



UNIVERSIDAD NACIONAL DEL ALTIPLANO
ESCUELA DE POSGRADO
MAESTRÍA EN INFORMÁTICA



TESIS

**IMPLEMENTACIÓN DE UN SISTEMA DE REGISTROS NOTARIALES EN
EL ARCHIVO REGIONAL DE PUNO 2012-2016**

PRESENTADA POR:

EDGAR APAZA CHOQUE

PARA OPTAR EL GRADO ACADÉMICO DE:

MAGISTER SCIENTIAE EN INFORMÁTICA

**MENCIÓN EN GERENCIA DE TECNOLOGÍAS DE INFORMACIÓN Y
COMUNICACIONES**

PUNO, PERÚ

2019



DEDICATORIA

Dedicado a mi familia por todo su apoyo y
comprensión. A mi esposa Jenny e hijos
Marjory, Adriana, Harry a mis padres
Leonidas y Glicería

Edgar



AGRADECIMIENTOS

- A la Universidad Nacional del Altiplano y a todos sus docentes por impartir conocimiento y dedicación hacia mi persona, siempre con palabras de aliento y deseos de superación.
- Gracias a Dios por permitirme poder disfrutar de esta vida tan maravillosa. A todos mis docentes de la UNA Puno, quienes, con sapiencia, lograron inculcar una profunda búsqueda de conocimiento en favor de los demás.
- Al personal de Archivo Regional de Puno por la oportunidad de trabajar con ellos y aportar con mi granito de arena al engrandecimiento de esta noble institución y permitiéndome poder desarrollar e implementar este software en esta institución.
- A mi familia por todo el apoyo y cariño brindado por todos los años de estudio comenzando por mi esposa Jenny, mis hijos Marjory, Adriana, Harry a mis padres Leonidas y Glicería, a mis suegros Gabino y Placida, a todos mis demás familiares quienes han estado dispuestos a colaborar en este sueño que ya está cumplido.



ÍNDICE GENERAL

	Pág.
DEDICATORIA	I
AGRADECIMIENTOS	II
ÍNDICE GENERAL	III
ÍNDICE DE TABLAS	V
ÍNDICE DE FIGURAS	VII
ÍNDICE DE ANEXOS	XI
RESUMEN	XII
ABSTRACT	XIII
INTRODUCCIÓN	1

CAPÍTULO I

REVISIÓN DE LITERATURA

1.1 Marco teórico	4
1.2 Antecedentes	10

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1 Identificación del problema	16
2.2 Enunciados del problema	17
2.3 Justificación	17
2.4 Objetivos	19
2.4.1 Objetivo general	19
2.5 Hipótesis	19
2.5.1 Hipótesis general	19

CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 Lugar de estudio	20
3.2 Población	20



3.3 Muestra	24
3.4 Metodología de investigación	25
3.5 Descripción detallada de métodos por objetivos específicos	27

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

CONCLUSIONES	127
RECOMENDACIONES	130
BIBLIOGRAFÍA	132
ANEXOS	135

Puno, 05 de abril de 2019

ÁREA: Investigación científica y tecnología.
TEMA: Sistemas de información.



ÍNDICE DE TABLAS

	Pág.
1. Servicios archivísticos 2001 al 2013	18
2. Distribución del personal por unidades orgánicas	21
3. Numero Notarios por Provincias	22
4. Fondos documentales y colecciones - Archivo Intermedio	22
5. Fondos documentales y colecciones - dirección Archivo Histórico	23
6. Recursos necesarios para el Inicio del Proyecto: Fondo Notarial	28
7. Funciones del sistema Búsqueda	54
8. Pila de productos Inicial para el ingreso Notarial	64
9. Pila de productos para el desarrollo del sistema de Búsquedas	65
10. Historia de usuarios para búsqueda de información	66
11. Validación de usuarios en el sistema	66
12. Redireccionamiento de usuarios según el código de acceso	67
13. Estandarización de código	85
14. Pila de producto sprint 1- ventana del administrador	86
15. Sprint 1 – Módulo de Ingresos al sistema y Seguridad	88
16. Sprint 2 – Módulo de Ingresos de Escrituras con MVC	89
17. Sprint 3 – Módulo de Reportes diarios de ingreso de escrituras	90
18. Finalización de procesos de desarrollo	91
19. Tabla Usuarios de la Base de Datos	94
20. Tabla Proyecto de la Base de Datos	94
21. Tabla de Notarios de la Base de Datos	95
22. Tabla escrituras de la Base de Datos	95
23. Tabla otorgantes – escritor de la Base de Datos	96
24. Tabla Favorecidos – escritfavor de la base de datos	96
25. Tabla de personas involucradas en las escrituras	96
26. Tabla Involucrados Jurídicos	96
27. Tabla Serie	97
28. Tabla Sub series	97
29. Ingreso de Escrituras correspondiente al año 2010	112
30. Ingreso de Escrituras correspondientes al año 2011	113
31. Ingreso de Escrituras correspondientes al año 2012	114



