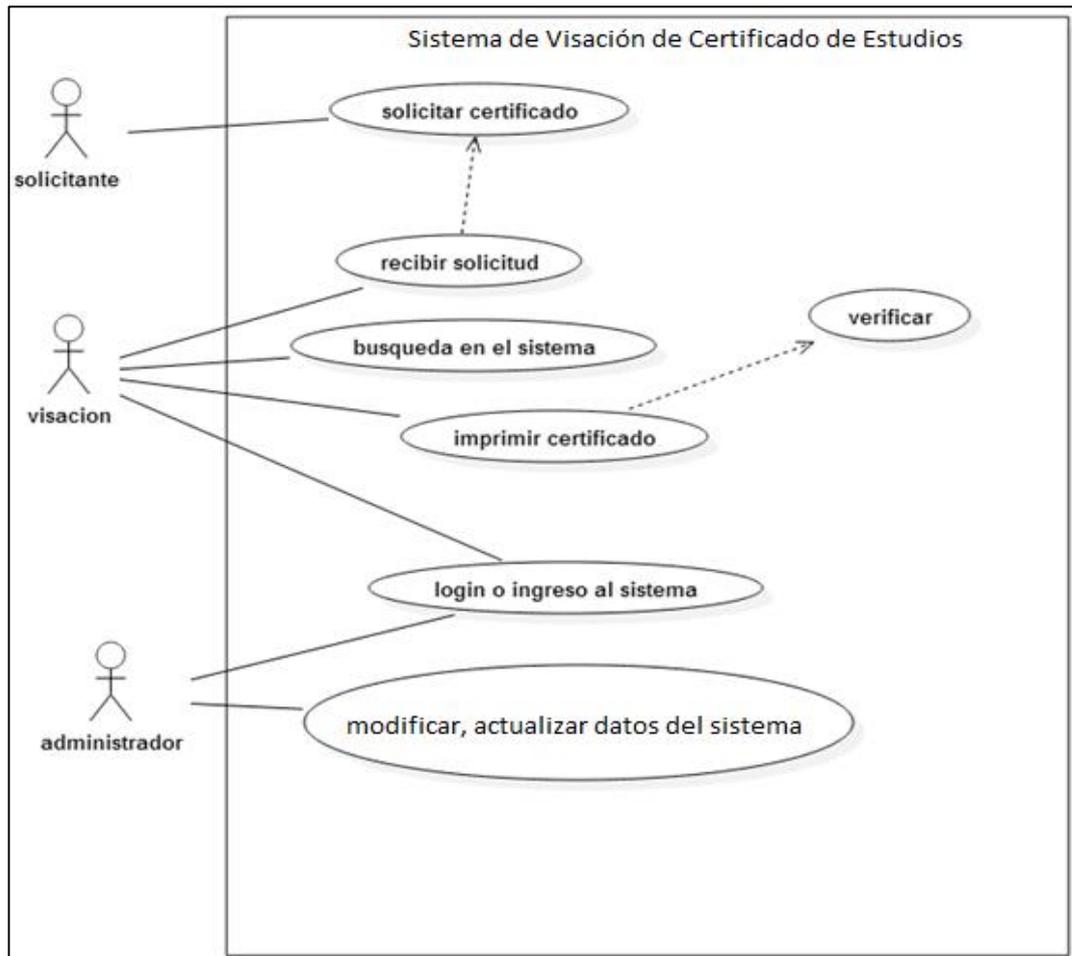
Un modelo de casos de uso de negocio describe los procesos del negocio en términos de casos de uso y actores (graficas o descripciones), para de esta forma entenderlo mejor el funcionamiento del sistema. Para ello se muestra a continuación el grafico de la visación de certificados UGEL.

Imagen 211: Modelo de Casos de Uso de Negocio Sistema VisaSoft



Fuente: Elaboración propia

Imagen 222: Modelo de Casos de Uso del sistema VisaSoft



Fuente: Elaboración propia

- **Actores de negocio**

A continuación se describen los actores de negocio encontrados en modelo de casos de usos de negocio.

Tabla 9: Descripción de los Actores de Negocio

Nombre	Descripción
Solicitante	Persona natural, jurídica, que se acerca a la entidad a realizar cualquier trámite.
Mesa de Partes	Usuario encargado de la recepción, registro y del movimiento de la solicitud.
Administrador del sistema	Usuario cuyo rol es administrar el sistema, administración técnica de la plataforma y actualización de las bases de datos.

Fuente: Elaboración propia

- **Requerimientos específicos**

A continuación se detallan los requerimientos específicos asociados al proceso de trámite documentario identificados en la recolección de datos:

A. Requerimientos funcionales

Tabla 9: Requerimientos funcionales

ID	Descripción
RF101	El sistema permitirá registrar un estudiante nuevo.
RF102	El sistema permitirá registrar las notas de un estudiante ya sea por años o promedios finales.
RF103	El sistema permitirá consultar detalles del estudiante.
RF104	El sistema permitirá modificar, actualizar y eliminar datos de la base de datos.
RF105	El sistema permitirá el acceso a la misma identificado con un usuario y una contraseña.
RF106	El sistema permitirá realizar búsquedas de estudiantes, instituciones, docentes y o directores.
RF107	El sistema permitirá registrar instituciones, docentes, usuarios o administradores del sistema.
RF108	El sistema permitirá la impresión del certificado de un estudiante seleccionado.
RF109	El sistema permitirá ver el ranquin de notas de una institución por secciones o años o grados.
RF110	El sistema permitirá mostrar y exportar reportes.
RF111	El sistema permitirá imprimir reportes.

Fuente: Elaboración propia

B. Requerimientos no funcionales

Tabla 10: Requerimientos no funcionales

ID	Descripción
RNF101	El sistema deberá funcionar de igual manera en los siguientes navegadores: Chrome, Firefox e Internet Explorer, tenerlo en cuenta la actualización de los softwares mencionados.
RNF102	El sistema deberá incluir el logotipo de la Institución así como el de su entidad reguladora máxima, UGEL.
RNF103	El sistema deberá presentar información de la manera más simple y organizada posible.
RNF104	El sistema deberá funcionar correctamente, sin caídas, fallos ni interrupciones de servicio, durante la jornada de trabajo de la Institución (8:00 a.m. – 6:00 p.m.)

Fuente: Elaboración propia

C. Requerimiento de gestión de información

Tabla 11: Requerimientos gestión de información

ID	Descripción
RG101	La información necesaria se extraerá de las bases de datos de otros sistemas de la institución y se cargarán al sistema antes de su puesta en marcha.
RG102	La información de nuevos estudiantes, empleados, instituciones, podrá ser registrada sólo por el Administrador del Sistema directamente a la Base de Datos.
RG103	En caso de olvido de la contraseña, el personal deberá contactar con el Administrador del Sistema para que le sea proporcionada.
RG104	La actualización de los datos que dependan de otros sistemas se hará periódicamente cada cierto tiempo por parte del Administrador del Sistema
RG105	El sistema no velará por los permisos de seguridad ni por la protección de la copia de seguridad de la Base de Datos del mismo a nivel de sistema operativo. Es responsabilidad del Administrador del Sistema el resguardo de dicha información para evitar usos indebidos y/o manipulados por personal no autorizado.
RG106	El sistema permitirá la impresión del certificado seleccionado de un estudiante específico, ya sea nivel primario, secundario, etc.

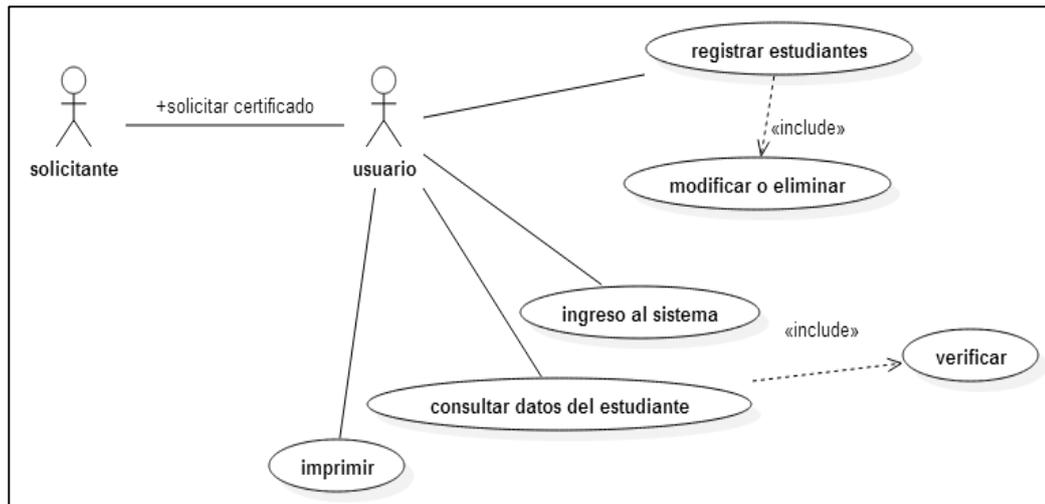
Fuente: Elaboración propia

- **Modelo de casos de uso de requerimientos**

Los modelos de casos de uso de requerimientos describen parte de la realidad en la cual se está trabajando y describen el sistema y su ambiente como entorno.

En ella se describirá la gestión de certificados.

Imagen 233: Modelo de CU de Gestión de certificados VisaSoft



Fuente: Elaboración propia

- **Matriz de trazabilidad de requerimientos funcionales vs casos de uso**

La siguiente matriz representa la correspondencia entre los requerimientos funcionales identificados y los casos de uso representados.

Tabla 12: Matriz de Trazabilidad de RF vs CU

requerimientos funcionales	El sistema permitirá registrar un estudiante nuevo.	X							X
	El sistema permitirá registrar las notas de un estudiante ya sea por años o promedios finales.	X							X
	El sistema permitirá consultar detalles del estudiante.			X	X				X
	El sistema permitirá modificar, actualizar y eliminar datos de la base de datos.						X		X
	El sistema permitirá el acceso a la misma identificado con un usuario y una contraseña.					X	X		X
	El sistema permitirá realizar búsquedas de estudiantes, instituciones, docentes y o directores.			X					X
	El sistema permitirá registrar instituciones, docentes, usuarios o administradores del sistema.	X							X
	El sistema permitirá la impresión del certificado de un estudiante seleccionado.			X					X
	El sistema permitirá ver el ranquin de notas de una institución por secciones o años o grados.							X	X
	El sistema permitirá mostrar y exportar reportes.				X			X	X
	El sistema permitirá imprimir reportes.				X				X
		solicitar certificado	recibir solicitud	busqueda de informaciones en el sistema	imprimir certificado	ingreso al sistema	modificar eliminar datos del sistema	consultar datos del estudiante	administración del sistema
casos de uso									

Fuente: Elaboración propia

- **Análisis y diseño preliminar**

El análisis y diseño preliminar es el paso intermedio entre el análisis y el diseño que consiste en librar de ambigüedades a los casos de uso identificados en la etapa anterior, plasmando explícitamente lo que se conoce explícitamente. Dicho de otra manera, el análisis y diseño preliminar consiste en hacer un primer boceto del diseño del sistema, a través de los casos de uso y el análisis de robustez. Además permite encontrar incongruencias entre los casos de uso y el modelo de dominio.

El objetivo principal de esta disciplina es servir de intermediaria entre el ¿qué va a realizar el sistema? Y el ¿cómo lo va a realizar el sistema? Los principales objetivos de esta disciplina son:

- Definir los pasos para realizar cada caso de uso.
- Definir un conjunto de artefactos (análisis de robustez) que serán usados posteriormente en el diseño detallado.
- Tener un mejor entendimiento acerca de qué va a realizar el sistema y un primer vistazo a cómo lo va a realizar.

