

FACULTAD DE INGENIERÍA

Escuela Académico Profesional de Ingeniería
de Sistemas e Informática

Trabajo de Investigación

**Análisis y diseño de un sistema de información para la
gestión y administración de procesos específicos de
deudas administrativas de docentes para la Unidad de
Gestión Educativa Local en Ilo**

Luis Alberto Borit Guitton

Para optar el Grado Académico de
Bachiller en Ingeniería de Sistemas e Informática

Arequipa, 2020

Repositorio Institucional Continental
Trabajo de investigación



Esta obra está bajo una Licencia "Creative Commons Atribución 4.0 Internacional" .

AGRADECIMIENTOS

Agradezco a esta casa de estudios por brindarme los conocimientos y formación académica de los que ahora gozo y los cuales me proveerán de frutos en el mañana.

DEDICATORIA

Esta tesina y todo el esfuerzo que implicó, se la dedico a mi madre, Yeny, por haber sido el empuje, soporte constante y mi gran productora de motivación, a pesar de todo; a mi padre, Luis, por haber confiado en mi esfuerzo, creído en mi capacidad y por ser mi ejemplo de superación; a mi hermano Abel que fue el más claro modelo de dedicación, continuidad, esfuerzo, sacrificio y apoyo incondicional, y a Dios, que, sin Él, nada de esto sería posible. Per Áspera Ad Astra.

Contenido

RESUMEN	5
CAPÍTULO I: PLANTEAMIENTO DEL ESTUDIO	6
1. PLANTEAMIENTO DEL ESTUDIO	6
1.1. Planteamiento y formulación del problema	6
1.1.1. Problema General:	6
1.1.2. Problemas específicos:.....	6
1.2. Objetivos	7
1.2.1. Objetivo General.....	7
1.2.2. Objetivos Específicos	7
1.3. Justificación	7
CAPITULO II: MARCO TEÓRICO.....	9
2. MARCO TEÓRICO	9
2.1. Antecedentes del problema	9
2.2. Antecedentes Históricos	12
2.3. Bases teóricas	12
2.4. Definición de términos básicos	15
CAPITULO 3: METODOLOGÍA.....	16
3. METODOLOGÍA	16
3.1. Metodología aplicada para el desarrollo de la solución	16
CAPÍTULO 4: ANÁLISIS Y DISEÑO	17
4. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN	17
4.1. Identificación de requerimientos	17
4.2. Análisis de la solución.....	24
4.3. Diseño.....	28
CAPÍTULO V: DISEÑO	30
5. Construcción	30
5.1. Desarrollo	30
5.1.1. Codificación de modelos en JavaScript.....	30
5.2. Pruebas y resultados	34
Conclusiones.....	35
Trabajos Futuros	36

RESUMEN

Este trabajo de investigación propone una solución informática, un sistema de información, para solucionar las necesidades de una entidad de gestión educativa local, tales como la gestión y administración de sus deudas, con la finalidad de poder continuar con la actividad educativa pública en beneficio de la comunidad.

La solución propuesta, contará con múltiples módulos de control, los cuales se construirán a partir de los requerimientos obtenidos, para el correcto diseño del software, logrando así, la correcta funcionalidad y continuidad de la entidad.

Las funciones de este sistema de información, están contempladas en un marco de trabajo donde los usuarios podrán acceder vía online, configurar parámetros, realizar consultas, efectuar y modificar registros, para posteriormente analizar reportes autogenerados.

Durante la construcción de este software, se emplearán metodologías ágiles tales como Xtreme Programming y Kanban, las que se complementan, puesto que la primera de ellas tiene como objetivo el desarrollo de software con seguimiento a errores durante la codificación, para lograr un desarrollo con funcionalidades de acuerdo a lo estipulado; y la segunda metodología hace seguimiento a las tareas por realizar, en proceso y las tareas culminadas; logrando así un control efectivo y un rendimiento óptimo para el cumplimiento de entrega de producto.

CAPÍTULO I: PLANTEAMIENTO DEL ESTUDIO

1. PLANTEAMIENTO DEL ESTUDIO

1.1. Planteamiento y formulación del problema

Puesto que la Unidad de Gestión Educativa Local, es un ente público que regula las actividades de las instituciones educativas de forma focalizada, surge dentro de esta entidad, una necesidad, la cual apunta al registro y control de deudas docentes; en un área propia, específicamente en la unidad de Administración General, que debería ser cubierta para el correcto desempeño de las áreas cercanas interrelacionadas, como: Logística, Pagos y Cobranzas, Planeamiento y Deudas; para continuar con la actividad en beneficio de la comunidad educativa por medio de la mejora de procesos a través de un sistema de información, que agilice la necesidad mencionada, en la Imagen 1 se observa la encargada de archivo general, de donde parten todos los informes y documentaciones de forma no automatizada.



Imagen 1: Archivo General de UGEL Ilo

1.1.1. Problema General:

¿Cuál sería una propuesta de análisis y diseño de un sistema de información para la gestión y administración de procesos específicos de deudas administrativas de docentes para la unidad de gestión educativa local en Ilo?

1.1.2. Problemas específicos:

¿Cuáles son los requisitos técnicos para realizar una propuesta de análisis y diseño de un sistema de información para la gestión y administración de procesos específicos de deudas administrativas de docentes para la unidad de gestión educativa local en Ilo?

¿Cuáles son los métodos necesarios para realizar una propuesta de análisis y diseño de un sistema de información para la gestión y administración de procesos específicos de deudas administrativas de docentes para la unidad de gestión educativa local en Ilo?

1.2. Objetivos

1.2.1. *Objetivo General*

Analizar y diseñar un sistema de información web (1), con la finalidad de administrar y gestionar los procesos específicos de deudas en la UGEL – ILO, brindando un apoyo a la toma de decisiones y sirviendo como soporte para las labores administrativas.