32.	Ingreso de Escrituras correspondientes al año 2013	115
33.	Ingreso de Escrituras correspondientes al año 2014	116
34.	Ingreso de Escrituras correspondientes al año 2015	117
35.	Ingreso de Escrituras correspondiente al año 2016	118
36.	Ingreso de Escrituras correspondientes al año 2017	119
37.	Número total de escrituras ingresadas por años y Total general	119
38.	Listado del personal involucrado en el Proyecto Fondo Notarial	121
39.	Tabla para la asignación de perfiles de usuarios y accesos al sistema	123
40.	Datos estadísticos recogidos de la Base de Datos del sistema Fondo notarial	124
41.	Número de solicitudes atendidas por años en el ARP 2011 al 2016	124

ÍNDICE DE FIGURAS

	Pág.
1. Modelo del Sistema de Red en el AR Puno	29
2. Modelo de la ventana de Ingreso – Panel de administración principal	30
3. Modelo de la Base de Datos – Sistema Fondo Notarial	31
4. Ingreso a Vagrant en Windows 7, usando el software Git Bash	32
5. Cargando y arrancando Vagrant usando la máquina virtual “machine1”.	33
6. Ingresando a la máquina virtual Machine1	33
7. La Figura muestra como inicia el Vagrant	34
8. Configurando los parámetros de inicio para la comunicación de datos	34
9. Figura que muestra e indica que la Máquina virtual está corriendo (Running)	35
10. Figura que indica que el servidor está corriendo	35
11. Pantalla de inicio del Buscador de Escrituras públicas.	36
12. Figura que muestra como poder acceder al sistema de búsqueda	36
13. Botones para buscar, por nombre, datos jurídicos, búsqueda por fecha	37
14. Resultado de la búsqueda (Ejemplo: Apellido paterno castellanos)	38
15. Lista de casos cuando estos Venden (Otorgantes) y cuando adquieren un bien (Favorecidos)	38
16. Búsqueda por nombres de Personas o empresas de carácter jurídico	39
17. Muestra la información del resultado de la búsqueda por persona jurídica	39
18. Ejemplo de búsqueda por fecha	40
19. Detalle de los datos de la escritura	40
20. Ingresando a la máquina virtual usando el protocolo SSH	41
21. La Figura muestra la versión del servidor que está configurado y la versión de Ubuntu	42
22. Muestra las versiones del servidor APACHE, PHP Y MySQL (MariaDB)	42
23. Vista del Servidor Ubuntu server dentro de vagrant	43
24. La Figura muestra que el servidor apache2 está activo y corriendo	43
25. La configuración de las interfaces de red en Ubuntu	44
26. Acceso al software usando la dirección del servidor	44



27.	Ejecutando el instalador del Sistema Fondo Notarial	45
28.	Vista del Asistente de Instalación del Software Notarial	45
29.	Ingreso de configuración para la aplicación	46
30.	Vista de Establecer conexión al sistema	46
31.	Pantalla inicial del software de ingreso de escrituras	47
32.	Pantalla de ingreso de escrituras públicas	47
33.	Vista del sistema GIT para el control de Versiones – Sistema Fondo Notarial	49
34.	Diagrama de ingreso al sistema (Modulo de Seguridad)	56
35.	Casos de uso del módulo de búsquedas	57
36.	Diagrama de paquete. Buscar Escrituras	57
37.	Diagrama de paquetes para búsqueda	58
38.	Diagrama de Casos de Uso – Trabajador búsquedas	59
39.	Diagrama de Secuencia. Búsqueda por Comprador	60
40.	Diagrama de Secuencia. Búsqueda por Vendedor	61
41.	Ejecutando el Git Bash y ubicarnos en la carpeta Vagrant/machine1	68
42.	Se muestra como inicia el Vagrant con la máquina virtual en Linux	68
43.	Uso de Vagrant SSH, para ingresar a la maquina en Linux	69
44.	Dentro del servidor e ingresando a la carpeta del proyecto	69
45.	Muestra del control de código y de cumplimiento de los Sprint con Git	70
46.	Reglas de SCRUM	71
47.	Diagrama de Clases. Clase Conexión	72
48.	Diagrama de Clase. Crear Grupos de Ingreso	73
49.	Sistema de Ingreso Fondo Notarial	73
50.	Diagrama de Paquetes. Administrador del Sistema	74
51.	Diagramas de Secuencia. Generar Estadísticas Mensuales	75
52.	Diagramas de Secuencia. Generar Estadísticas Mensualizadas por Persona	76
53.	Diagrama. Estadísticas Mensualizadas por persona	76
54.	Diagramas de Secuencia. Corrección de Ingresos a Base de Datos por Código	77
55.	Diagramas de Secuencia. Corrección de Ingreso a Base de Datos por Escrituras y Protocolo	78