A continuación se muestran los resultados del Análisis y Diseño preliminar realizado:

- **Especificación de casos de uso de requerimientos**

Las especificaciones de casos de uso tiene como finalidad la desambiguación de éstos últimos, permitiendo al equipo de trabajo, entender qué realiza exactamente un determinado caso de uso, identificando para ello: actores, flujo básico y alterno; y pre y post condiciones.

Gestión de expedientes

Tabla 13: Especificación de CU-001

CU-001	Solicitar visacion certificado
Actor	Actor 1: solicitante (una persona natural)
Descripción	El caso de uso permite al usuario de Mesa de Partes decepcionar la solicitud.
Flujo básico	El caso de uso inicia cuando el solicitante presenta documentos para solicitar el certificado
Flujos alternos	
Pre-condiciones	
Post-condiciones	

Fuente: Elaboración propia

Tabla 14: Especificación de CU-002

CU-002	Recibir solicitud
Actor	Actor 1: área de visacion
Descripción	El caso de uso permite al usuario manipular el documento recibido
Flujo básico	El caso de uso inicia cuando el usuario recibe ya el documento pidiendo el certificado.
Flujos alternos	
Pre-condiciones	Verificar los datos del solicitante.
Post-condiciones	

Fuente: Elaboración propia

Tabla 15: Especificación de CU-003

CU-003	Búsqueda en el sistema actas de notas
Actor	Actor 1: área de visacion
Descripción	El caso de uso permite al usuario de encargado de la vizacion hacer búsquedas de los datos del solicitante, para de esta forma contrastarlo.
Flujo básico	El caso de uso inicia cuando el usuario encuentra el estudiante en el sistema.
Flujos alternos	
Pre-condiciones	El usuario debe haber accedido buscado al estudiante o solicitante a seleccionar.
Post-condiciones	El sistema debe mostrar en pantalla las actas de notas pertenecientes al estudiante seleccionado.

Fuente: Elaboración propia

Tabla 16: Especificación de CU-004

CU-004	verificar
Actor	Actor 1: área de visacion
Descripción	El caso de uso permite al usuario verificar los datos del documento solicitado con los resultados del sistema.
Flujo básico	El caso de uso inicia cuando el usuario encuentra los datos necesarios del estudiante en el sistema.
Flujos alternos	1. Estudiante no se encuentra registrado El sistema muestra un indicando que no existen coincidencias en la base de datos del estudiante buscado.
Pre-condiciones	El usuario debe haber accedido a la página de Búsqueda.
Post-condiciones	

Fuente: Elaboración propia

Tabla 17: Especificación de CU-005

CU-005	Ingreso al sistema
Actor	Actor 1: Usuario, administrador.
Descripción	El caso de uso permite al usuario, administrador y otras personas ingresar al sistema, para ello se necesita un usuario y una contraseña.
Flujo básico	El caso de uso inicia cuando el usuario ingresa el usuario y la contraseña correctamente, pero antes tener un usuario y una contraseña.
Flujos alternos	1. Solicitante no se encuentra registrado El sistema muestra un mensaje indicando que el usuario no existe o esta desactivo.
Pre-condiciones	El usuario debe haber ingresado los datos necesarios.
Post-condiciones	El sistema debe estar seguro de los datos ingresados, y de esta forma acceder al sistema.

Fuente: Elaboración propia

Tabla 18: Especificación de CU-006

CU-006	Modificar, eliminar y actualizar
Actor	Actor 1: Usuario con permisos adecuados
Descripción	El caso de uso permite al usuario poder registrar, modificar, eliminar los datos necesarios que se tienen en el sistema.
Flujo básico	El caso de uso inicia cuando el usuario este logeado al sistema, y que tenga permisos adecuados para realizar estas operaciones.
Flujos alternos	
Pre-condiciones	El usuario deberá tener privilegios suficientes para realizar estas operaciones.
Post-condiciones	El sistema debe emitir un mensaje de confirmación como éxito o fracaso, dependiendo de la actividad realizada.

Fuente: Elaboración propia

- **Análisis de Robustez**

El análisis de robustez sirve para pasar de los casos de uso al diseño detallado (y luego al código). Para ello es necesario vincular los casos de uso a objetos. El análisis de robustez ayuda a cerrar la brecha entre el análisis y diseño a través del análisis del texto de casos de uso y la identificación de un conjunto de objetos de primera aproximación vinculados a cada caso de uso.

A. Gestión de la verificación del certificado notas

- **CU – 001: Registrar nuevos estudiantes, acta de notas, instituciones, usuarios.**
- **CU – 002: Actualizar los datos necesarios**
- **CU – 003: Consultar datos necesarios**
- **CU – 004: Buscar estudiantes, acta de notas, usuarios.**
- **CU – 005: Recepcionar solicitud. Gestión de la solicitud**
- **Verificar la existencia del usuario.**

B. Gestión de la solicitud.

- **CU – 006: buscar estudiante solicitante en el sistema.**
- **CU - 007: verificar los datos con el documento y el resultado mostrado del sistema.**
- **CU – 008: mostrar las actas de notas del estudiante.**
- **CU – 009: Concluir atención de la solicitud.**

- **Diagrama de clases**

El diagrama de clases representa las especificaciones de las clases e interfaces software en una aplicación. Entre la información general encontramos:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información acerca del tipo de los atributos.
- Navegabilidad.
- Dependencias.

A diferencia de las clases conceptuales del Modelo del Dominio, las clases del diagrama de clases muestran las definiciones de las clases de software en lugar de los conceptos del mundo real. Además de las asociaciones y atributos básicos, el diagrama se amplía para representar, ver diagrama anexo 4.

4.3.2.2. Diseño de las tarjetas CRC

Tarjetas CRC: se consideró estas tarjetas.

- Clases (Nombre de clase: define un vocabulario)
- Responsabilidades para cada clase: muestran los problemas que van a ser resueltos, mediante la utilización de una frase que inicia con un verbo activo (Una responsabilidad es algo que la clase sabe o hace, Indicadores de las acciones son los verbos).

Tabla 19: Registrar Acta

TARJETA CRC		
Numero:1	Escenario Registrar Usuario	
Nombre CRC: Registrar acta	Nombre clase: UsuarioController	
<u>Responsabilidades:</u> <ul style="list-style-type: none"> • Redirigir a la vista correspondiente. • Recibir el recuest. • Optener los datos del request. • Guardar o registrar los datos respectivos en la bd. • Redirigir a la vista Usuario. 	<u>Colaboradores:</u> <ul style="list-style-type: none"> • UsarioFormRequest. User'. • Request. • Redirect. • DB. • Auth. 	<u>Métodos</u> <ul style="list-style-type: none"> • Créate • Store
Observaciones:		

Fuente: Elaboración propia

Tabla 20: Editar Usuario

TARJETA CRC		
Numero:2	Escenario :Editar Usuario	
Nombre CRC: Editar Usuario	Nombre	clase:
	UsuarioController	
<u>Responsabilidades:</u> <ul style="list-style-type: none"> • Recibir el ID de un Usuario. • Obtener datos de dicha Usuario. • Redirigir a la vista enviando los datos correspondientes. 	<u>Colaboradores:</u> <ul style="list-style-type: none"> • UsuarioFormRequest. User. • Request. • Redirect. • DB. • Auth. 	<u>Métodos</u> <ul style="list-style-type: none"> • Edit. • Update.

<ul style="list-style-type: none"> • Recibir el recuest. • Optener los datos del request. • Actualizar datos de la bd. • Redirigir a la vista usuario. 		
Observaciones:		

Fuente: Elaboración propia

Tabla 21: Eliminar Usuario

TARJETA CRC		
Numero:3	Escenario :Eliminar Usuario	
Nombre CRC: elimina el Usuario	Nombre clase: UsuarioController	
<u>Responsabilidades:</u> Recibir el ID del Usuario. Eliminar usuario. Redirigir a la vista Usuario.	<u>Colaboradores:</u> UsuarioFormRequest. User. Request. Redirect. DB. Auth.	<u>Métodos</u> Destroy.
Observaciones:		

Fuente: Elaboración propia

Tabla 22: Buscar Estudiante

TARJETA CRC		
Numero:4	Escenario :buscar Usuario	
Nombre CRC: Buscar Usuario	Nombre clase: UsuarioController	
<u>Responsabilidades:</u> Recibir request de la vista Usuario. Solicitar los datos de la bd según el request, Redirigir a la vista enviando los datos correspondientes.	<u>Colaboradores:</u> UsuarioFormReq uest. User. Request. Redirect. DB. Auth.	<u>Métodos</u> Index.

Observaciones:

Fuente: Elaboración propia

Tabla 23: Registrar Estudiante

TARJETA CRC		
Numero:5	Escenario: Registrar acta	
Nombre CRC: Registrar acta	Nombre clase: ActaController	
<u>Responsabilidades:</u> Optener datos heredados de otras tablas. Y enviarlo a la vista. Recibir el recuest. Optener los datos del request. Guardar copia del doc, escaneado. Guardar o registrar de los datos respectivos en la bd. Redirigir a la vista acta.	<u>Colaboradores:</u> Actafrom request. Acta. Request. Redirect. DB. Auth. Input.	<u>Métodos</u> Créate Store
Observaciones:		

Fuente: Elaboración propia

Imagen 24: Editar Acta

TARJETA CRC	
Numero:6	Escenario :EditarActa
Nombre CRC: editar el Acta	Nombre clase: ActaController

<u>Responsabilidades:</u> Recibir el ID de una acta. Obtener datos de dicha Acta. Redirigir a la vista enviando los datos correspondientes. Recibir el recuest. Obtener los datos del request. Guardar copia del nuevo doc escaneado. Actualizar datos de la bd. Guardar o registrar los datos respectivos en la bd. Redirigir a la vista acta.	<u>Colaboradores:</u> Actafrom request. Acta. Request. Redirect. DB. Auth. Input	<u>Métodos</u> Edit. Update.
Observaciones:		

Fuente: Elaboración propia

Tabla 24: Eliminar Acta

TARJETA CRC		
Numero:7	Escenario: Eliminar Acta	
Nombre CRC: Eliminar el Acta	Nombre clase: ActaController	
<u>Responsabilidades:</u> Recibir el ID de una acta. Eliminar el acta respectiva. Redirigir a la vista acta.	<u>Colaboradores:</u> Actafrom request. Acta. Request. Redirect. DB. Auth. Input	<u>Métodos</u> Destroy.
Observaciones:		

Fuente: Elaboración propia

Tabla 25: Buscar Acta

TARJETA CRC		
Numero:8	Escenario :buscarActa	
Nombre CRC: buscar acta	Nombre clase: ActaController	
<u>Responsabilidades:</u> Recibir el request de acta. Solicitar los datos de la bd según el request, Redirigir a la vista enviando los datos correspondientes.	<u>Colaboradores:</u> Actafrom request. Acta. Request. Redirect. DB. Auth. Input	<u>Métodos</u> Index.
Observaciones:		

Fuente: Elaboración propia

Tabla 26: Acta Estudiante

TARJETA CRC		
Numero:9	Escenario: asignar estudiante al acta.	
Nombre CRC: Acta estudiante	Nombre clase: ActaEstudiante controller	
<u>Responsabilidades:</u> Optener datos heredados de la tabla actas. Y enviarlo a la vista. Recibir el recuest. Optener los datos del request. Guardar o registrar de los datos respectivos en la bd. Redirigir a la correspondiente vista.	<u>Colaboradores:</u> ActaEstudiantefrom Request. ActaEstudiante. Request. Redirect. DB. Auth.	<u>Métodos</u> Créate Store
Observaciones:		