1.2.2. *Objetivos Específicos*

- Elaborar el análisis y diseño de un sistema de información, tomando como punto de partida los requisitos de la institución donde se implementará el proyecto.
- Seleccionar y definir la arquitectura donde se implementará el sistema de información, con la finalidad de la propia escalabilidad (2).
- Diseñar el esquema o modelo de base de datos no relacional que integre y se adapte, a los requisitos y procesos de deudas de la institución.
- Diseñar la interfaz de usuario, donde se realizarán todos los procesos del lado cliente del sistema de información.

1.3. Justificación

Este trabajo de investigación, es una propuesta del diseño y análisis para la posterior construcción de un sistema de información (3), creando así los pilares para el desarrollo de este software, en el cuál se abordarán y trabajarán procesos específicos de deudas de la UGEL – ILO, teniendo la finalidad de integrar los requisitos de la institución donde se implementará a futuro este sistema, con una solución diseñada a medida, para la correcta funcionalidad de los procesos requeridos y la escalabilidad de este sistema.

Ilo, 09 de marzo del 2018

Carta N° 001-2018-LABG

Sres.
UNIDAD DE GESTION EDUCATIVA LOCAL ILO
Presente.-

ASUNTO: CULMINACION DE SERVICIO

De mi Especial consideración.

Me es grato dirigirme a Ud. Para saludarlo cordialmente y a la vez hacerle saber la culminación del servicio del "Sistema de Gestión y Administración de Deudas Sociales", así como también hacerle entrega de los productos referentes a los cuales hago referencia a continuación.



• **ENTREGABLES.**

Se esta entregando adjunto a la presente los siguientes documentos:

- Manual de Usuario, el cual contiene la parte del usuario y una guía de requerimientos de instalación del software.
- Manual de Administrador, Diagrama de Base de Datos y Diccionario de datos
- Carta de garantía a favor de UGEL Ilo.
- Carta de Sesión de Derecho de Autor a favor de UGEL Ilo.
- Carta de Autorización de pago.
- Recibo por honorarios electrónicos N° E001-2
- Suspensión de 4ta Categoría-Formulario 1609
- Acta de capacitación del usuario.

• **ENTREGABLES CD.**

Se esta entregando 1 cd conteniendo la siguiente información:

- Programa fuentes del Software de Gestión y Administración de Deudas Sociales.
- Base de Datos
- Manual de Usuario, el cual contiene la parte del usuario y una guía de requerimientos de instalación del software.

Sin otro en particular me despido de Ud. No si antes de expresarle las muestra de mi especial consideración y estima personal.

Atentamente.


BORIS GUIFFON LUIS ALBERTO
R.U.C.: 10701361976

C.c.Archivo

Evidencia de la Culminación del Requerimiento de Servicio de Desarrollo de Software: sistema de información para la gestión y administración de procesos específicos de deudas administrativas de docentes para la unidad de gestión educativa local en Ilo (UGEL-ILO).

CAPITULO II: MARCO TEÓRICO

2. MARCO TEÓRICO

2.1. Antecedentes del problema

- Antecedente 1:
 - Problemática
 - Previamente, sólo ha habido existencia de informes acerca de la necesidad de controlar y administrar las deudas docentes, sin embargo, no se ha construido ningún tipo de sistema de información que resuelva el problema, ni diseñado o analizado a profundidad el tema en cuestión, tomando como referencia otras UGEL.
 - Objetivo
 - Diseño y análisis de una solución digital que resuelva el problema de la administración de deudas docentes.
 - Metodología
 - Análisis de la problemática
 - Diseño del proyecto
 - Diseño de las interfaces
 - Presentación de la solución
 - Conclusión
 - Mediante el análisis y diseño de una solución digital se busca resolver el problema de la administración de deudas docentes de una manera automatizada.
- Antecedente 2:
 - Problemática:
 - Se ha dado el caso de que también se vio la misma necesidad de un sistema de información que ayude a la gestión y administración de las deudas, pero han controlado el problema, de una forma no automatizada y manual con hojas de cálculo en Microsoft Excel.
 - Objetivo:
 - Obtener zonas de deficiencia para conseguir un punto de partida en cuanto a la implementación de la solución.
 - Metodología:
 - Entrevistas presenciales a miembros del personal
 - Conclusiones:

- Se obtuvieron zonas de deficiencia importantes para conseguir un punto de partida para lograr en una próxima fase la implementación de la solución al problema de la administración de las deudas docentes.
- Antecedente 3:
 - Problemática:
 - Existencia de pérdida de información; este caso es de suma relevancia, porque para recuperar la información de las deudas se tuvo que revisar la documentación almacenada en archivo, logrando una demora en la ejecución de procesos.
 - Objetivo:
 - Encontrar una justificación de valor para la realización del proyecto y basarse en ella, para las futuras características técnicas.
 - Metodología:
 - Entrevistas presenciales a miembros del personal
 - Conclusiones:
 - Se encontró una justificación de valor para la realización del proyecto, siendo esta el pilar fundamental para la implementación.
- Antecedente 4:
 - Problemática:
 - Se requirió la elaboración de un sistema que permita, mediante accesos de usuario, mostrar resúmenes de información a la gerencia de Administración, con la finalidad de mejorar la toma de decisiones.
 - Objetivo:
 - Reunir requerimientos para el desarrollo del sistema
 - Metodología:
 - Observación y entrevistas presenciales a miembros del personal
 - Conclusiones:

- Se reunieron los requerimientos necesarios para el desarrollo del sistema que resolverá la necesidad de mejorar la toma de decisiones.
- Antecedente 5:
 - Problemática:
 - No se contaba con la agilidad necesaria cuando otras gerencias distintas a la solicitante del requerimiento, pedían resúmenes de información sobre deudas docentes. Esto ralentizaba los flujos de información en toda la entidad.
 - Objetivo:
 - Remediar la lentitud de los flujos de información de la entidad
 - Metodología:
 - Desarrollo de software, scrum, XP
 - Conclusiones:
 - Desarrollar en base a los requerimientos obtenidos un sistema que remedie, como parte de los objetivos, la falta de agilidad en el flujo de información.
- Antecedente 6:
 - Problemática:
 - La seguridad de la información se encontraba en peligro, puesto que no estaba digitalizada, es por ello que se requirió el desarrollo del software en mención, para evitar pérdidas o modificaciones externas de datos.
 - Objetivo:
 - Mejorar el tratamiento de la información al digitalizarla y almacenarla de forma segura
 - Metodología:
 - Desarrollo de software, scrum, XP, entrevistas presenciales a miembros del personal