56.	Diagramas de Secuencia. Administración de Grupos	79
57.	Diagramas de Secuencia. Estadísticas de Grupos	79
58.	Diagramas de Secuencia. Agregar Trabajador	80
59.	Diagrama de Secuencia. Agregar Notario	80
60.	Diagramas de Secuencia. Visualizar Ingresos Diarios por Persona	81
61.	Módulos de Administración de Grupos	81
62.	Diagramas de Secuencia. Corrección de Ingresos a Base de Datos por Código	82
63.	Diagramas de Secuencia. Corrección de Ingreso a Base de Datos por Escrituras y Protocolo	83
64.	Diagramas de Secuencia. Estadísticas de Grupos	84
65.	Diagrama de secuencias, Consulta de Escrituras	84
66.	Vista de todas las Tablas del Sistema Fondo Notarial ARP	93
67.	En la Figura se muestra todas las Tablas de la Base de Datos “dbarp”	98
68.	La figura muestra una vista en MySQL para el listado de Notarios	98
69.	La figura muestra las otras vistas utilizadas en MySQL	99
70.	La Figura muestra la lista de los usuarios registrados	99
71.	Muestra de la Tabla escrituras	100
72.	La Figura muestra la lista de los NOTARIOS	100
73.	La Figura muestra la Tabla “fondos documentales”	101
74.	La Figura muestra el listado de la Tabla “escritor” (Otorgantes Escrituras)	101
75.	La Figura muestra la lista de la Tabla “involjuridicas1” (Involucrados Jurídicos)	102
76.	Las Figura presenta la lista de proyectos creados con el software	102
77.	Vista del código del programa escrito en SublimeText3, archivo index.html	104
78.	Árbol de carpetas utilizadas y se aprecia el MVC (Model, View, Controller)	105
79.	Vista del árbol de carpetas y los permisos por cada carpeta y archivo	106
80.	La carpeta Coreapp, conexión directa con la Base de Datos	106
81.	Vista del código de búsqueda usando Clases en PHP	107
82.	Vista de la Clase Listado	107



83.	Vista del archivo CheckListadoNew.php	108
84.	Vista del Archivo de detalles de Escrituras	108
85.	Vista de reportes e ingreso a Base de Datos por fechas.	109
86.	Menú principal del módulo de Administración	111
87.	Ventana de reportes y gráficos estadísticos de la aplicación por años	111
88.	Figura del reporte de ingresos correspondientes al año 2013	112
89.	Ingreso de Escrituras por años 2010- 2017	120
90.	Número de solicitudes recibidas para búsqueda por años	125



ÍNDICE DE ANEXOS

	Pág.
1. Código fuente principales características	1365

RESUMEN

El presente trabajo de investigación tiene por finalidad desarrollar un sistema de Registros Notariales usando un sistema de virtualización Vagrant (Docker) para acelerar la búsqueda y recuperación de información notarial y de otros fondos documentales (expedientes judiciales, partida de nacimiento, entre otras) en el Archivo Regional de Puno, del distrito, provincia y departamento del mismo nombre; almacenándolas en una base de datos, con los documentos digitalizados de cada escritura. El Sistema denominado “Fondo Notarial” consta de 3 módulos: 1) Modulo de Ingreso de escrituras, 2) Modulo de Búsqueda 3) Modulo de Administración y corrección. La metodología utilizada para el desarrollo de este software fue la metodología XP para el desarrollo del software y la metodología SCRUM para la gestión del equipo de desarrollo y mejora de la calidad, utilizando un sistema de control de versiones denominado Git. Los resultados obtenidos de la implementación de este software han sido muy satisfactorios, ya que, en el primer año de su implementación, se logró ingresar a esta Base de Datos 42,370 escrituras y hasta el año 2017 se tiene 369,724 escrituras ingresadas, incrementándose este número cada año. Para el desarrollo de este proyecto se viene trabajando con más de 75 personas, entre personal nombrado y contratado, siendo el primer Archivo Regional a nivel del Perú en implementar este modelo de innovación tecnológica, siendo reconocidos como una de las mejores instituciones a nivel nacional por el AGN (Archivo General de la Nación) y denominado como institución piloto en gobierno electrónico por el Gobierno Regional de Puno.