Fuente: Elaboración propia

Tabla 27: Editar Estudiante

TARJETA CRC		
Numero:10	Escenario :Editar Estudiante	
Nombre CRC: editar estudiante	Nombre	clase:
	ActaEstudianteController	
<u>Responsabilidades:</u> Recibir el ID de una acta. Optener datos de la tabla Acta_Estudiante. Redirigir a la vista enviando los datos correspondientes. Recibir el request. Optener los datos del request. Actualizar datos de la bd. Guardar o registrar los datos respectivos en la bd. Redirigir a la vista acta_Estudiante	<u>Colaboradores:</u> ActaEstudiantefrom request. ActaEstudiante. Request. Redirect. DB. Auth.	<u>Métodos</u> Edit. Update.
Observaciones:		

Fuente: Elaboración propia

Tabla 28: Eliminar Estudiante

TARJETA CRC		
Numero:11	Escenario :EliminarEstudiante	
Nombre CRC: Eliminar estudiante	Nombre	clase:
	ActaController	
<u>Responsabilidades:</u> Recibir el ID de una acta. Eliminar el acta respectiva. Redirigir a la vista acta.	<u>Colaboradores:</u> ActaEstudiantefrom request. ActaEstudi9999ante. Request. Redirect. DB. Auth.	<u>Métodos</u> Destroy.
Observaciones:		

Fuente: Elaboración propia

Tabla 29: Buscar Estudiante

TARJETA CRC		
Numero:12	Escenario :buscarestudiente	
Nombre CRC: buscar estudiante	Nombre clase: ActaController	
<u>Responsabilidades:</u> Recibir el request del acta. Solicitar los datos de la bd según el request, Redirigir a la vista enviando los datos correspondientes.	<u>Colaboradores:</u> Actafrom request. Acta	<u>Métodos</u> Index. Show.
Observaciones:		

Fuente: Elaboración propia

Tabla 30: Reporte Actas

TARJETA CRC		
Numero:13	Escenario Reporte de Actas	
Nombre CRC: reporte de actas	Nombre clase: ReporteController	
<u>Responsabilidades:</u> Recibir el recuest y Optener los parámetros de búsqueda. Solicitar los datos respectivos de acuerdo a los parámetros de búsqueda. Mostar las actas de acuerdo a parámetros de búsqueda. Redirigir a la vista reportes.	<u>Colaboradores:</u> Request. Redirect. DB. Auth.	<u>Métodos</u> Index.
Observaciones:		

Fuente: Elaboración propia

Esquema de la base de datos

El esquema de base de datos describe la estructura de una base de datos, en un lenguaje formal soportado por un sistema de gestión de base de datos. En una base de datos relacional, el esquema define sus tablas, sus campos en cada tabla y las relaciones entre cada campo y cada tabla. Generalmente el esquema es definido en un lenguaje de base de datos, el término se usa a menudo para referirse a una representación gráfica de la estructura de la BD. Ver diagrama entidad relación anexo 2.

4.3.3. Codificación

En las reuniones del sistema se escribieron las siguientes tareas de ingeniería.

En XP la codificación se inicia prácticamente desde el principio, favoreciendo el objetivo de estar haciendo entregas frecuentemente al cliente.

Algunos de los elementos más importantes en cuanto a la codificación son:

Tareas de ingeniería

Una Historia de Usuario se descompone en varias tareas de ingeniería, estas describen las actividades que se realizarán en cada historia de usuario y permiten un acercamiento al código.

Tabla 31 : Tarea Ingeniería Diseño Básico de la Base de Datos

TAREA DE INGENIERÍA	
Número de Tarea: 1	Número de Historia: 1 y 2
Nombre de Tarea: Diseño Básico de la Base de Datos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 05/12/2016	Fecha Fin: 14/05/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una Base de Datos que contendrá las tablas necesarias para guardar los datos de las Actas de Notas, Actas de Subsanación; las modalidades, niveles, grados, asignaturas y turnos.	

Fuente: Elaboración propia

Tabla 32: Diseño de la interfaz para las modalidades

TAREA DE INGENIERÍA	
Número de Tarea: 2	Número de Historia: 1
Nombre de Tarea: Diseño de la interfaz para las modalidades	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.8
Fecha Inicio: 08/12/2017	Fecha Fin: 12/12/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá registrar, modificar y eliminar los datos de las modalidades	

Fuente: Elaboración propia

Tabla 33 Tarea Ingeniería Diseño de la interfaz para los niveles

TAREA DE INGENIERÍA	
Número de Tarea: 3	Número de Historia: 1 y 2
Nombre de Tarea: Diseño de la interfaz para los niveles	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.8
Fecha Inicio: 13/12/2016	Fecha Fin: 27/12/2016
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá registrar, modificar y eliminar los distintos niveles de educación	

Fuente: Elaboración propia

Tabla 34: Tarea Ingeniería Diseño de la interfaz para los grados

TAREA DE INGENIERÍA	
Número de Tarea: 4	Número de Historia: 1 y 2
Nombre de Tarea: Diseño de la interfaz para los grados	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.8
Fecha Inicio: 28/12/2017	Fecha Fin: 06/01/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá registrar, modificar y eliminar los distintos grados, que dependerán de un nivel	

Fuente: Elaboración propia

Tabla 35 Diseño de la interfaz para registrar Acta de Notas

TAREA DE INGENIERÍA	
Número de Tarea: 5	Número de Historia: 1
Nombre de Tarea: Diseño de la Interfaz para Registrar Acta de Notas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 09/01/2017	Fecha Fin: 02/02/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá registrar, modificar y eliminar los datos pertinentes a un Acta de Notas	

Fuente: Elaboración propia

Tabla 36 : Tarea Ingeniería Diseño de la interfaz para los grados

TAREA DE INGENIERÍA	
Número de Tarea: 6	Número de Historia: 1 y 2
Nombre de Tarea: Diseño de la interfaz para los grados	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.8
Fecha Inicio: 03/02/2017	Fecha Fin: 13/02/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá registrar, modificar y eliminar los distintos grados, que dependerán de un nivel	

Fuente: Elaboración propia

Tabla 37: Tarea Ingeniería Diseño de la interfaz para las asignaturas

TAREA DE INGENIERÍA	
Número de Tarea: 7	Número de Historia: 2
Nombre de Tarea: Diseño de la interfaz para las asignaturas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.8
Fecha Inicio: 15/02/2017	Fecha Fin: 28/02/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá registrar, modificar y eliminar las distintas asignaturas, que dependerán de un nivel	

Fuente: Elaboración propia

Tabla 38 : Tarea Ingeniería Diseño de la interfaz para los turnos

TAREA DE INGENIERÍA	
Número de Tarea: 8	Número de Historia: 2
Nombre de Tarea: Diseño de la interfaz para los turnos	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.8
Fecha Inicio: 01/03/2017	Fecha Fin: 07/03/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá registrar, modificar y eliminar los distintos turnos que existen	

Fuente: Elaboración propia

Tabla 39: Tarea Ingeniería Diseño de la interfaz para registrar Acta de Subsanación

TAREA DE INGENIERÍA	
Número de Tarea: 9	Número de Historia: 2
Nombre de Tarea: Diseño de la interfaz para registrar Acta de Subsanación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 2
Fecha Inicio: 08/03/2017	Fecha Fin: 17/03/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá registrar los datos pertinentes del Acta de Notas	

Fuente: Elaboración propia

Tabla 40 : Tarea Ingeniería Diseño de la interfaz para buscar Acta de Subsanación

TAREA DE INGENIERÍA	
Número de Tarea: 10	Número de Historia: 3
Nombre de Tarea: Diseño de la interfaz para buscar Acta de Subsanación	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 20/03/2017	Fecha Fin: 04/04/2017
Programador Responsable: Milagros Candi Pérez Torres	

Descripción: Se diseñará una página que permitirá buscar y mostrar las Actas de Subsanción de acuerdo al nombre de estudiante

Fuente: Elaboración propia

Tabla 41: Tarea Ingeniería Diseño de la interfaz para buscar Acta de Notas

TAREA DE INGENIERÍA	
Número de Tarea: 11	Número de Historia: 4
Nombre de Tarea: Diseño de la interfaz para buscar Acta de Notas	
Tipo de Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 05/04/2017	Fecha Fin: 21/04/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá buscar y mostrar las Actas de Notas de acuerdo al nombre de estudiante	

Fuente: Elaboración propia

Tabla 42: Tarea Ingeniería Diseño de la interfaz para registrar datos de UGEL

TAREA DE INGENIERÍA	
Número de Tarea: 12	Número de Historia: 5
Nombre de Tarea: Diseño de la interfaz para registrar datos de UGEL	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.1
Fecha Inicio: 24/04/2017	Fecha Fin: 02/05/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá ingresar, modificar y eliminar los datos de una UGEL	

Fuente: Elaboración propia

Tabla 43: Tarea Ingeniería Diseño de la interfaz para registrar datos de la Institución Educativa

TAREA DE INGENIERÍA	
Número de Tarea: 13	Número de Historia: 6
Nombre de Tarea: Diseño de la interfaz para registrar datos de la Institución Educativa	
Tipo de Tarea: Desarrollo	Puntos Estimados: 0.5
Fecha Inicio: 03/05/2017	Fecha Fin: 15/05/2017
Programador Responsable: Milagros Candi Pérez Torres	
Descripción: Se diseñará una página que permitirá ingresar, modificar y eliminar los datos de una Institución educativa	

Fuente: Elaboración propia

4.3.4. Pruebas

Pruebas funcionales

Son pruebas que se crean a partir de las historias de usuario. En ellas se especifican los escenarios, desde la perspectiva del cliente, para probar que una historia de usuario ha sido implementada correctamente, garantizando que los requerimientos fueron cumplidos y que el sistema es aceptable.

La historia de usuario solo se considerará completa hasta que ha pasado por las pruebas funcionales, las mismas que se muestran a continuación.

Tabla 44: Prueba Funcional Registro de Datos de Acta Notas

Prueba Funcional	
Número Prueba: 1	Numero de Historia de Usuario: 1
Nombre de la Prueba: Verificación de registro de Datos de Acta Notas	
Descripción: El usuario, al entrar en el sistema verá una pantalla con un botón nuevo que lo redirigirá a otra ventana donde podrá ingresar y seleccionar los datos, para una nueva Acta de Notas, si los datos ingresados cumplen con el formato correcto serán guardados cuando el usuario presione el botón guardar e inmediatamente será redirigido a otra ventana donde observará los datos registrados.	
Condiciones de ejecución: Ninguna	
Entrada: El usuario entra en el sistema de información. Se muestra un botón nuevo. El usuario presiona en el botón y es redirigido a una nueva ventana. En la nueva ventana el usuario ingresa los datos pertinentes a una nueva Acta de Notas. El usuario presiona en el botón “Guardar” de dicha ventana y el sistema comprueba el formato de los datos. El sistema redirige al usuario a otra ventana donde verá los datos ingresados ya guardados.	
Resultado esperado: El usuario puede guardar nuevas actas de notas	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 45: Prueba Funcional Registro de Acta de Subsanación

Prueba Funcional	
Número Prueba: 2	Numero de Historia de Usuario: 2
Nombre de la Prueba: Verificación de registro de Datos de Acta de Subsanación	
Descripción: El usuario, al entrar en el sistema verá una pantalla con un botón nuevo que lo redirigirá a otra ventana donde podrá ingresar y seleccionar los datos de para una nueva Acta de Subsanación, si los datos ingresados cumplen con el formato correcto serán guardados cuando el usuario presione el botón guardar e inmediatamente será redirigido a otra ventana donde observará los datos registrados.	
Condiciones de ejecución: Ninguna	
<p>Entrada:</p> <p>El usuario entra en el sistema de información.</p> <p>Elige el menú Acta Subsanación.</p> <p>Se muestra un botón nuevo.</p> <p>El usuario presiona en el botón y es redirigido a una nueva ventana.</p> <p>En la nueva ventana el usuario ingresa los datos pertinentes a una nueva Acta de Subsanación.</p> <p>El usuario presiona en el botón “Guardar” de dicha ventana y el sistema comprueba el formato de los datos.</p> <p>El sistema redirige al usuario a otra ventana donde verá los datos ingresados ya guardados.</p>	
Resultado esperado: El usuario puede guardar nuevas actas de Subsanación	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 46: Prueba Funcional Búsqueda de Acta de Notas