- Conclusiones:
 - En base al desarrollo de un sistema web, la información digitalizada se almacenará en servidores, disminuyendo significativamente el riesgo de pérdida

2.2. Antecedentes Históricos

Tal cual cita Susana Montero (4) “[...], que plantea un proceso sistemático, integrador e independiente de la plataforma de implementación para modelar y evaluar aplicaciones [...]”, es una necesidad de la institución, UGEL – ILO, contar con una metodología que se integre con un sistema de apoyo a la toma de decisiones y de gestión y administración de deudas.

Teniendo en cuenta de que la tecnología web es una tendencia (4), es un punto a favor de la portabilidad y escalabilidad del sistema de información a analizar e interpretar.

En la UGEL – ILO, se ha venido usando el lenguaje de programación JavaScript, en el lado cliente, sin embargo, para este proyecto usaremos, también dicho lenguaje en el lado servidor, siendo el caso del Framework Express y su servidor NodeJS, (5).

Para las metodologías, se pondrá en práctica Scrum y XP. Siendo éstas las que van más de acorde al tipo de proyecto que se busca diseñar, Scrum, es una metodología ágil que es ideal para proyectos pequeños y multidisciplinarios, mientras que su complemento sería XP, siglas de Xtreme Programming, que busca usar la tendencia de desarrollo de pair programming (programación en pareja), en la que se desarrolla software con un desarrollador de software, y un guía el cuál supervise las implementaciones en el sistema de información en construcción.

2.3. Bases teóricas

Desde una de las dos perspectivas del desarrollo de software, en este caso la orientada a la creación de productos tecnológicos, la definición del término “producto”, denota una inclinación complementaria del término “proceso”, siendo este el que identifica la labor humana en muchas instancias de la actividad del hombre.

El de desarrollo de software se alinea con los conjuntos de actividades desarrolladas por el hombre, adecuando esta práctica a la creación de una idea basada en la abstracción de una actividad, transfigurada en un proceso llamado “entorno de trabajo de software” o “software framework”.

Cabe mencionar que tanto “producto” como “proceso” son complementarios, y no pueden ser tratados cada uno, singularmente como base para creación explícita del otro.

- Bases de las Tecnologías Web

Definiendo en una comparativa entre sitio y aplicación web, Oxford lo cita como "un programa o conjunto de programas para ayudar al usuario de un ordenador para procesar una tarea específica".

Una aplicación es un conjunto de programas que integrados que funciona a través de la web, en comparativa a un sitio web que es una forma no dinámica de presentación de información. La aplicación en web, es útil a la hora de ejercer su rol como herramienta ya que dinamiza con el usuario final, su información y acciones (6).

- El Lado Cliente

Lenguajes del lado cliente: Son los lenguajes no compilados, pero si interpretados por el navegador, usualmente son estandarizados.

○ HTML:

HyperText Markup Language (HTML) no es un lenguaje de programación propiamente dicho, pero si es un lenguaje de etiquetas que permite la estructura de todo el contenido de la web (7).

○ JavaScript:

Lenguaje de programación interpretado por la web para dinamizar información, contenido, datos y acciones del usuario final, mediante estándares de la web (7).

○ CSS:

Estas hojas de estilos modifican atributos y formatos de diferentes partes y bloques del código HTML. (7)

- El Lado Servidor

Lenguajes del lado servidor: Se ejecutan de forma compilada o interpretada mediante el servidor y es enviado al lado cliente de una forma concisa para el entendimiento de este lado de la web:

○ PHP

○ C#

- JavaScript (también usado del lado servidor)
- Golang
- Python
- Java
- Bases de Datos:

Las bases de datos son elementos tecnológicos que permiten el almacenamiento de información estructurada y ordenada, mediante atributos, campos e identificadores. Dentro de las operaciones que se realizan desde la web hasta esta utilidad están las de: leer, modificar, eliminar y actualizar información (8)

- Metodologías Ágiles:

Definición:

Una Metodología de desarrollo de software, consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo. Regularmente este tipo de metodología, tienen la necesidad de venir documentadas, para que los programadores que estarán dentro de la planeación del proyecto, comprendan perfectamente la metodología y en algunos casos el ciclo de vida del software que se pretende seguir (10)

- Extreme Programming – XP:

Es una metodología ágil que promueve el trabajo en equipo. XP está definida por el constante feedback desde los desarrolladores hasta el cliente final. XP es muy útil para proyectos con ingeniería de requerimientos poco precisos y relativamente fluctuantes” (11).

- Scrum:

“Desarrollada por Ken Schwaber, Jeff Sutherland y Mike Beedle. Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos, sus principales características se pueden resumir en dos”. “El desarrollo de software se realiza mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente”, otra particularidad relevante son los meets durante el desarrollo”. “Éstas son las verdaderas protagonistas, especialmente la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración” (11).