Palabras clave: Archivo regional, información notarial, institución piloto. módulos, registro, software.



ABSTRACT

The present research work has for purpose to develop a system of Notarial Registries using a virtualization system Vagrant (Docker) to accelerate the search and retrieval of notarial information and other documentary funds (court records, birth certificate, among others) in the Regional Archive of Puno, of the district, province and department of the same name. storing them in a database, with the digitized documents of each writing. The System called "Notarial Fund" consists of 3 modules: 1) Entry Module deeds, 2) Search Module 3) Administration and Correction Module. The methodology used for the development of this software was the XP methodology for software development and the SCRUM methodology for managing the development and quality improvement team, using a version control system called Git. The results obtained from the implementation of this software have been very satisfactory, since in the first year of its implementation, 42,370 deeds were entered into this database and until 2017 there are 369,724 deeds entered, increasing this number every year. For the development of this project, more than 75 people have been working, among appointed and hired personnel, being the first Regional Archive at the level of Peru to implement this technological innovation model, being recognized as one of the best institutions nationwide by the AGN (General Archive of the Nation) and named as a pilot institution in electronic government by the Regional Government of Puno.

Keywords: Notarial information, modules, pilot institution, regional archive, registry, software.

INTRODUCCIÓN

Nuestra investigación se basa en un problema que atravesaba el Archivo Regional de Puno, que todo trabajo que se realizaba en esta institución era manual, desde la recepción de solicitudes de búsqueda, las búsquedas, registros de atención a usuarios, entre otras actividades, se hacían a mano, en formatos de hojas mimeografiados. Para la época del 1983 hasta el 2006 (23 años de funcionamiento) la cantidad de usuarios no pasaba los 3,000 usuarios al año, siendo las cantidades de solicitudes de búsqueda pequeñas, se abastecían casi sin problemas. En los siguientes años, la cantidad de transferencias de documentos notariales se incrementa de manera muy notoria, y la cantidad de público empieza a crecer, debido a que varios notarios antiguos ya habían fallecido, y los documentos de estos ahora se encuentran en poder el Archivo Regional de Puno, y por lo tanto las personas deben acudir a esta institución, para solicitar una copia de testimonios de los notarios ya fallecidos, llegando a un punto donde la forma de trabajar era inviable, requiriendo una nueva forma de trabajo más ágil y rápida, para lo cual se ha elaborado este proyecto en base a las proyecciones y estudios realizados en estos años.

El Archivo Regional de Puno es un organismo desconcentrado del Gobierno Regional de Puno, depende administrativamente de la Gerencia Regional de Desarrollo Social; técnica, funcional y normativamente del Archivo General de la Nación, órgano rector y central del Sistema Nacional de Archivos. Está encargado de la defensa, conservación y servicio del Patrimonio Documental de la Nación en la región (Archivo Regional de Puno, 2015), siendo una institución única debido a la naturaleza de sus funciones. Esta guarda documentos desde el Siglo XVI, con cerca de 2 Km de documentación, almacenada en repositorios y estantería. Siendo una institución única en su clase, también presenta muchas debilidades y/o falencias en su forma de trabajo, debido a que la mayoría de trabajos se realizaban de manera manual, y desde que comenzamos a trabajar en esta institución, se han puesto en marcha varias iniciativas de mejoras, pero una de los más importes es el ingreso de los registros notariales (testimonios, testamentos, compra venta, etc.) a un sistema que nos permita ubicar las escrituras de manera más rápida, junto con otros software desarrollados como: Sistema de Recepción de solicitudes, donde ya no se duplica la información de un usuario y con los datos que proporcionan las personas sobre su escritura pública, el sistema guarda esta información e imprime una hoja de solicitudes

con diferentes formatos y usos, y un sistema de Caja que registra los pagos realizados por las personas y permite tener un control más detallado de los ingresos, que son reportados al gobierno regional. Aunque son mejoras implementadas, también se han realizado varias jornadas de capacitación en el uso de los nuevos sistemas, ofimática, redes e internet entre otras, siendo este proyecto solo una de las mejoras y cambios que ha sufrido esta institución a lo largo de casi 10 años de trabajo.

La cantidad de documentos notariales se incrementa año tras año, debido a que cuando un notario fallece o deja de ejercer como notario público, todos los actuados (protocolos, libros, registros, índices) pasan automáticamente por 2 años al colegio de Notarios de Puno. Una vez concluido este tiempo, todos los documentos del ex notario pasan de manera permanente al archivo regional donde se conservarán de manera perpetua. Así que, en el archivo regional de Puno, tiene los documentos de los ex notarios de las provincias de Puno, San Román, Azángaro, Carabaya, Chucuito, El Collao, Huancané, Lampa, Melgar, Moho, San Antonio de Putina, Sandia y Yunguyo; todas estas son celosamente custodiadas y puestas al servicio de los usuarios que la soliciten, previa identificación y pago del TUPA aprobado por el gobierno regional.