Prueba Funcional	
Número Prueba: 3	Numero de Historia de Usuario: 3
Nombre de la Prueba: Verificación de búsqueda de Acta de Notas	
Descripción: El usuario, al entrar en el sistema verá una pantalla con un campo de texto, en el cual deberá teclear un código de estudiante y la presionar ENTER o en el botón BUSCAR el sistema mostrará las actas que coincidan con dicho código.	
Condiciones de ejecución: Ninguna	
<p>Entrada:</p> <p>El usuario entra en el sistema de información.</p> <p>Se muestra un campo de texto.</p> <p>El usuario escribe en el campo de texto un código de estudiante.</p> <p>Presiona ENTER o en el botón BUSCAR.</p> <p>El sistema tomará el código ingresado y buscará en la base de datos.</p> <p>El sistema mostrará en pantalla todas las actas que coinciden con dicho código de estudiante.</p>	
Resultado esperado: El usuario puede buscar actas de notas con un código.	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 47: Prueba Funcional Búsqueda de Acta de Sustentación

Prueba Funcional	
Número Prueba: 4	Numero de Historia de Usuario: 4
Nombre de la Prueba: Verificación de búsqueda de Acta de Subsanación	
Descripción: El usuario, al entrar en el sistema verá una pantalla con un campo de texto, en el cual deberá teclear un nombre de estudiante y la presionar ENTER o en el botón BUSCAR el sistema mostrará las actas que coincidan con dicho nombre.	
Condiciones de ejecución: Ninguna	
<p>Entrada:</p> <p>El usuario entra en el sistema de información.</p> <p>Elegirá el menú Acta de Subsanación.</p> <p>Se muestra un campo de texto.</p> <p>El usuario escribe en el campo de texto un nombre de estudiante.</p> <p>Presiona ENTER o en el botón BUSCAR.</p> <p>El sistema tomará el nombre ingresado y buscará en la base de datos.</p> <p>El sistema mostrará en pantalla todas las actas que coinciden con dicho nombre de estudiante.</p>	
Resultado esperado: El usuario puede buscar actas de Subsanación con un nombre.	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 48: Prueba Funcional Verificar Registro de datos de UGEL

Prueba Funcional	
Número Prueba: 5	Numero de Historia de Usuario: 5
Nombre de la Prueba: Verificación de registro de datos de UGEL	
Descripción: El usuario, al entrar en el sistema, elegirá la opción UGEL en el menú, presionará el botón NUEVO y aparecerá una pantalla para que registre los datos de una UGEL y los guarde.	
Condiciones de ejecución: Ninguna	
<p>Entrada:</p> <p>El usuario entra en el sistema de información.</p> <p>Elegirá el menú UGEL y el sistema muestra una nueva pantalla.</p> <p>Presiona en el botón NUEVO y el sistema cargará un formulario.</p> <p>El usuario digita los datos de una UGEL en el formulario.</p> <p>Presiona el botón GUARDAR.</p> <p>El sistema verifica los datos ingresados.</p> <p>Si los datos cumplen con el formato establecido se mostrara otra pantalla con los datos ya guardados.</p>	
Resultado esperado: El usuario puede guardar datos de UGEL .	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: Elaboración propia

Tabla 49: Prueba Funcional Verificar Registro de Institución Educativa

Prueba Funcional	
Número Prueba: 6	Numero de Historia de Usuario: 6
Nombre de la Prueba: Verificación de registro de Institución Educativa	
Descripción: El usuario, al entrar en el sistema, elegirá la opción Institución Educativa en el menú, presionará el botón NUEVO y aparecerá una pantalla para que registre los datos de una Institución Educativa y los guarde.	
Condiciones de ejecución: Ninguna	
<p>Entrada:</p> <p>El usuario entra en el sistema de información.</p> <p>Elegirá el menú Institución Educativa y el sistema muestra una nueva pantalla.</p> <p>Presiona en el botón NUEVO y el sistema cargará un formulario.</p> <p>El usuario digita los datos de una Institución Educativa en el formulario.</p> <p>Presiona el botón GUARDAR.</p> <p>El sistema verifica los datos ingresados.</p> <p>Si los datos cumplen con el formato establecido se mostrará otra pantalla con los datos ya guardados.</p>	
Resultado esperado: El usuario puede guardar datos de UGEL.	
Evaluación de la prueba: Prueba satisfactoria.	

Fuente: Elaboración propia

CAPÍTULO V

EVALUACION DE LA SOLUCIÓN

En este capítulo se analiza los resultados alcanzados en función de los objetivos

5.1. APLICACIÓN DE TÉCNICAS DE INVESTIGACIÓN

Recolección de información: Para la recolección de la información y evaluar el proceso de verificación y, analizar el tiempo del proceso se realizó utilizando ficha de observación.

Planteamiento de soluciones: Para el planteamiento de la solución se dio por conveniente desarrollar el sistema de información para generar certificado de estudios en el nivel básico regular de la UGEL Andahuaylas. Utilizando la metodologías de desarrollo XP.

Análisis de resultados: Para el procesamiento de datos y el análisis de resultados se ha utilizado el programa ofimático Microsoft Excel 2013 y Cuadros de fichas utilizados.

Ficha de observación: para el tiempo de del proceso de verificación. La ficha nos permite obtener los tiempos que toma dicho proceso, los cuales son comparados con la ficha pre-prueba y post- prueba con la finalidad de conocer el tiempo que toma el proceso que se realiza actualmente, con el tiempo al haber implantado el sistema de información.

5.2. CONSTATACIÓN DE OBJETIVOS

Determinar la metodología y sus procesos en el desarrollo del Sistema de Información para Generar Certificado de Estudios del Nivel Básico Regular de la UGEL Andahuaylas

La metodología de desarrollo del Sistema de Información es la Programación Extrema (XP), siendo especialmente útil para proyectos de software de pequeña y mediana envergadura, pequeña (para el caso los involucrados en el sistema fueron siete), donde el equipo de desarrollo es reducido, no mayor a 20 personas, además es requisito indispensable que el área usuaria del software esté involucrada en todas las fases de desarrollo.

Los procesos de XP fueron ejecutados de tal manera que culminaron con la entrega de un Sistema de Información VisaSoft, que cumple con todas las especificaciones del usuario.

Implementar el Sistema de Información para reducir el proceso de verificación de notas del certificado de estudios del nivel básico regular en la UGEL Andahuaylas.

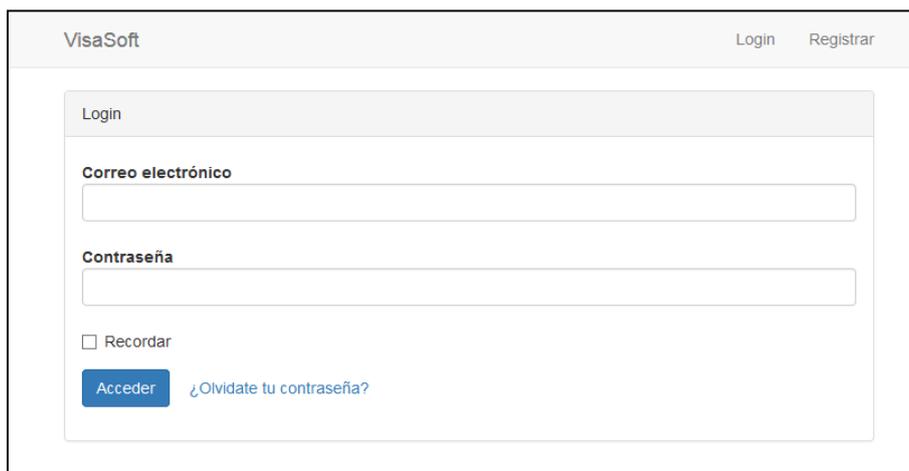
El Sistema de Información desarrollado se implementó el mismo que consta de los módulos (registrar actas, registrar colegios, reportes), los cuales están asociados con los nombres de los estudiantes a sus respectivas actas de notas escaneadas, por lo tanto se realiza más rápida el proceso de verificación de notas; confiable al tener las actas de notas escaneadas.

Implementación de la Solución

- **INTERFACE DEL SISTEMA DE INFORMACION VISAsOFT**

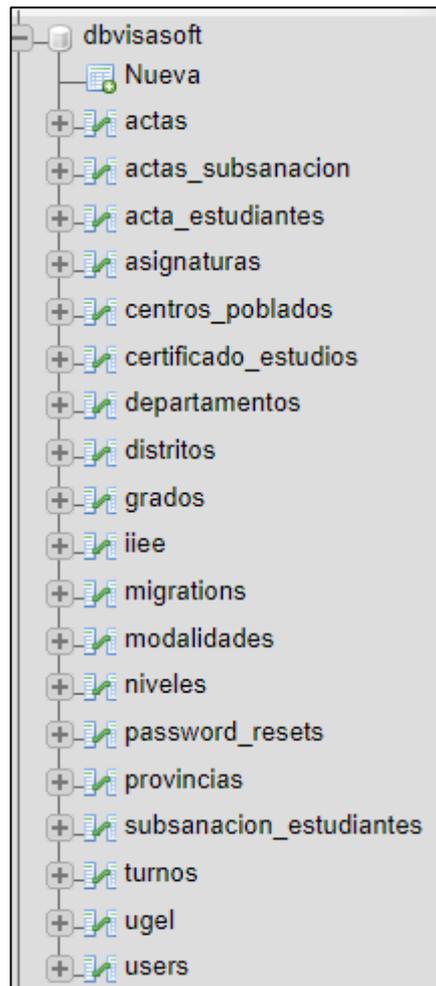
El sistema de información VISAsOFT está desarrollado con el framework laravel y se realizó la programación en el lenguaje de programación en PHP utilizando el editor ATOM.

Imagen 25: Login de VisaSoft



Fuente: Elaboración propia

Imagen 26: Tablas de la BD de VISAsoft



Fuente: Elaboración propia

Imagen 27: Crear Nueva Acta

Fuente: Elaboración propia

Imagen 28: Detalle Acta

Id Acta	Estudiante
33	TRASMONTE CONDOR Mariel Yoli Felicia
32	SERRANO ALLAUCA Jorge Luis

Fuente: Elaboración propia

Imagen 29: Reportes Actas

Id	Cod. Ugel	I.E.	Modalidad	Grado	Periodo	Escaneado	Opciones
1	UGEL Andahuaylas	19 NIÑO JESUS DE TALAVERA	EDUCACIÓN BÁSICA REGULAR	3	2013	ACTA	Detalles
2	UGEL Andahuaylas	19 NIÑO JESUS DE TALAVERA	EDUCACIÓN BÁSICA REGULAR	4	2011	ACTA	Detalles
3	UGEL Andahuaylas	19 NIÑO JESUS DE TALAVERA	EDUCACIÓN BÁSICA REGULAR	4	2014	ACTA	Detalles
4	UGEL Andahuaylas	19 NIÑO JESUS DE TALAVERA	EDUCACIÓN BÁSICA REGULAR	5	2012	ACTA	Detalles
5	UGEL Andahuaylas	19 NIÑO JESUS DE TALAVERA	EDUCACIÓN BÁSICA REGULAR	3	2013	ACTA	Detalles

Fuente: Elaboración propia

Reducir el tiempo del proceso de verificación de notas para generar certificado de estudios del nivel básico regular en la UGEL Andahuaylas.