2.4. Definición de términos básicos

- Producto: Es el resultado de una actividad tangible del hombre.
- Proceso: Denota el despliegue de la actividad realizada (12).
- Xtreme Programming: Metodología Ágil de desarrollo de software donde se implementan actividades de revisión de código. (11)
- Pair Programming: Metodología de desarrollo de software donde existe un guía, el cuál revisará el código en proceso de construcción y propondrá mejoras, y el programador, el cual se encargará de la mayor parte del desarrollo. (11)
- Framework: Mayormente entendido como entorno de desarrollo, es un modelo que sirve de guía para la construcción de un proceso genérico o específico, dependiendo de la finalidad del mismo framework. (11)
- Aplicación Web: son las aplicaciones que los clientes finales utilizan como herramientas desde un navegador. (12)

CAPITULO III: METODOLOGÍA

3. METODOLOGÍA

3.1. Metodología aplicada para el desarrollo de la solución

Se aplicó para el desarrollo de la solución, las siguientes metodologías:

Xtreme Programming para la construcción del software:

Durante la construcción del software, esta metodología ágil sirvió de apoyo para la gestión y corrección de errores, ya que se basa en la revisión constante del código de a dos personas, a la vez de su mejor implementación ya que se optimizan los procesos mediante rutinas de pair programming. (13)

Kanban para la gestión de requerimientos y procesos durante el desarrollo y construcción:

En la gestión de requerimientos y procesos del software, se usó la metodología Kanban, la cual administra cada tiempo y ciclo, de cada requerimiento, haciendo un seguimiento completo a los procesos que servirán para la construcción y desarrollo del software. (14)

CAPÍTULO 4: ANÁLISIS Y DISEÑO

4. ANÁLISIS Y DISEÑO DE LA SOLUCIÓN

4.1. Identificación de requerimientos

- El software requiere:

- Módulo de Inicio de Sesión

Historia de Usuario	
Número: 1	Usuario: Cliente
Nombre historia: Iniciar Sesión	
Prioridad en negocio: Media	Riesgo en desarrollo: Baja
Puntos estimados: 2	Iteración asignada: 1
Programador responsable: Luis Borit	
Descripción: Quiero iniciar sesión en el sistema	
Validación: El cliente puede iniciar sesion en el sistema con usuario y contraseña	

- Módulo de Pantalla de Bienvenida

Historia de Usuario	
Número: 2	Usuario: Cliente
Nombre historia: Pantalla de Bienvenida	
Prioridad en negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Luis Borit	
Descripción: Quiero ver la pantalla de bienvenida del sistema	
Validación: El cliente puede ver la pantalla de bienvenida del sistema con exito	

- Módulo de configuraciones

Historia de Usuario	
Número: 3	Usuario: Cliente
Nombre historia: Modulo de configuraciones	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 2
Programador responsable: Luis Borit	
Descripción: Quiero usar el modulo de configuraciones	
Validación: El cliente puede usar el modulo de configuraciones (edicion y actualizacion de datos)	

- Módulo de deudas

Historia de Usuario	
Número: 4	Usuario: Cliente
Nombre historia: Modulo de Deudas	
Prioridad en negocio: Media	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Luis Borit	
Descripción: Quiero usar el modulo de deudas	
Validación: El cliente puede usar el modulo de deudas (reportes)	

- Módulo de registro de Deudas

Historia de Usuario	
Número: 5	Usuario: Cliente
Nombre historia: Modulo de Registro de Deudas	
Prioridad en negocio: Alta	Riesgo en desarrollo: Medio
Puntos estimados: 3	Iteración asignada: 3
Programador responsable: Luis Borit	
Descripción: Quiero usar el modulo de deudas	
Validación: El cliente puede usar el modulo de deudas (registro de datos)	

- Módulo de Lista de Deudas

Historia de Usuario	
Número: 6	Usuario: Cliente
Nombre historia: Modulo de Lista de Deudas	
Prioridad en negocio: Medio	Riesgo en desarrollo: Bajo
Puntos estimados: 2	Iteración asignada: 3
Programador responsable: Luis Borit	
Descripción: Quiero ver la lista de deudas	
Validación: El cliente puede usar el modulo de lista de deudas (lista de datos)	

- Módulo de Edición de Deuda

Historia de Usuario	
Número: 7	Usuario: Cliente
Nombre historia: Modulo de Edicion de Deudas	
Prioridad en negocio: Alto	Riesgo en desarrollo: Alto
Puntos estimados: 3	Iteración asignada: 4
Programador responsable: Luis Borit	
Descripción: Quiero editar una deuda	
Validación: El cliente puede usar el modulo de edicion de deudas (edicion de datos)	

- Módulo de Reportes de Deudas

Historia de Usuario	
Número: 8	Usuario: Cliente
Nombre historia: Modulo de Reportes Especificos de Deudas	
Prioridad en negocio: Alto	Riesgo en desarrollo: Medio
Puntos estimados: 2	Iteración asignada: 4
Programador responsable: Luis Borit	
Descripción: Quiero ver un reporte de deudas especificas	
Validación: El cliente puede usar el modulo de reportes especificos de deudas (reporte de datos)	

PLAN DE TRABAJO

SOFTWARE DE GESTIÓN Y ADMINISTRACIÓN DE DEUDAS SOCIALES

DATOS GENERALES:

PROYECTO.

SOFTWARE DE GESTIÓN Y ADMINISTRACIÓN DE DEUDAS SOCIALES
DE UGEL ILO.

UBICACIÓN DEL PROYECTO.

DEPARTAMENTO DE MOQUEGUA PROVINCIA DE
ILO DISTRITO DE ILO.

ENTIDAD SOLICITANTE.

UGEL ILO
AREA DE ADMINISTRACIÓN

1.- OBJETIVO

El objetivo del servicio es lograr registrar y emitir reportes en base a las deudas sociales existentes y futuras, de la UGEL ILO.

2.- PLAZO DE EJECUCION

El plazo de ejecución del servicio es de un mes.

3.- ACTIVIDADES A DESARROLLAR

a. SEMANA 1

Revisión de documentación y levantamiento de información fuente en las áreas involucradas.

b. SEMANA 2

Diseño lógico de los módulos conformantes del Sistema integrado.

Diseño físico de los módulos conformantes del Sistema Integrado y establecimiento de las Bases de Datos.

Programación en JavaScript de los módulos conformantes del Sistema integrado, esto incluye la configuración de las Bases de Datos.

d. SEMANA 3

Prueba de vacío y corrección de errores

e. SEMANA 4

Instalación y entrega de documentación.