Sobre esta realidad y a la gran cantidad de información que se custodia, es que se ha diseñado, analizado y codificado un Software que permita ingresar toda esta información de los Notarios mediante un sistema informático, almacenado en una base de datos, junto con un buscador que permita la rápida identificación de una escritura. Haciendo uso de metodologías de software ágil, es que este sistema ha pasado por varias revisiones y correcciones (versiones del software), hasta alcanzar una versión estable, y los cambios son más de apariencia del software, sin que este afecte al funcionamiento del programa.

Los resultados han sido muy satisfactorios y la respuesta de los trabajadores ha sido una de las mejores, debido a que el software no solo facilita la ubicación, sino que muestra también algunos errores que al momento del vaciado de la información al sistema se han anotado en un campo información, lo que posibilita que el “Calificador” —que es una persona (usualmente con un conocimiento en derecho) con un alto grado de conocimiento legal y archivístico— pueda en base a su función, amparado bajo los reglamentos que el AGN (Archivo General de la Nación) y otras que existen en nuestra constitución, validar o no la escritura pública como tal, debido a que una de las funciones de este trabajador es que deba leer documento en su totalidad, y de encontrar algún vicio legal, o el no ser capaz de identificar las firmas de los actuados, o encontrar algún indicio de falsedad o



datos faltantes, dentro de sus prerrogativas, está el informar al solicitante que su documento tiene algún problema legal y se lo deriva a asesoría jurídica para las aclaraciones del caso.

Este software proporciona también una parte estadística que permite ver en tiempo real los ingresos diarios y también los de días anteriores, meses y años. Presenta reportes con el fin de evaluar el rendimiento del equipo y los entregables mensuales que deben presentarse, que el mismo tiempo también se informan al gobierno regional como al AGN.

En los siguientes capítulos, se presentará la información a nivel de desarrollo de software haciendo uso de la metodología XP y SCRUM para varias partes del proyecto, y su funcionalidad; así como muestras del código fuente más importante, reportes del sistema, el sistema de control de versiones GIT. También se describirá la forma de cómo se usó “Vagrant” que es una herramienta que nos permite cargar todo el software necesario y distribuirlo a un equipo de trabajo con diferentes sistemas operativos y tipos de hardware, lo cual nos ha resultado de las más adecuadas para nuestra investigación.

CAPÍTULO I

REVISIÓN DE LITERATURA

1.1 Marco teórico

Para nuestro marco teórico consideramos importante los siguientes conceptos, que se detallan a continuación.

1.1.1 Fondo Documental

Según la Directiva Nro. 10-2019-AGN/DDPA del AGN indica que el fondo documental “Corresponde a todos los documentos producidos (emitidos y recibidos) en la Entidad Pública, por todas las unidades de organización (productores) en el ejercicio de sus funciones o en el desarrollo de sus actividades, estructurados de acuerdo a criterios orgánicos, funcionales o mixtos” (Archivo General de la Nación, 2019)

1.1.2 Notario

El DS 10-2010-JUS lo define de la siguiente forma:

Artículo 4.- De la definición. - El notario es el profesional del derecho encargado, por delegación del Estado, de una función pública consistente en recibir y dar forma a la voluntad de las partes, redacta los instrumentos adecuados a ese fin, les confiere autenticidad, conserva los originales y expide traslados que dan fe de su contenido. Su función también comprende la comprobación de hechos y la tramitación de asuntos no contenciosos previstos en las leyes de la materia.(Normas Legales DS 10-2010-JUS, 2010)

El Art. 2 de la Ley del Notariado D.L. N° 1049[7] en su primera disposición final establece:

El Notario es el profesional del derecho que está autorizado para dar fe de los actos y contratos que ante él se celebran, para ello formaliza la voluntad de los otorgantes, redactando los instrumentos a los que confiere autenticidad, conserva los originales y expide los traslados correspondientes. Su función también comprende la comprobación de hechos y la tramitación de asuntos no contenciosos previstos en la ley de la materia.(Decreto Ley N° 1049, 2006)

1.1.3 Instrumentos Públicos Notariales Protocolares

Son aquellas escrituras públicas y actas que el notario incorpora al protocolo notarial, que debe conservar y expedir los traslados instrumentales que la ley determina (testimonio, boleta y parte notarial).(Perú - Ministerio Publico, 2014)

Es la colección de REGISTROS sobre la misma materia en los que el notario extiende los INSTRUMENTOS PÚBLICOS PROTOCOLARES con arreglo a ley. Base legal; Art. 36 D. Legislativo N° 1049.