El Sistema de Información desarrollado reduce el tiempo en el proceso de verificación de notas. Se realizó la post-prueba en el proceso de verificación de notas y se redujo este proceso en promedio de 9 minutos por solicitud, en comparación a lo que demora lo convencional de 11 horas con 2 minutos promedio; obteniéndose una reducción promedio del 98% por tanto se hace más eficiente y eficaz el trabajo del personal en la Unidad de Visación.

Cuadro de fichas usadas

Las fichas nos permite la obtención de los tiempos que toma para cada solicitud, de los cuales se analizan para conocer el tiempo promedio que se emplea esta actividad en referencia al tiempo con el implantado del sistema de información VISAssoft. En la colección de la información requerida sobre la solicitud, se uso ficha de observación.

Para el tomado de las muestras fue intencional. Donde se tomó la muestra en una semana.

Tabla 50: Presentación de Solicitud del Certificado de Estudios – PRE.PRUEBA

Nro	Fecha	Nombre Interesado	hora	IDENTIFI CADOR
1	16/01/2017	ELSA RODRIGUES VALVOA	10:35 a.m.	0001
2	16/01/2017	EBETH RODAS FIERRO	11:35 a.m.	0002
3	16/01/2017	MARGOTH LEGUIA BERROCAL	12:35 p.m.	0003
4	16/01/2017	NELIDA FARFAN CASAS	3.15 p.m	0004
5	16/01/2017	GILBERTO SOTOMAYOR ORALES	2:35 p.m.	0005
6	16/01/2017	EDGAR AYQUIPA TAIPE	3:35 p.m.	0006
7	17/01/2017	MIRIAM CANALES LICLA	9:35 a.m.	0007
8	17/01/2017	CESAR CHACON FLORES	9:35 a.m.	0008
9	17/01/2017	DEAN POLANCO RAMIRES	9:45 a.m.	0009
10	17/01/2017	JUAN SACCACO QUISPE	10:20 a.m.	0010
11	17/01/2017	JHANET ALARCON CUSI	10:33 a.m.	0011
12	17/01/2017	LIDIA GUSMN CHAVES	1105 a.m.	0012
13	17/01/2017	JUAN JIMENES CONTRERAS	11:15 a.m.	0013
14	17/01/2017	FRESIA ÑAHUIMPACHA BECERRA	11:32 a.m.	0014
15	17/01/2017	LIDIA GUSMAN CHAVES	12:00 p.m.	0015

Fuente: Elaboración propia

Tabla 51: Inicio de Búsqueda de Notas en las Actas – PRE.PRUEBA

Nr o	Fecha	Nombre Interesado	hora	IDENTIFI CADOR
1	16/01/2017	ELSA RODRIGUES VALVOA	10:38 a.m.	0001
2	16/01/2017	EBETH RODAS FIERRO	03:20 p.m.	0002
3	17/01/2017	MARGOTH LEGUIA BERROCAL	10:15 a.m.	0003
4	17/01/2017	NELIDA FARFAN CASAS	03.15 p.m.	0004
5	17/01/2017	GILBERTO SOTOMAYOR MORALES	05:35 p.m.	0005
6	18/01/2017	EDGAR AYQUIPA TAIPE	08:14 a.m.	0006
7	18/01/2017	MIRIAM CANALES LICLA	09:35 a.m.	0007
8	18/01/2017	CESAR CHACON FLORES	11:55 a.m.	0008
9	18/01/2017	DEAN POLANCO RAMIRES	03:35 p.m.	0009
10	18/01/2017	JUAN SACCACO QUISPE	05:45 p.m.	0010
11	19/01/2017	JHANET ALARCON CUSI	08:35 a.m.	0011

12	19/01/2017	LIDIA GUSMÁN CHAVES	10:55 a.m.	0012
13	20/01/2017	JUAN JIMENES CONTRERAS	04:35 p.m.	0013
14	23/01/2017	FRESIA ÑAHUIMPACHA BECERRA	11:35 a.m.	0014
15	23/01/2017	LIDIA GUSMÁN CHAVES	04:53 p.m.	0015

Fuente: Elaboración propia

Tabla 52: Finaliza la búsqueda de las notas en las actas –PRE.PRUEBA

Nro	Fecha	Nombre Interesado	hora	IDENTIFICADOR
1	16/01/2017	ELSA RODRIGUES VALVOA	03:10 p.m.	0001
2	17/01/2017	EBETH RODAS FIERRO	10:13 a.m.	0002
3	17/01/2017	MARGOTH LEGUIA BERROCAL	03.15 p.m.	0003
4	18/01/2017	NELIDA FARFAN CASAS	05:30 p.m.	0004
5	19/01/2017	GILBERT SOTOMAYOR MORALES	08:10 a.m.	0005
6	20/01/2017	EDGAR AYQUIPA TAIPA	09:33 a.m.	0006
7	20/01/2017	MIRIAM CANALES LICLA	11:55 a.m.	0007
8	20/01/2017	CESAR CHACON FLORES	03:32 p.m.	0008
9	23/01/2017	DEAN POLANCO RAMIRES	09:15 a.m.	0009
10	23/01/2017	JUAN SACCACO QUISPE	04:35 p.m.	0010
11	24/01/2017	JHANET ALARCON CUSI	10:55 a.m.	0011
12	24/01/2017	LIDIA GUSMÁN CHAVES	12:35 p.m.	0012
13	24/01/2017	JUAN JIMENES CONTRERAS	04:33 p.m.	0013
14	25/01/2017	FRESIA ÑAHUIMPACHA BECERRA	08:53 a.m.	0014
15	25/01/2017	LIDIA GUSMÁN CHAVES	04:43 p.m.	0015

Fuente: elaboración propia

Tabla 53: Presentación de Solicitud del Certificado de Estudios – POST.PRUEBA

Nro	Fecha	Nombre Interesado	hora	IDENTIFICADOR
1	1/06/2017	CARRION GONZALES JHIM JHONEL	09:05 a.m.	1
2	1/06/2017	MOREYRA CARHUAS CEYLI FLOR	11:35 a.m.	2
3	1/06/2017	ALATA ACUÑA RUTH YANETH	11.40 a.m.	3

4	1/06/2017	ALTAMIRANO PECEROS,JESCY G	11.42 a.m.	4
5	1/06/2017	CHIRCCA OCAMPO, Elida	11.43 a.m.	5
6	2/06/2017	MAÑUICO CUEVAS, Magaly Marili	3:35 p.m.	6
7	2/06/2017	AMABLE ATAÑO, Rosmary	3:37 p.m.	7
8	2/06/2017	CAÑARI ALLCCA, Raul	3:37 p.m.	8
9	2/06/2017	PASTOR MORENO, Yolanda	3:39 p.m.	9
10	2/06/2017	PACHECO OROSCO, Frida	3:00 p.m.	10
11	5/06/2017	PALOMINO CENTENO, Zeida	3:05 p.m.	11
12	5/06/2017	RODAS CRISOLES, Daniel	3:07 p.m.	12
13	5/06/2017	SILVA CHAVEZ, Ana Maria	3:10 p.m.	13
14	5/06/2017	HERRERA SOTO, Lenin Abel	3:12 p.m.	14
15	5/06/2017	SOTO PECEROS JENIFER	3:16 p.m.	15
16	6/06/2017	BAUTISTA HUAMAN PAUL ALDAHIR	09:00 a.m.	16
17	6/06/2017	HERMOZA ALARCON,CLAUDIA MERCEDES	09:05 a.m.	17
18	6/06/2017	RAMIREZ ALARCON RUTH ESMERALDA	09:08 a.m.	18
19	6/06/2017	PACHECO RODRIGUEZ LISSETH YASHEMY	09:09 a.m.	19
20	6/06/2017	SISA YUTO RAQUEL	09:12 a.m.	20

Fuente: elaboración propia

Tabla 54: Finaliza la búsqueda de las notas en las actas –POST.PRUEBA

Nro	Fecha	Nombre Interesado	hora	IDENTIFICADOR
1	1/06/2017	CARRION GONZALES JHIM JHONEL	09:05 a.m.	1
2	1/06/2017	MOREYRA CARHUAS CEYLI FLOR	11.36 a.m.	2
3	1/06/2017	ALATA ACUÑA RUTH YANETH	11:55 a.m.	3
4	1/06/2017	ALTAMIRANO PECEROS,JESCY GABRIELA	12:09 a.m.	4
5	1/06/2017	CHIRCCA OCAMPO, Elida	10:25 a.m.	5
6	2/06/2017	MAÑUICO CUEVAS, Magaly Marili	03:37 p.m.	6
7	2/06/2017	AMABLE ATAÑO, Rosmary	03:38 p.m.	7
8	2/06/2017	CAÑARI ALLCCA, Raul	03:48 p.m.	8
9	2/06/2017	PASTOR MORENO, Yolanda	04:00 p.m.	9

10	2/06/2017	PACHECO OROSCO, Frida	04:10 p.m.	10
11	5/06/2017	PALOMINO CENTENO, Zeida	03:10 p.m.	11
12	5/06/2017	RODAS CRISOLES, Daniel	03:21 p.m.	12
13	5/06/2017	SILVA CHAVEZ, Ana Maria	03:32 p.m.	13
14	5/06/2017	HERRERA SOTO, Lenin Abel	03:42 p.m.	14
15	5/06/2017	SOTO PECEROS JENIFER	03:51 p.m.	15
16	6/06/2017	BAUTISTA HUAMAN PAUL ALDAHIR	09:05 a.m.	16
17	6/06/2017	HERMOZA ALARCON,CLAUDIA MERCEDES	09:16 a.m.	17
18	6/06/2017	RAMIREZ ALARCON RUTH ESMERALDA	09:27 a.m.	18
19	6/06/2017	PACHECO RODRIGUEZ LISSETH YASHEMY	09:37 a.m.	19
20	6/06/2017	SISA YUTO RAQUEL	09:47 a.m.	20

Fuente: elaboración propia

FICHA DE OBSERVACIÓN PARA EL TIEMPO DEL PROCESO DE VERIFICACION DE NOTAS PRE.PRUEBA VERSUS POST.PRUEBA

Nos permite obtener los tiempos de búsqueda de notas en la verificación empleados en cada certificado de estudios.