4.- REQUERIMIENTO DE INFOMACION.

Se deberá otorgar las facilidades del caso para poder tener acceso a la información que cubra las especificaciones.

5.- PRODUCTO FINAL.

Software de gestión y administración de deudas sociales de la UGEL ILO.

- Registro de Configuraciones
- Registro de Deudas
- Reportes Generales
- Reportes Específicos

6.- RECURSOS A UTILIZAR.

Los recursos a utilizar serán asumidos íntegramente por el presentador del servicio y consta de:

- 01 laptop para el desarrollo de los módulos.
- 01 impresora laser.
- Suministros diversos.

7.- COSTO ELABORACION DEL SOFTWARE.

La elaboración del software es S/. 3,500 nuevos soles.

8.- CRONOCRAMA DE EJECUCION.

El cronograma de ejecución del servicio es el siguiente:

Ítem	Actividad	01	02	03	04
1	Revisión de documentación y levantamiento de información fuente	■			
2	Diseño Lógico, Diseño Físico y base de datos	■	■		
3	Desarrollo de Programas			■	
4	Prueba de vacíos y corrección de errores				■
5	Instalación y puesta en marcha				■

Evidencias de Requerimiento: Modelo de Plan de Trabajo usado por la entidad que requirió el servicio de desarrollo del software (UGEL-ILO)

CARTA DE GARANTIA

SRS. UGEL ILO
RESPONSABLE DEL ÁREA DE INFORMÁTICA

Presente.-

Por intermedio del presente me es grato presentarle mis saludos y a la vez hacerle llegar la Carta de Garantía sobre el software desarrollado, la cual se desglosa en.

Del Software.

SOFTWARE DE GESTIÓN Y ADMINISTRACIÓN DE DEUDAS SOCIALES

De la Garantía.

Se otorga la **Garantía** contra defectos de desarrollo y operación derivada directamente del SOFTWARE DE GESTIÓN Y ADMINISTRACIÓN DE DEUDAS SOCIALES por un periodo de 12 meses calendario, los cuales entraran en vigencia a partir de la recepción de la presente carta.

De la Resolución de la Garantía.

La presente carta de garantía quedará resuelta en los siguientes casos:

- Por cumplimiento de los 12 meses como periodo pactado de la garantía.
- Por la manipulación de los programas fuentes sin autorización durante el periodo de garantía.

Me despido de Ud. Brindándole los saludos de mi mayor consideración y estima personal.


BORIS GUILLÓN LUIS ALBERTO
R.U.C.: 10701361976

C.L.A/03/00

Evidencia del Requerimiento de Servicio de Desarrollo de Software: Carta de Garantía Entregada a la entidad solicitante

CARTA DE DERECHO DE AUTOR

Sres. UGEL ILO
RESPONSABLE DEL ÁREA DE INFORMÁTICA

Presente.-

Por intermedio del presente me es grato presentarle mis saludos y a la vez hacerle llegar la Carta de Sesión de Derechos de Autor sobre el software desarrollado, la cual se desglosa en.

Del Software.
SOFTWARE DE GESTIÓN Y ADMINISTRACIÓN DE DEUDAS SOCIALES

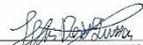
De los Derechos de Autor.
Se otorga los Derechos de Autoría a la UGEL de Ilo, sobre SOFTWARE DE GESTIÓN Y ADMINISTRACIÓN DE DEUDAS SOCIALES, con la finalidad que pueda Registrar el software en mención como **PROPIO** ante las instancias correspondientes.

Del Alcance.

Los Derechos derivados a la UGEL de Ilo, corresponde a.

- Código Fuente
- Base de datos
- Demás librerías que comprende a la estructura del software.

Me despido de Ud. Brindándole los saludos de mi mayor consideración y estima personal.


GONZALO LUIS ALBERTO
R.U.C.: 10701361976

C:\Archivo

Evidencia del Requerimiento de Servicio de Desarrollo de Software: Carta Cede de Derechos de Autor a la entidad solicitante

09 de marzo de 2018

Acta de Capacitación

Yo Flor María Gonzales Cerpa, con DNI 00799428, afirmo que estuve presente en la primera capacitación y exposición del "Software de Gestión y Administración de Deudas Sociales" dada por Luis Alberto Borit Guitton de la empresa Borit Labs con RUC 10701361976, donde participaron la, en ese entonces administradora, Rosa Elena Rosales, y el director de la UGEL ILO Isidro Estrada Canavire, y afirmo que hoy 09 de marzo de 2018 en la segunda capacitación del sistema antes mencionado, también estuve presente.


Sra. Flor M. Gonzales Cerpa
Secretaria
UGEL ILO
FIRMA
DNI 00799428

Evidencia del Requerimiento de Servicio de Desarrollo de Software: Acta de Capacitación al personal de la entidad solicitante

4.2. Análisis de la solución

Módulo de Inicio de Sesión:

Ingresamos nuestro usuario y contraseña, luego hacemos click en el botón de enviar.



Figura 1 - Inicio de Sesión

Módulo de Pantalla de Bienvenida:

Tenemos un menú superior, donde indica el cierre de sesión, el Módulo de configuraciones, y el Módulo de deudas.