1.1.4 Registros notariales

Forman el protocolo notarial los siguientes registros:

- De escrituras públicas (Arts. 50°-59° de la Ley).
- De testamentos (Arts. 67°-74° de la Ley y 686°-706° del C.C.)
- De protesto (75°-77° de la Ley y Ley N° 27287°, Arts. 73°, 78°, 85°, entre otros).
- De actas de transferencia de bienes muebles registrables (Arts 78°- 80° de la Ley, D.S. 036-2001-JUS).
- De actas y escrituras de procedimientos no contenciosos (Ley N° 26662, entre otras).

1.1.5 Registro de escrituras públicas

Base legal: Arts. 50° y siguientes Decretos Legislativos 1049.

ESCRITURA PUBLICA es todo documento matriz incorporado al protocolo notarial, autorizado por el notario, que contiene uno o más actos jurídicos.(Perú, 2014)

PARTES DE LA ESCRITURA PÚBLICA:

- 1) Introducción: Art. 54°.

2) Cuerpo: Art. 57°.

3) Conclusión: Art. 59°

1.1.6 Copias certificadas

Se presenta en aquellos casos en los que se solicita la transcripción por el notario de actas de entidades o personas jurídicas en general (Por ejemplo: transcribir el acta de la junta de accionistas de una Sociedad Anónima Cerrada, por la que se acuerda remover al Gerente General y nombrar a su reemplazante).

El notario da fe de la fiel transcripción, total o de la parte pertinente, con indicación del libro en el que consta dicha acta, los folios correspondientes y cualquier otra circunstancia que sea necesaria.

El notario no asume responsabilidad por el contenido del libro, del acta, ni firma, identidad, capacidad o representación de quienes aparecen suscribiéndolo. (Romero, 2014)

1.1.7 Bases de datos

El término ha sido utilizado para referirse a cualquier cosa, desde una colección de nombres y direcciones hasta un complejo sistema de recuperación y almacenamiento de datos que se basa en interfaces de usuarios y una red de computadoras y servidores. Hay tantas definiciones para la palabra base de datos como libros sobre éstas. Por otra parte, los distintos proveedores de DBMS han desarrollado diferentes arquitecturas, por lo que no todas las bases de datos están diseñadas de la misma manera. A pesar de la falta de una definición absoluta, la mayoría de las fuentes coinciden en que una base de datos, por lo menos, es una colección de datos organizada en un formato estructurado que es definido como metadatos que describe esa estructura. Puede pensar en los metadatos como información sobre los datos almacenados, que define cómo se almacenan éstos en una base de datos. (Oppel y Sheldon, 2010 p. 44)

El lenguaje estructurado de consultas (SQL, Structured Query Language) apoya la creación y mantenimiento de la base de datos relacional y la gestión de los datos dentro de la base de datos. Sin embargo, antes de entrar en discusión acerca de las bases de datos relacionales, quiero explicar el significado del término base de datos.

El término ha sido utilizado para referirse a cualquier cosa, desde una colección de nombres y direcciones hasta un complejo sistema de recuperación y almacenamiento de datos que se basa en interfaces de usuarios y una red de computadoras y servidores. Hay tantas definiciones para la palabra base de datos como libros sobre éstas. Por otra parte, los distintos proveedores de DBMS han desarrollado diferentes arquitecturas, por lo que no todas las bases de datos están diseñadas de la misma manera. A pesar de la falta de una definición absoluta, la mayoría de las fuentes coinciden en que una base de datos, por lo menos, es una colección de datos organizada en un formato estructurado que es definido como metadatos que describe esa estructura. Puede pensar en los metadatos como información sobre los datos almacenados, que define cómo se almacenan éstos en una base de datos. (Oppel y Sheldon, 2010 p. 24)

1.1.8 Normalización De Base de Datos

La parte central de los principios del modelo relacional es el concepto de normalización, una técnica para producir un conjunto de relaciones que poseen un conjunto de ciertas propiedades que minimizan los datos redundantes y preservan la integridad de los datos almacenados tal como se mantienen (añadidos, actualizados y eliminados).

La normalización de bases de datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional. (Oppel, 2010 p. 27)

Las bases de datos relacionales se normalizan para:

- Evitar la redundancia de los datos.
- Disminuir problemas de actualización de los datos en las Tablas.
- Proteger la integridad de los datos.