Tabla 55: Tiempo de verificación –PRE.PRUEBA

Nro	IDENTIFICADOR	TIEMPO (H)	TIEMPO(min)
1	0001	2 horas ,32 min	152
2	0002	4 horas ,7 mim	247
3	0003	3 horas	180
4	0004	7 horas ,45 min	465
5	0005	8 horas ,39 min	519
6	0006	9 horas ,.19 min	559
7	0007	10 horas ,2 min	602
8	0008	9 horas ,37 min	577
9	0009	11 horas ,4min	664
10	0010	16horas ,35 min	995
11	0011	18 horas , 2 min	1082
12	0012	14 horas ,35 min	875
13	0013	14 horas ,17 min	857

14	0014	19 horas ,7 min	1147
15	0015	19 horas ,10 min	1150
TIEMPO PROMEDIO		11 horas ,2min	671.4

Fuente: elaboración propia

Tabla 56: Tiempo de verificación –POST.PRUEBA

Nro	IDENTIFICADOR	TIEMPO (min)
1	1	10 min
2	2	14 min
3	3	12 min
4	4	11 min
5	5	10 min
6	6	11 min
7	7	9 min
8	8	9 min
9	9	9 min
10	10	9 min
11	11	10 min
12	12	9 min
13	13	8 min
14	14	8 min
15	15	9 min
16	16	10 min
17	17	9 min
18	18	8 min
19	19	9 min
20	20	8 min
TIEMPO PROMEDIO		9.6 min

Fuente: elaboración propia

**RESULTADO OBTENIDO DE PRE-PRUEBA VERSUS PRUEBA DEL TIEMPO
PROMEDIO EN PROCESO DE VERIFICACIÓN DE NOTAS**

Tabla 57: Resultado Pre-Prueba/Post-Prueba

PRUEBA	TIEMPO PROMEDIO(min)
PRE-PRUEBA(Pre-P)	671.4
POST-PRUEBA(Post-P)	9.6
REDUCCION DEL TIEMPO(RT)	661.8
PORCENTAJE DE REDUCCION (PR) (%)	98.5 %

Fuente: elaboración propia

Calculo de resultados:

$$RT (min) = (Pre-P) - (Post-P)$$

$$PR (%) = \{[(Pre-P) - (Post-P)] / (Pre-P)\} * 100\%$$

Tabla 58: Resultado Pre-Prueba/Post-Prueba

Tiempo Promedio en la verificación de Notas			
Pre-P = 671.4 min	Pre-P = 9.6 min	RT = 661.8 min	PR=98.5%
<p>Con el uso del Sistema de Información en el proceso verificación de notas se reduce el tiempo, haciendo este análisis de la comparación con la Pre-Prueba / Post-Prueba, este proceso en promedio de 9.6 minutos por solicitud, en comparación a lo que demora lo convencional que es de 11 horas con 2 minutos promedio. Por lo tanto se hace más rápida el tiempo en este proceso de Verificación de Notas en Promedio del 98%, facilitando al personal de la Unidad de Visación a realizar este proceso y podrá continuar con las otras tareas que realiza.</p>			

Fuente: elaboración propia

CONCLUSIONES

En el desarrollo del Sistema de información se realizó con la metodología XP, puesto que su fácil el manejo de sus procesos; planificación, desarrollo, codificación, pruebas en conjunto permitió cumplir las historia de usuarios por completo y la satisfacción del cliente ya que contamos con la participación del personal de la Unidad de Visación y el director de la institución (UGEL).

Se implementó el sistema de información para el proceso de verificación de notas el mismo que consta de los módulos (registrar actas, registrar colegios, reportes), así mismo están asociados los nombres de los estudiantes a las respectivas actas escaneadas lo cual hace más rápida la verificación de las notas de los estudiantes y confiable.

Se realizó la post-prueba en la Unidad de Visación y se redujo el tiempo en el proceso de verificación de notas a un promedio de 9 minutos por solicitud; A comparación de la atención convencional que demora en promedio de 11 horas con 2 minutos. Reduciéndose en promedio del 98% lo cual facilita al personal realizar este proceso en menor tiempo y así, el solicitante poder ser atendido en menor tiempo y poder continuar con sus actividades.

RECOMENDACIONES

Al ser este tema de tesis de desarrollo de software damos algunas recomendaciones para todas las personas, que estas interesadas en realizar aplicaciones prácticas utilizando diferentes tipos de metodologías de desarrollo de software. La ingeniera del Software posee múltiples métodos que nos ayudan a solucionar muchos problemas a través del desarrollo e implementación de sistemas. Si hablamos de herramientas de desarrollo podemos decir que en la actualidad contamos con una infinidad de herramientas para el desarrollo pero si debemos tomar en cuenta el momento de elegir una, es que nos apoye y facilite el desarrollo de aplicaciones. En la presente tesis nos hemos apoyado con la metodología XP porque facilita en el manejo de sus procesos en conjunto además es amigable, siempre tomando en cuenta la presencia del usuario en el desarrollo.

Para que este trabajo de tesis tenga un efecto positivo no solo para el proceso de verificación en la búsqueda de las actas de notas sino también que cuente con el modulo del generado del certificado de notas y poder de esta manera brindar la emisión de los certificados de notas se recomienda la extensión de un sistema integral puesto que se cuenta con una Base de Datos escalable.

Los objetivos planteados en el trabajo de tesis fueron enfocados exclusivamente para el proceso de verificación sin embargo se observa la necesidad de que en el futuro se incorpore un Módulo de emisión de certificados y registro de atenciones que vaya integrado al Sistema Propuesto estando sujetos a las tareas que se realizan en esta unidad. Tales módulos tanto de ventas como de producción deberían de estar a nivel de Sitio Web permitiendo la opción de consultas en línea, Este servicio crearía un medio adicional de atención al cliente porque reduce el tiempo y facilita al usuario en la respuesta a su petición.

REFERENCIA BIBLIOGRAFICAS

- Paredes, M. C. (2015). Registro Visación de Certificado de Estudios. Andahuaylas. Instituciones Educativas UGEL Andahuaylas. (2016). [Tabla]. *Recuperado de: Padrón Instituciones de Unidad de Gestión Educativa Local Andahuaylas.*
- Murugesan, S. (2001). A NEW Discipline For Development of Web-Based Systema. *Web Engineering.*
- O'brien. (2001). *Sistema de Informacion.*
- Contreras H., & Felipe Forrero G. (2005).. *Diseño de un modelo para la implantación de un sistema de gestión documental en áreas u organizaciones jurídicas.*
- Guevara Alcalde & Luis Alberto(2007). *Implementación de un Sistema de Gestión Documentaria basado en Tecnología Web para mejorar la gestión de documentos en el Gobierno Regional Lambayeque.*
- Montero, S y Castro, M. *Ingeniería de la Web y Patrones de Diseño. España : Pearson Educación, 2005.*
- Pressman, R. *Ingeniería del Software un Enfoque Práctico. Sexta Edición . Mexico : McGrawHill, 2006.*
- kenneth y Jame.(2004) *Sistema de Informacion.*
- S, Murugesan, y otros. 2001. *Web Engineering.* 2001.
- Zerodivx. Ingeniería Web. [En línea] <http://www.zerodivx.com/index!.htm>.
- Diaz C., Mario y Suclupe A., Danny. *Sistema de Información y el Plan de Tecnología de clasificación y búsqueda de expedientes del Archivo Regional de Lambayeque.*
- . Murugesan, S. A.(2001) *NEW Discipline For Development of Web-Based Systema..*
- Chile, Ministerio de Educacion. 2002. *Portal de Certificados en Linea. [En línea]*
- Rolando, Zapata Bello. *Portal de Certificados en Linea. [En línea]*
<https://www.chileatiende.gob.cl/>
- Quero, & C. Enrique. *lenguaje de programacion. España. 2003*
- Vaswani. *lenguaje de programacion, php (p. 12). 2014*
- Canós, J; Letelier, P y Penadés, M. s.f. *Metodología de Software Ágil. Consultado, 06 de mayo de 2015.*
- Bertalanffy, Ludwing Von . *Teoría General de los Sistemas. S/Ed. México.2003*

Carmen de Pablos. José y López Hermoso, Joaquín. *Informática y Comunicaciones en la Empresa ESIC*. Editorial Madrid .2011

Chiavenato I.(2002) *Gestión del talento humano*. McGraw-Hill/interamericana editores, S.A.

Codina, Luis . *Análisis de sistemas y metodología de diseño de bases de datos documentales*. Scire. -- Vol.2, no.2 (Jul.-Dic. 1996).

Eslava Muñoz, Vicente . *El Nuevo PHP, Conceptos Avanzados*. Ed, Bubok Publishing S.L, España. 2010

G. Kappel, P. Birgit, R. Siegfried, R. Werner.(2003) *Web Engineering: The Discipline of Systematic Development of Web Applications*.

Gómez- Mejías L y otros . *Gestión de recursos humanos* .Pearson-prentice hall S.A. España.2008

Gómez-Llera,G.(1998) *Dirigir es educar*. Madrid, España: McGraw-Hill.

Aedo, P. Díaz, S. Montero y M. Castro.(2005) *El desarrollo hipermedia y web como proceso de ingeniería*. España: Pearson Educación.

Mannino, Michael V. *Administración de Bases de Datos: Diseño y Desarrollo de Aplicaciones*. 3ª Edición. Editora McGrawHill México. 2007

Marchionni, Enzo Augusto.(2006) *Administrador de servidores*. 1ª Edición. Buenos Aires: Fox Andina; Banfield - Lomas de Zamora: Gradi.

MIGUEL, Adoración de y PIATTINI, Mario G. *Diseño de bases de datos. Problemas resueltos*. Madrid. RA-MA, Notes in Computer Science 2016 Springer, pag 3 – 13. 2006.

Prieto Espinosa, Alberto y otros . *Introducción a la Informática*. 3ra. Edición, Editora McGrawHill. México.2002

Quero Catalinas, Enrique. *Sistemas Operativos y Lenguajes de Programación*. Ediciones Thomson PARANINFO, Madrid.2003

RAMOS, MARÍA J. y otros(2006). *Sistemas gestores de bases de datos*. Editora: McGraw-Hill/Interamericana, España, S.A.U.

S. Castejon Garrido. *Arquitectura y diseño de sistemas Web modernos*. InforMAS, Revista de Ingeniería Informáticadel CIIRM, num. 1, diciembre, 2004.
<http://www.ciimurcia.es/informas/ene05/articulos/>.

SILBERSCHATZ, Abraham y otros.(200) *Fundamentos de bases de datos*. Editora. McGraw-Hill, Madrid.

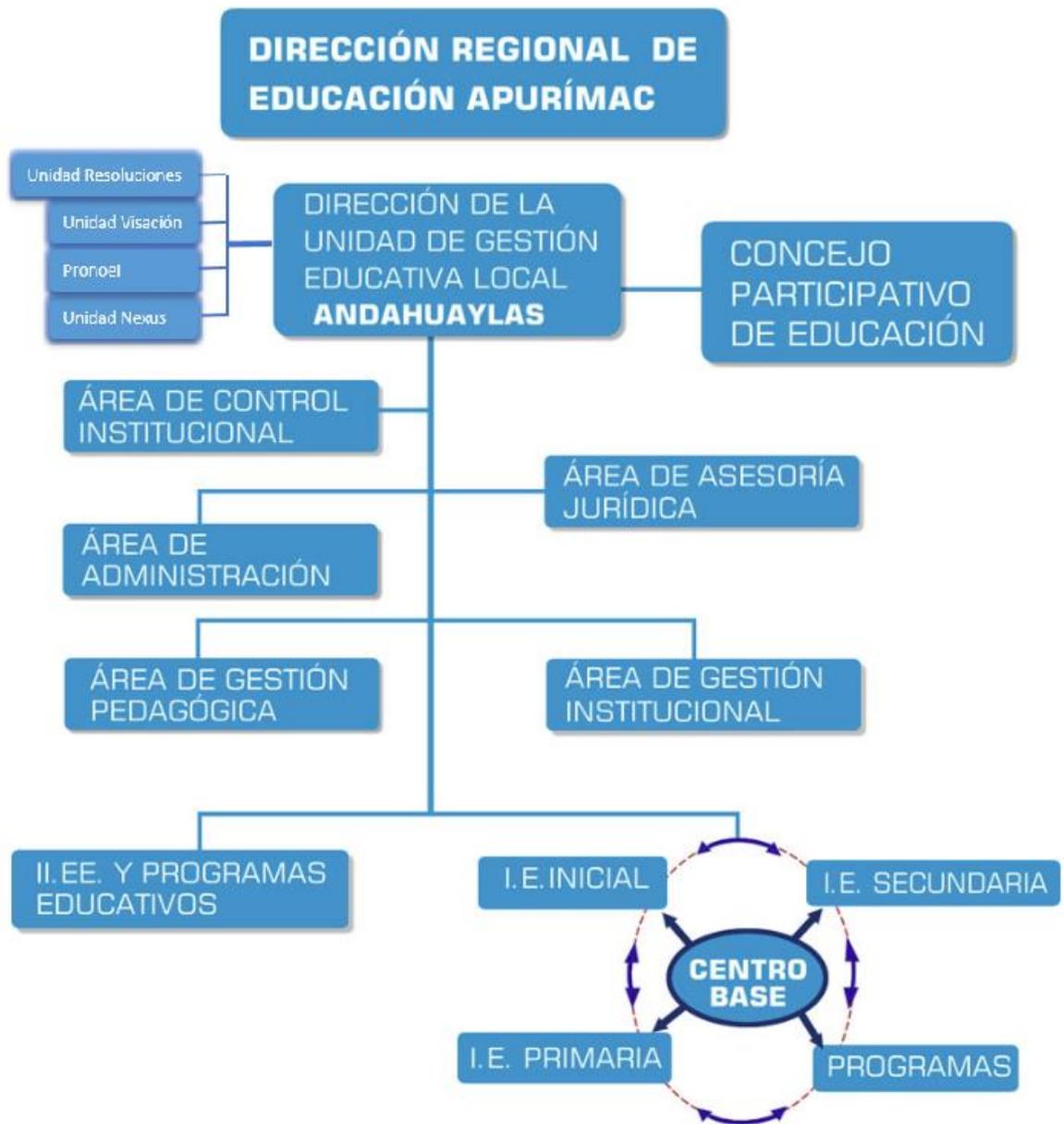
Vaswani, Vikram.(2012) *Fundamentos de PHP*. Editores McGRAW HILL/INTERAMERICANA, España.