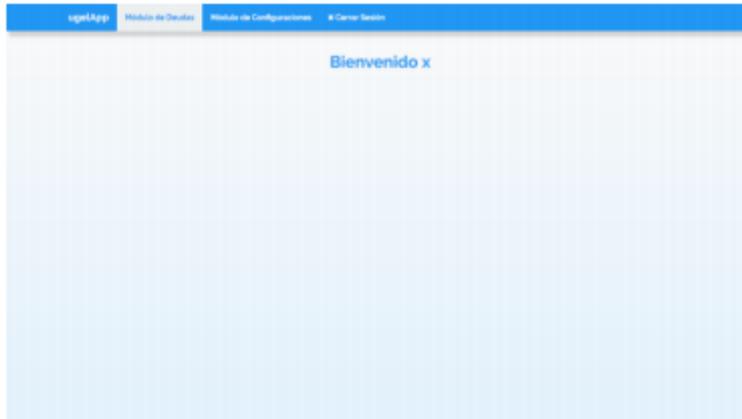


Figura 2 – Pantalla de Inicio

Módulo de configuraciones:

En este módulo podemos agregar las siguientes configuraciones:

- Tipo de Servidor
- Situación Laboral
- Tipo de Deuda
- Concepto de Deuda
- Datos del Deudor



Figura 3 - Configuraciones

Módulo de deudas:

En éste módulo contamos con tres submenús, donde indican:

- Módulo de registro de deudas
- Módulo de Lista de Deudas
- Módulo de Reportes de Deudas

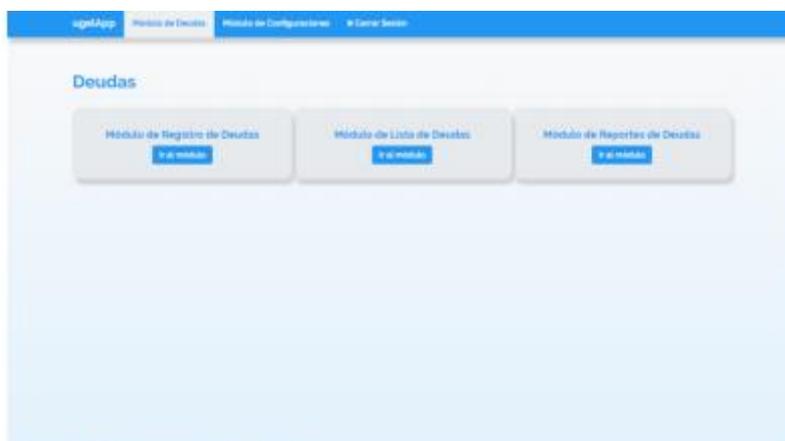


Figura 4 - Modulo de Deudas

Módulo de registro de Deudas:

En este módulo podemos ingresar la deuda, llenando los datos indicados.

Figura 5 - Registro de Deudas

Módulo de Lista de Deudas:

En este módulo podremos visualizar la totalidad de registros para su reporte específico y edición posterior.

Nombre	Apellido Paterno	Apellido Materno	Número de Resolución	Monto Deuda	Expediente Judicial	Fecha	Documento	Opcional 1
Ordóñez	Paz	Guerra	0	0.00	0	00	000	0000

Figura 6 - Lista de Deudas

Módulo de Edición de Deuda:

En este módulo podremos modificar el registro mediante sus campos.

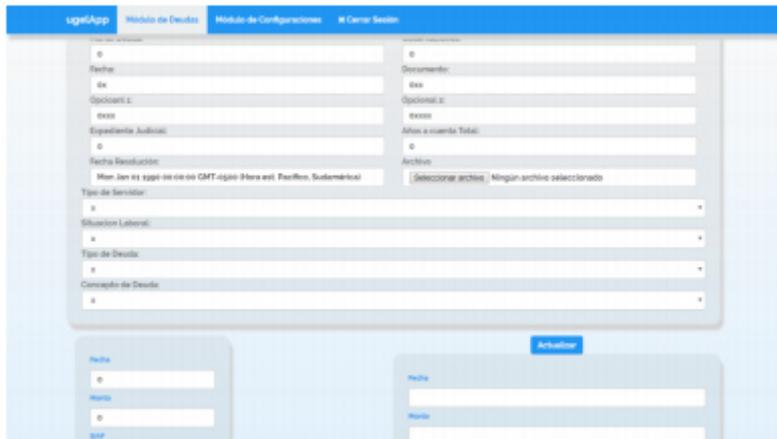


Figura 7 - Edición de Deuda

Módulo de Reportes de Deudas:

En éste módulo podemos emitir y exportar reportes en Excel, con tan solo especificar los múltiples campos en los filtros con la tecla CONTROL.

También podemos crear reportes específicos, haciendo click en las cabeceras del reporte que queremos que no aparezcan en el reporte final.

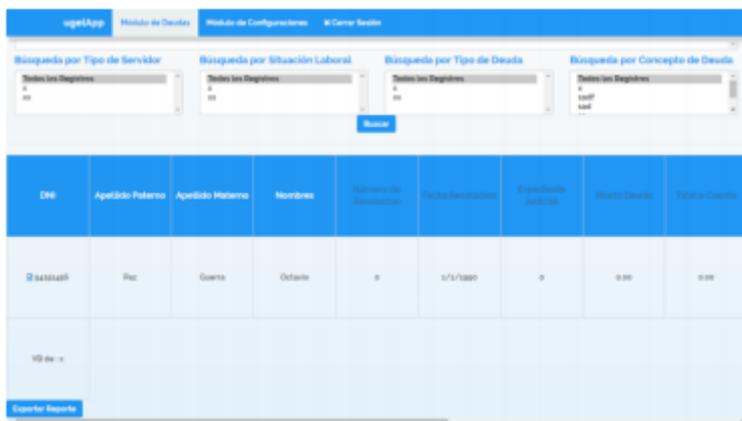


Figura 8 - Reportes de Deuda

4.3. Diseño

Diagrama de Base de Datos no Relacional:

En base a los requerimientos del sistema de información, el diseño del diagrama mostrado en la Figura 9, se construyó creando tablas enlazadas de acuerdo a la necesidad de la construcción futura, añadiendo campos específicos, y sus tipos de datos para la posterior y correcta funcionalidad del sistema.