1.1.9 Programación Orientada a Objetos (OOP)

La OOP son un conjunto de técnicas que nos permiten incrementar enormemente nuestro proceso de producción de software; aumentando drásticamente nuestra productividad por un lado y permitiéndonos abordar proyectos de mucha mayor

envergadura por otro. Usando estas técnicas, nos aseguramos la re-usabilidad de nuestro código, es decir, los objetos que hoy escribimos, si están bien escritos, nos servirán para "siempre". Hasta aquí, no hay ninguna diferencia con las funciones, una vez escritas, estas nos sirven siempre. Pero es que, y esto sí que es innovador, con OOP podemos re-usar ciertos comportamientos de un objeto, ocultando aquellos otros que no nos sirven, o redefinirlos para que los objetos se comporten de acuerdo a las nuevas necesidades.(Villar, 2009 p. 13)

1.1.10 Marco de trabajo MVC

MVC, son las siglas de modelo-vista-controlador (o en inglés, model-view-controller), que es uno de los tantos patrones de arquitectura de software. Antes de abordar de lleno este patrón, vamos a intentar hacer una introducción a la arquitectura de software, desmembrándola de lo general hacia lo particular, para al fin llegar al detalle, procurando entender exactamente de qué se trata, en el contexto adecuado. Probablemente, este capítulo sea el más complejo (y mucho más extenso en relación al Capítulo I). Pero si quieres poder aplicar de forma [a]decuada el patrón MVC, debes hacer el esfuerzo de seguirlo con especial interés y actitud “entusiasta”. Bahit (2011 p. 33)

1.1.11 Arquitectura de Software

La Arquitectura de Software es la forma en la que se organizan los componentes de un sistema, interactúan y se relacionan entre sí y con el contexto, aplicando normas y principios de diseño y calidad, que fortalezcan y fomenten la usabilidad a la vez que dejan preparado el sistema, para su propia evolución. (Bahit, 2011 p. 33)

El diseño del software se enfoca en cuatro atributos distintos del programa; la estructura de los datos, la arquitectura del software, el detalle procedimental y la caracterización de la interfaz. El proceso de diseño traduce los requisitos en una representación del software con la calidad requerida antes de que comience la codificación. (Carlos Ble y Beas, 2010 p. 36)

1.1.12 Metodología del Desarrollo de Software

Según (García-Holgado y García-Peñalvo, 2018) indica que desde una perspectiva de Ingeniería de Software, una metodología Describe cómo se organiza un proyecto, establece el orden en el que la mayoría de las actividades tienen que realizarse y los enlaces entre ellas, indica cómo tienen que realizarse algunas tareas proporcionando las herramientas concretas e intelectuales. Con una metodología se intentan cubrir las siguientes necesidades [Piattini et al., 2004]: Mejores aplicaciones, Mejor proceso de desarrollo, Establecer un proceso estándar en una organización.

1.1.13 Metodología Ágil SCRUM.

SCRUM es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En SCRUM se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, SCRUM está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.

SCRUM también se utiliza para resolver situaciones en que no se está entregando al cliente lo que necesita, cuando las entregas se alargan demasiado, los costes se disparan o la calidad no es aceptable, cuando se necesita capacidad de reacción ante la competencia, cuando la moral de los equipos es baja y la rotación alta, cuando es necesario identificar y solucionar ineficiencias sistemáticamente o cuando se quiere trabajar utilizando un proceso especializado en el desarrollo de producto. (Proyectos Agiles.org, n.d.)

1.2 ANTECEDENTES

Laiza Torres (2015) realizó una tesis de grado titulada: Uso y eficacia del “Linux” y “Alfresco” para implementar un software de documentos digitalizados en la red local de la empresa Supplies Corporation del Perú S.A., consideró como objetivo mejorar la Gestión de Documentos Digitalizados en la Red Local de la empresa Supplies Corporation del Perú S.A. Este mismo trabajo concluye que con relación a la primera hipótesis específica: “El uso del Linux y Alfresco disminuyen considerablemente los tiempos de búsqueda de documentos digitalizados en la red local de la empresa Supplies Corporation del Perú S.A.”, se logró demostrar que el uso de Linux y Alfresco disminuyen considerablemente los tiempos de búsqueda de documentos, de 1,392 a 84 horas por año. (...) se logró demostrar que el uso de Linux y Alfresco reducen significativamente el uso de personas para la búsqueda de documentos, de un equivalente de 2,520 personas al año, a 0 (cero) personas (p. 46)

Muñoz Soro y Noguerras-Iso (2014) en su artículo: La digitalización de documentos en la Administración de Justicia concluye que cuando se compara la evolución del comercio y de la administración electrónica se observa que mientras que en el primero la necesidad de realizar operaciones comerciales de forma rápida y efectiva ha impulsado la toma de decisiones y acuerdos tanto a nivel tecnológico como organizativo, en el segundo se adoptaron con frecuencia requerimientos innecesariamente estrictos en cuestiones formales y de seguridad, lo que ha supuesto una barrera de tipo práctico muy importante. Este podría ser, de nuevo, el caso de la exigencia de un expediente integralmente electrónico, la cual no existe en las entidades privadas que, sin embargo, avanzan cada vez más en la “oficina sin papeles”.