Welling, Luke y Thomson, Laura .(2005) *Desarrollo web con PHP y MySQL*, Ediciones ANAYA MULTIMEDIA, Madrid.

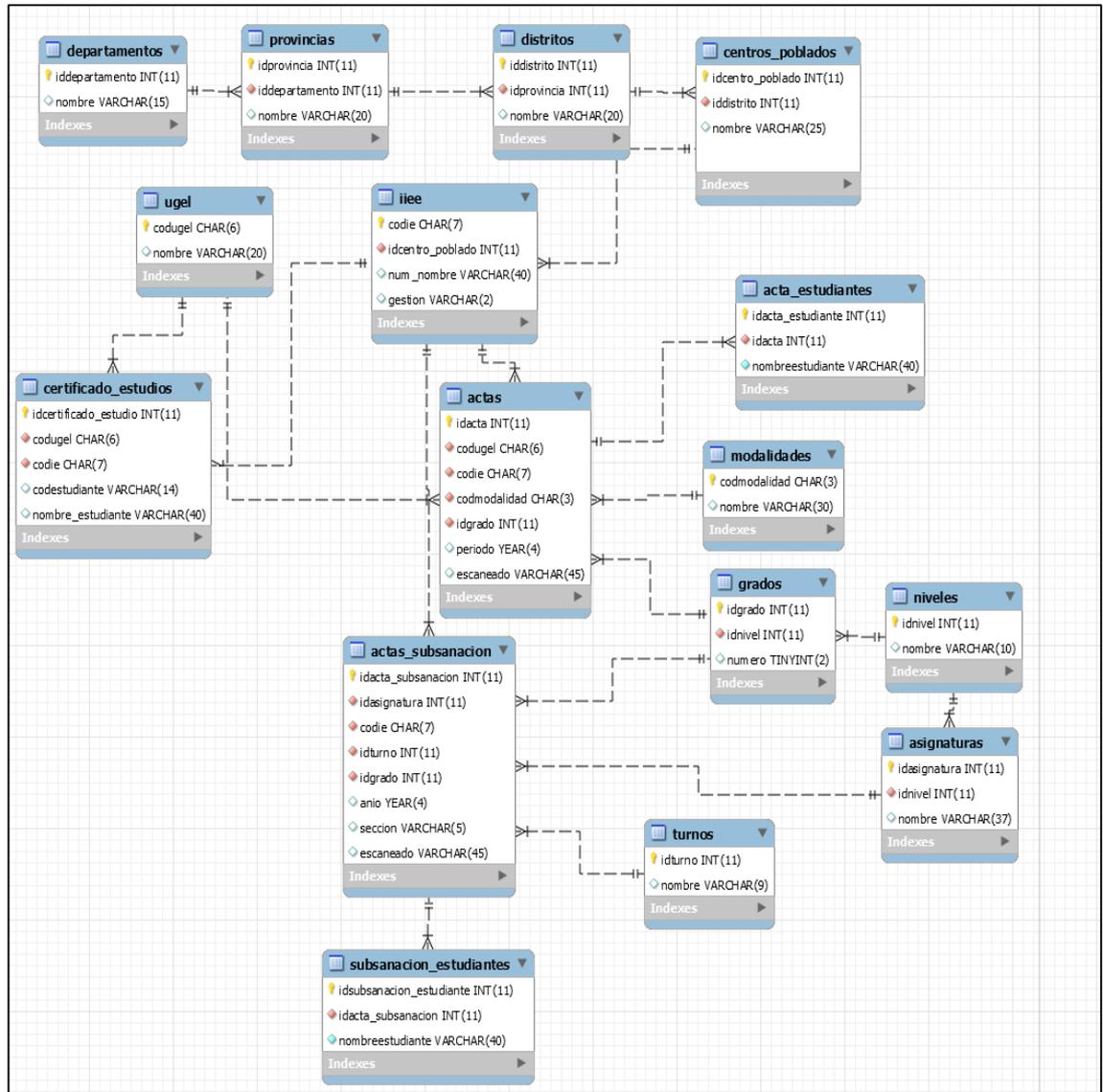
ZERODIVX.(2007) *Ingeniería Web*.

ANEXOS

1. ORGANIGRAMA UGEL ANDAHUAYLAS

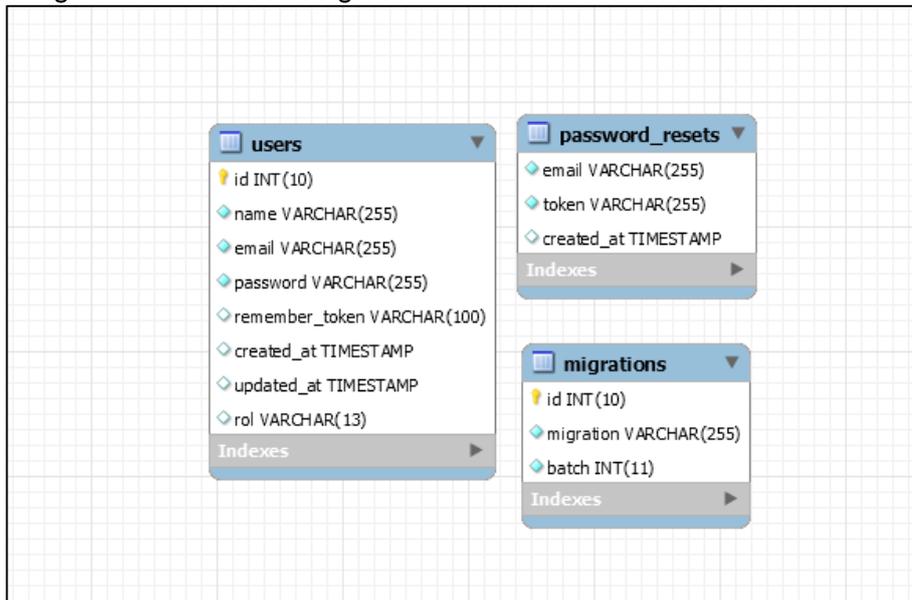


2. Diagrama del Modelo Entidad/Relación



Fuente: Elaboración propia

Imagen 30: Tablas del Login de VisaSof



Fuente: Elaboración propia

3. Código Fuente de los Controladores Acta controller

```
class ActaController extends Controller
{
    public function index(Request $request)
    {
        if($request){
            $query = trim($request->get('searchText'));
            $actas = DB::table('actas as a')
                ->join('ugel as u', 'a.codugel', '=', 'u.codugel')
                ->join('IIIEE as i', 'a.codie', '=', 'i.codie')
                ->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
                ->join('grados as g', 'a.idgrado', '=', 'g.idgrado')
                ->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
                    'a.periodo', 'a.escaneado',
                    'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as nmodalidad',
                    'g.numero as ngrado'
                )
                ->where('a.periodo', 'LIKE', '%'.$query.'%'
```

```

->orderBy('a.idacta')
->paginate(7);

return view('registro.acta.index', ["actas"=>$actas, "searchText"=>$query]);
}
}

public function create()
{
    $ugeles = DB::table('ugel')->get();
    $ieducativas = DB::table('IIEE')->get();
    $modalidades = DB::table('modalidades')->get();
    $niveles = DB::table('niveles')->get();
    $grados = DB::table('grados')->get();

    return view("registro.acta.create", ["ugeles"=>$ugeles,
"ieducativas"=>$ieducativas,
"modalidades"=>$modalidades, "niveles"=>$niveles,
"grados"=>$grados]);
}

public function store(ActaFormRequest $request)
{
    $acta = new Acta;
    $acta->codugel = $request->get('codugel');
    $acta->codie = $request->get('codie');
    $acta->codmodalidad = $request->get('codmodalidad');
    $acta->idgrado = $request->get('idgrado');
    $acta->periodo = $request->get('periodo');

    if(Input::hasFile('escaneado')){
        $file = Input::file('escaneado');
        $file->move(public_path().'/imagenes/escaneados/', $file-
>getClientOriginalName());
        $acta->escaneado = $file->getClientOriginalName();
    }
}

```

```

}

$acta->save();
return Redirect::to('registro/acta');
}

public function show($id)
{
    // Solicitamos los datos que necesitaremos
    $query = trim($id);
    $acta = DB::table('actas as a')
->join('ugel as u', 'a.codugel', '=', 'u.codugel')
->join('IIIEE as i', 'a.codie', '=', 'i.codie')
->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
->join('grados as g', 'a.idgrado', '=', 'g.idgrado')
->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
        'a.periodo', 'a.escaneado',
        'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as nmodalidad',
        'g.numero as ngrado'
        )
->where('a.idacta', 'LIKE', '%'.$query.'%')
->orderBy('a.idacta')
->first();
    $actaestudiantes = DB::table('acta_estudiantes')
//->join('actas as a', 'ae.idacta', '=', 'a.idacta')
//->select('ae.idacta_estudiante', 'ae.idacta', 'ae.codestudiante')
->where('idacta', 'LIKE', '%'.$query.'%')
->orderBy('idacta_estudiante', 'desc')
//->orderBy('idacta_estudiante', 'desc')
->paginate(7);
    return view('registro.acta.show', ["acta"=>$acta,
"actaestudiantes"=>$actaestudiantes]);
}
/*
// NUEVO METODO QUE AGREGA UN COD ESTUDIANTE

```

```

public function adestudiante(ActaFormRequest $request, $id){
    show(1);
}
*/
public function edit($id)
{
    $acta = Acta::findOrFail($id)
->join('grados as g', 'actas.idgrado','=', 'g.idgrado')
->where('idacta', 'LIKE', '%'.$id.'%')
->select('idacta', 'codugel', 'codie', 'codmodalidad', 'actas.idgrado as idgrado',
        'periodo', 'escaneado',
        'g.idnivel as idnivel')
->first();// El primer elemento que coincide

    $ugeles = DB::table('ugel')->get();
    $ieducativas = DB::table('IIEE')->get();
    $modalidades = DB::table('modalidades')->get();
    $niveles = DB::table('niveles')->get();
    $grados = DB::table('grados')->get();

    return view("registro.acta.edit", ["acta"=>$acta, "ugeles"=>$ugeles,
        "ieducativas"=>$ieducativas, "modalidades"=>$modalidades,
        "niveles"=>$niveles, "grados"=>$grados]);
}

public function update(ActaFormRequest $request, $id)
{
    $acta = Acta::findOrFail($id);
    $acta->codugel = $request->get('codugel');
    $acta->codie = $request->get('codie');
    $acta->codmodalidad = $request->get('codmodalidad');
    $acta->idgrado = $request->get('idgrado');
    $acta->periodo = $request->get('periodo');

    //$acta->escaneado = $request->get('escaneado');
}