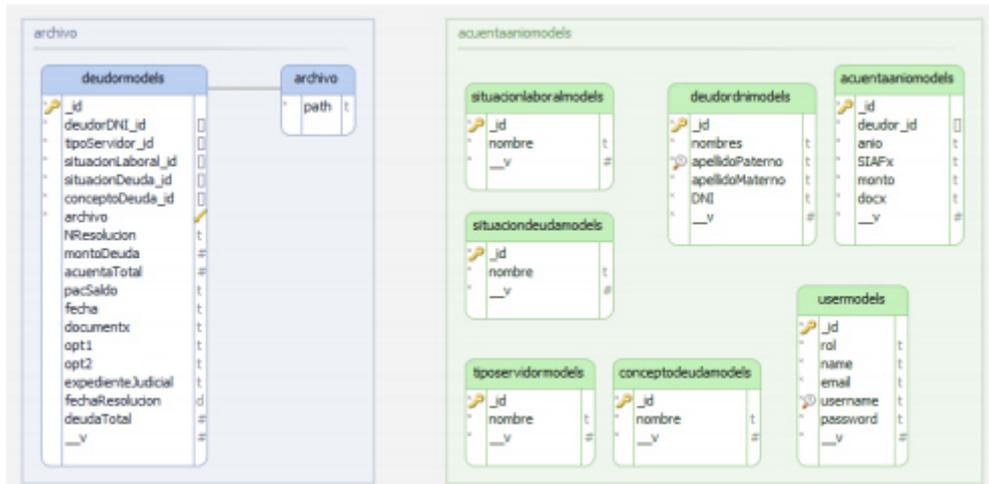


Figura 9 - Diagrama de Base de Datos

CAPÍTULO V: DISEÑO

5. Construcción

5.1. Desarrollo

5.1.1. Codificación de modelos en JavaScript

/acuentaAnio.js

```
const mongoose = require('mongoose');

const deudorModel = require('../models/deudor');
// tipoServidor Schema
const acuentaAnioSchema = mongoose.Schema({
  anio: {
    type: String,
  },
  monto: {
    type: String,
  },
  SIAFx: {
    type: String,
  },
  docx: {
    type: String,
  },
  deudor_id : [{type: mongoose.Schema.Types.ObjectId, ref: 'deudorModel'}],
});

const acuentaAnioModel = module.exports = mongoose.model('acuentaAnioModel', acuentaAnioSchema);
```

/conceptoDeuda.js

```
const mongoose = require('mongoose');

// tipoServidor Schema
const conceptoDeudaSchema = mongoose.Schema({
  nombre: {
    type: String
  }
});

const conceptoDeudaModel = module.exports = mongoose.model('conceptoDeudaModel', conceptoDeudaSchema);
```

/deudor.js

```
const mongoose = require('mongoose');
const tipoServidorModel = require('../models/tipoServidor');
const situacionLaboralModel = require('../models/situacionLaboral');
const situacionDeudaModel = require('../models/situacionDeuda');
const conceptoDeudaModel = require('../models/conceptoDeuda');
const deudorDNIModel = require('../models/deudorDNI');

// User Schema
const DeudorSchema = mongoose.Schema({
  deudorDNI_id : [{type: mongoose.Schema.Types.ObjectId, ref: 'deudorDNIModel'}],
  tipoServidor_id : [{type: mongoose.Schema.Types.ObjectId, ref: 'tipoServidorModel'}],
  situacionLaboral_id : [{type: mongoose.Schema.Types.ObjectId, ref: 'situacionLaboralModel'}],
  situacionDeuda_id : [{type: mongoose.Schema.Types.ObjectId, ref: 'situacionDeudaModel'}],
  conceptoDeuda_id : [{type: mongoose.Schema.Types.ObjectId, ref: 'conceptoDeudaModel'}],
  NResolucion: {
    type: String
  },
  fechaResolucion: {
    type: Date,
  },
  montoDeuda: {
    type: Number
  },
  pacSaldo: {
    type : String
  },
  acuentaTotal: {
    type: Number
  },
  expedienteJudicial: {
    type: String
  },
  deudaTotal: {
    type: Number
  },
  archivo : {path: String},
  fecha: {
    type : String
  },
  documentx: {
    type : String
  },
  opt1: {
    type : String
  },
  opt2: {
    type : String
  },
});

const DeudorModel = module.exports = mongoose.model('DeudorModel', DeudorSchema);
```

/deudorDNI.js

```
const mongoose = require('mongoose');

// User Schema
const deudorDNISchema = mongoose.Schema({
  apellidoPaterno: {
    type: String,
    index: true
  },
  apellidoMaterno: {
    type: String
  },
  nombres: {
    type: String
  },
  DNI: {
    type: String
  }
});

const DeudorModel = module.exports = mongoose.model('deudorDNIModel', deudorDNISchema);
```

/situacionDeuda.js

```
const mongoose = require('mongoose');

// tipoServidor Schema
const situacionDeudaSchema = mongoose.Schema({
  nombre: {
    type: String
  }
});

const situacionDeudaModel = module.exports = mongoose.model('situacionDeudaModel', situacionDeudaSchema);
```

/situacionLaboral.js

```
const mongoose = require('mongoose');

// tipoServidor Schema
const situacionLaboralSchema = mongoose.Schema({
  nombre: {
    type: String
  }
});

const situacionLaboralModel = module.exports = mongoose.model('situacionLaboralModel', situacionLaboralSchema);
```

/tipoServidor.js

/tipoServidor.js

```
const mongoose = require('mongoose');

// tipoServidor Schema
const tipoServidorSchema = mongoose.Schema({
  nombre: {
    type: String,
  }
});

const tipoServidorModel = module.exports = mongoose.model('tipoServidorModel', tipoServidorSchema);
```

/user.js

```
// User Schema
const UserSchema = mongoose.Schema({
  username: {
    type: String,
    index:true
  },
  password: {
    type: String
  },
  email: {
    type: String
  },
  name: {
    type: String
  },
  rol: {
    type: Number, default:3,
  }
});

const UserModel = module.exports = mongoose.model('UserModel', UserSchema);

module.exports.createUser = (newUser, callback) => {
  bcrypt.genSalt(10, (err, salt) => {
    bcrypt.hash(newUser.password, salt, (err, hash) => {
      newUser.password = hash;
      newUser.save(callback);
    });
  });
});

module.exports.getUserByUsername = (username, callback) => {
  var query = {username: username};
  UserModel.findOne(query, callback);
}

module.exports.getUserById = (id, callback) => {
  UserModel.findById(id, callback);
}

module.exports.comparePassword = (candidatePassword, hash, callback) => {
  bcrypt.compare(candidatePassword, hash, (err, isMatch) => {
    if(err) throw err;
    callback(null, isMatch);
  });
}
```

5.2. Pruebas y resultados

Prueba\Aprobacion	Defectuoso	Con dificultad	Intermedio	Comprensible	Muy Simple
Pudo usar el Inicio de sesión sin dificultades				x	
Demoró en cargar el sistema tras iniciar sesión			x		
Cree que la Interfaz es amigable y sencilla				x	
Ubicó los botones que activan las funcionalidades			x		
Sirvió de apoyo a la toma de decisiones				x	
Brindó soporte a las labores administrativas			x		
Resulta escalable de acuerdo a las necesidades				x	
Funciona correctamente			x		

Tabla #1 – Pruebas de Funcionalidad

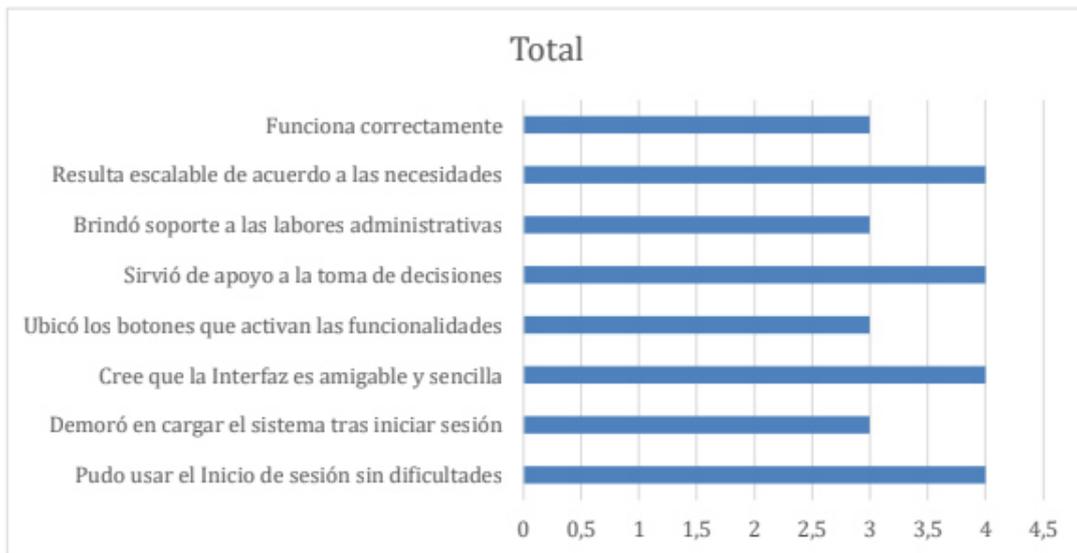


Tabla #2 – Pruebas de Funcionalidad Tabuladas

Conclusiones

Según los resultados obtenidos en la Tabla#2 se concluye que, el sistema de información construido a partir de los requerimientos específicos recolectados, tiene un margen de aceptación de [Intermedio - Comprensible] según los parámetros de la matriz de aceptación de la Tabla #1.

Este margen de aceptación es ideal, ya que no se inclina al límite superior, el cual indica una sobre expresión del entendimiento, sobrepasando la funcionalidad por un estado de inutilidad por la simpleza de este mismo parámetro.

Al igual que supera al margen inferior, y al subsiguiente, logrando así estar entre los parámetros de Intermedio Entendimiento y Entendimiento Comprensible.

Trabajos Futuros

Ante una futura implementación de módulos relacionados al sistema de información construido en este proyecto de implementación, se deberá tener en cuenta lo siguiente:

- La flexibilidad de añadir y remover campos en las tablas de la estructura de la base de datos. - La modificación de alguna instancia no debería afectar las rutinas de los

procesos de inicialización del servidor. - La experiencia del usuario en cuanto a la interfaz tiene que ser óptima para el

libre desarrollo de las actividades del personal administrativo.

Referencias

1. **MIT.** *System Software*.
EEUU : MIT, 2001.
2. **Christof Ebert, Jürgen Salecker.** *Embedded Software—Technologies and Trends*. EEUU : IEEE, 2013.
3. **Niquette, Paul.** *Softword: Provenance for the Word 'Software'*.
2006. 1-58922-233-4.
4. **DÍAZ PÉREZ, PALOMA, MONTERO, SUSANA and AEDO CUEVAS, IGNACIO.** *Ingeniería de la web y patrones de diseño*. Madrid : Pearson Educación, 2005.
5. **Haverbeke, Marijn.** *Eloquent Javascript: Autoedición*. 2018.
6. **Barzanallana.** *Historia del desarrollo de aplicaciones Web*. Murcia : Universidad de Murcia, 2012.
7. **Valdés, D. Perez.** *Los diferentes lenguajes de programación para la web*. 2007.
8. **Ochando, M. Blázquez.** *Fundamentos y Diseño de Bases de Datos*. 2014.
9. **Alvarez, S.** *Introducción a las Bases de Datos*. 2007.
10. **Hosting, Ok.** *Metodologías de Desarrollo de Software*. s.l. : Ok Hosting, 2016.
11. **EcuRed.** *Metodologías de desarrollo de Software*.
s.l. : EcuRed, 2018.
12. **Floyd, C.** *Theory and Practice of Software Development*. Hamburg : University of Hamburg.
13. **Informatics-UK.** *Human Centred Technology Workshop*. 2005.
14. **Waldner, Jean-Baptiste.** *Principles of Computer-Integrated Manufacturing*. London : s.n., 1992. 0-471-93450-X.