```

```

if(Input::hasFile('escaneado')){
    $file = Input::file('escaneado');
    $file->move(public_path().'imagenes/escaneados/', $file-
>getClientOriginalName());
    $acta->escaneado = $file->getClientOriginalName();
}

$acta->update();
return Redirect::to('registro/acta');
}

public function destroy($id)
{
    Acta::destroy($id);
    return Redirect::to('registro/acta');
}
}

```

ActaEstudiante controller

```

class ActaEstudianteController extends Controller
{
    public function index(Request $request)
    {
        if($request)
        {
            $query = trim($request->get('searchText'));
            $actaestudiantes = DB::table('acta_estudiantes as ae')
            // Por gusto
            //->join('actas as a', 'ae.idacta', '=', 'a.idacta')
            ->select('ae.idacta_estudiante', 'ae.idacta', 'ae.nombreestudiante')
            ->where('ae.nombreestudiante', 'LIKE', '%'.$query.'%')
            ->orderBy('ae.idacta_estudiante')
            ->paginate(7);

            return view('registro.actaestudiante.index',
["actaestudiantes"=>$actaestudiantes, "searchText"=>$query]);

```

```

    }
}

public function create()
{
    $actas = DB::table('actas')->get();
    return view("registro.actaestudiante.create", ["actas"=>$actas]);
}

public function store(ActaEstudianteFormRequest $request)
{
    $actaestudiante = new ActaEstudiante;
    $actaestudiante->idacta = $request->get('idacta');
    $actaestudiante->nombrestudiante = $request->get('nombrestudiante');

    $actaestudiante->save();

    if($request->get('inacta'))
        return Redirect::to('registro/acta/'.$request->get('idacta'));
    else
        return Redirect::to('registro/actaestudiante');
}

public function show($id)
{
    return view('registro.actaestudiante.show',
["actaestudiante"=>ActaEstudiante::findOrFail($id)]);
}

public function edit($id)
{
    $actaestudiante = ActaEstudiante::findOrFail($id);
    $actas = DB::table('actas')->get();

```

```
    return view("registro.actaestudiante.edit", ["actaestudiante"=>$actaestudiante,
"actas"=>$actas]);
}
```

```
public function update(ActaEstudianteFormRequest $request, $id)
{
    $actaestudiante = ActaEstudiante::findOrFail($id);
    $actaestudiante->idacta = $request->get('idacta');
    $actaestudiante->nombreestudiante = $request->get('nombreestudiante');

    $actaestudiante->save();
    return Redirect::to('registro/actaestudiante');
}
```

```
public function destroy($id)
{
    ActaEstudiante::destroy($id);
    return Redirect::to('registro.actaestudiante');
}
}
```

Reporte controller

```
class ReporteController extends Controller
{
    public function index(Request $request)
    {
        if($request){
            $queryPeriodo = trim($request->get('periodo'));
            $queryNombreEstudiante = trim($request->get('nombreestudiante'));
            $queryIdNivel = trim($request->get('idnivel'));

            $niveles = DB::table('niveles')->get();
            $idnivelselect = $queryIdNivel;
            if($queryIdNivel==1)
                $idnivelselect="";

            if(!$queryNombreEstudiante){
                $actas = DB::table('actas as a')
                    ->join('ugel as u', 'a.codugel', '=', 'u.codugel')
                    ->join('IIIEE as i', 'a.codie', '=', 'i.codie')
                    ->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
                    ->join('grados as g', 'a.idgrado', '=', 'g.idgrado')
                    //->join('acta_estudiantes as ae', 'a.idacta', '=', 'ae.idacta')
                    ->join('niveles as n', 'n.idnivel', '=', 'g.idnivel')
                    ->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
                        'a.periodo', 'a.escañado',
                        'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as
nmodalidad',
                        'g.numero as ngrado'
                    )
                    ->where('a.periodo', 'LIKE', '%'.$queryPeriodo.'%')
                    //->where('ae.nombreestudiante', 'LIKE', '%'.$queryNombreEstudiante.'%')
                    ->where('n.idnivel', 'LIKE', '%'.$idnivelselect.'%')
                    ->orderBy('a.idacta')
                    ->paginate(7);
            }
        }
    }
}
```

```

}else{
    $actas = DB::table('actas as a')
    ->join('ugel as u', 'a.codugel', '=', 'u.codugel')
    ->join('IIIEE as i', 'a.codie', '=', 'i.codie')
    ->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
    ->join('grados as g', 'a.idgrado', '=', 'g.idgrado')
    ->join('acta_estudiantes as ae', 'a.idacta', '=', 'ae.idacta')
    ->join('niveles as n', 'n.idnivel', '=', 'g.idnivel')
    ->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
        'a.periodo', 'a.escaneado',
        'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as
nmodalidad',
        'g.numero as ngrado'
        )
    ->where('a.periodo', 'LIKE', '%'.$queryPeriodo.'%')
    ->where('ae.nombreestudiante', 'LIKE', '%'.$queryNombreEstudiante.'%')
    ->where('n.idnivel', 'LIKE', '%'.$idnivelselect.'%')
    ->orderBy('a.idacta')
    ->paginate(7);
}

return view('registro.reporte.index', ["actas"=>$actas,
"periodo"=>$queryPeriodo,
"nombreestudiante"=>$queryNombreEstudiante, "niveles"=>$niveles,
"idnivelselect"=>$queryIdNivel]);
}
}

```

```

private function selectActasPeriodo($queryPeriodo){
    // Selecciona todas las actas de acuerdo al periodo
    $actas = DB::table('actas as a')
    ->join('ugel as u', 'a.codugel', '=', 'u.codugel')
    ->join('IIIEE as i', 'a.codie', '=', 'i.codie')
    ->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
    ->join('grados as g', 'a.idgrado', '=', 'g.idgrado')

```

```

->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
        'a.periodo', 'a.escaneado',
        'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as nmodalidad',
        'g.numero as ngrado'
        )
->where('a.periodo', 'LIKE', '%'.$queryPeriodo.'%')
->orderBy('a.idacta')
->paginate(7);

return $actas;
}

```

```

private function selectActasNombreEstudiante($queryNombreEstudiante){
    // Selecciona todas las actas de acuerdo al codigo estudiante
    // Por eso se une con la tabla hija acta_estudiantes
    $actas = DB::table('actas as a')
->join('ugel as u', 'a.codugel', '=', 'u.codugel')
->join('IIEE as i', 'a.codie', '=', 'i.codie')
->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
->join('grados as g', 'a.idgrado', '=', 'g.idgrado')
->join('acta_estudiantes as ae', 'a.idacta', '=', 'ae.idacta')
->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
        'a.periodo', 'a.escaneado',
        'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as nmodalidad',
        'g.numero as ngrado'
        )
->where('ae.nombreestudiante', 'LIKE', '%'.$queryNombreEstudiante.'%')
->orderBy('a.idacta')
->paginate(7);

return $actas;
}

```

```

private function selectActasIdNivel($queryIdNivel){
    // Selecciona todas las actas de acuerdo al Id del nivel
    // Por eso se une con la tabla abuela (jejeje) niveles

```

```

    $actas = DB::table('actas as a')
->join('ugel as u', 'a.codugel', '=', 'u.codugel')
->join('IIEE as i', 'a.codie', '=', 'i.codie')
->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
->join('grados as g', 'a.idgrado', '=', 'g.idgrado')
->join('niveles as n', 'n.idnivel', '=', 'g.idnivel')
->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
        'a.periodo', 'a.escaneado',
        'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as nmodalidad',
        'g.numero as ngrado'
        )
->where('n.idnivel', 'LIKE', '%'.$queryIdNivel.'%')
->orderBy('a.idacta')
->paginate(7);

return $actas;
}
private function selectActasAll($queryPeriodo, $queryNombreEstudiante,
$queryIdNivel){
    $actas = DB::table('actas as a')
->join('ugel as u', 'a.codugel', '=', 'u.codugel')
->join('IIEE as i', 'a.codie', '=', 'i.codie')
->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
->join('grados as g', 'a.idgrado', '=', 'g.idgrado')
->join('acta_estudiantes as ae', 'a.idacta', '=', 'ae.idacta')
->join('niveles as n', 'n.idnivel', '=', 'g.idnivel')
->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
        'a.periodo', 'a.escaneado',
        'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as nmodalidad',
        'g.numero as ngrado'
        )
->where('a.periodo', 'LIKE', '%'.$queryPeriodo.'%')
->orWhere('ae.nombreestudiante', 'LIKE', '%'.$queryNombreEstudiante.'%')
->orWhere('n.idnivel', 'LIKE', '%'.$queryIdNivel.'%')
->orderBy('a.idacta')

```

```

->paginate(7);

return $actas;
}
/*
public function create()
{
    $ugeles = DB::table('ugel')->get();
    $ieducativas = DB::table('IIEE')->get();
    $modalidades = DB::table('modalidades')->get();
    $niveles = DB::table('niveles')->get();
    $grados = DB::table('grados')->get();

    return view("registro.acta.create", ["ugeles"=>$ugeles,
"ieducativas"=>$ieducativas,
"modalidades"=>$modalidades,
"niveles"=>$niveles,
"grados"=>$grados]);
}
public function store(ActaFormRequest $request)
{
    $acta = new Acta;
    $acta->codugel = $request->get('codugel');
    $acta->codie = $request->get('codie');
    $acta->codmodalidad = $request->get('codmodalidad');
    $acta->idgrado = $request->get('idgrado');
    $acta->periodo = $request->get('periodo');

    //$acta->escaneado = $request->get('escaneado')
    if(Input::hasFile('escaneado')){
        $file = Input::file('escaneado');
        $file->move(public_path().'/imagenes/escaneados/', $file-
>getClientOriginalName());
        $acta->escaneado = $file->getClientOriginalName();
    }
    $acta->save();
}

```

```

    return Redirect::to('registro/acta');
}
*/
public function show($id)
{
    // Solicitamos los datos que necesitaremos
    $query = trim($id);
    $acta = DB::table('actas as a')
    ->join('ugel as u', 'a.codugel', '=', 'u.codugel')
    ->join('IIIEE as i', 'a.codie', '=', 'i.codie')
    ->join('modalidades as m', 'a.codmodalidad', '=', 'm.codmodalidad')
    ->join('grados as g', 'a.idgrado', '=', 'g.idgrado')
    ->select('a.idacta', 'a.codugel', 'a.codie', 'a.codmodalidad', 'a.idgrado',
            'a.periodo', 'a.escaneado',
            'u.nombre as nugel', 'i.num_nombre as nnie', 'm.nombre as nmodalidad',
            'g.numero as ngrado'
            )
    ->where('a.idacta', 'LIKE', '%'.$query.'%') ->orderBy('a.idacta')
    ->first();
    $actaestudiantes = DB::table('acta_estudiantes')
    //->join('actas as a', 'ae.idacta', '=', 'a.idacta')
    //->select('ae.idacta_estudiante', 'ae.idacta', 'ae.nombreestudiante')
    ->where('idacta', 'LIKE', '%'.$query.'%')
    ->orderBy('idacta_estudiante', 'desc')
    //->orderBy('idacta_estudiante', 'desc')
    ->paginate(7);

    return view('registro.acta.show', ["acta"=>$acta,
"actaestudiantes"=>$actaestudiantes]);
}
}

```

4. Diagrama de Clases