En el artículo, se ha propuesto una medida para la progresiva adopción del expediente electrónico en la Administración de Justicia en base a la digitalización y conservación en formato electrónico de los documentos aportados por las partes. Desde el punto de vista práctico, tiene la ventaja de que los medios adicionales necesarios son los mínimos posibles. Pero, un punto clave es garantizar la seguridad en la conservación de los documentos electrónicos, y esta difícilmente puede obtenerse sin recurrir a repositorios especializados que sean accesibles a las oficinas judiciales de forma remota. Dado que su finalidad será únicamente el archivo y no la comunicación, deberían diseñarse de forma que solo el órgano judicial pudiera acceder a las copias. De esta forma, con pocos medios podría realizarse una experiencia piloto que permitiría avanzar en la modernización de la

oficina judicial, familiarizando a sus miembros con el manejo de las copias electrónicas auténticas. (p. 53).

Godoy (2017) concluye que se diseñó y construyó un Modelo Dinámico de Simulación para Proyectos de Software que se gestiona siguiendo la metodología SCRUM. Este modelo puede ser de utilidad para los SCRUM Master y el TEAM puedan analizar el efecto del uso conjunto de la metodología SCRUM, Bloques de Tiempo, Artefactos y Reglas en la gestión de los mismos en diferentes escenarios. La flexibilidad de modelo permite modificar los valores de los parámetros tanto al inicio de la simulación como al momento de la ejecución de la misma. Dentro de los parámetros que se pueden establecer previos al inicio de cada simulación se encuentran: la duración y la velocidad de cada Sprint, la velocidad estimada de desarrollo de las tareas, Factores de Cansancio, de Presión en el plazo, Cantidad de integrantes del TEAM según su experiencia en la metodología y las tareas extras que se prevén puedan surgir. A través de la modificación de valores de los parámetros el usuario puede establecer o modificar la cantidad de integrantes del TEAM que abandonan el proyecto, clasificar al TEAM mediante la asociación de estos a en base a su experiencia en SCRUM como Juniors o Expertos, cambiar la cantidad de horas estimadas de duración del proyecto, generar horas extras e inasistencia de los integrantes de manera determinística o pseudoaleatoria, entre otros. En cuanto a la metodología de dinámica de sistemas utilizada para construir el modelo, se ha validado su aplicación en este tipo de simulaciones. La validación permitió ajustar el modelo para que su comportamiento pueda reproducir con efectividad el comportamiento de los casos reales estudiados. Los casos de experimentación dieron la posibilidad de probar políticas alternativas para resolver situaciones que se puedan presentar en proyectos gestionados con SCRUM. Finalmente, luego de validar y experimentar con el modelo, se ha llegado a la conclusión de que el mismo puede ser utilizado como herramienta para evaluar el impacto de políticas alternativas de gestión y la detección de cuellos de botella en el desarrollo de proyectos gestionados con la metodología SCRUM. (p.162).

García-holgado (2014) explica que Los ecosistemas tecnológicos suponen un paso adelante en el desarrollo de aplicaciones software adaptadas a las necesidades de las empresas e instituciones, tanto para la gestión del conocimiento y del aprendizaje como para cualquier otra tarea que se desarrolle dentro de las mismas. La definición y desarrollo de este tipo de soluciones posee una mayor complejidad frente a los sistemas de

información que se venían utilizando hasta hace algunos años. Esto se debe a que el ecosistema tecnológico se compone de un conjunto de organismos, componentes software, que deben permitir los flujos de información necesarios para gestionar adecuadamente el conocimiento de una entidad. Un flujo de información entre dos aplicaciones o componentes software implica que exista una integración entre estos componentes. Unido a esto, una de las principales características del ecosistema tecnológico es la capacidad de evolución y de adaptación a las necesidades y los cambios del contexto en el que se enmarca el ecosistema. Por tanto, la integración de componentes no se limita únicamente a establecer una conexión entre dos o más elementos, sino que, además, la integración debe soportar la evolución de los componentes por separado y del ecosistema en su conjunto.

Cada ecosistema tecnológico es único, es muy difícil que dos instituciones o empresas compartan exactamente los mismos problemas y objetivos respecto a la gestión del conocimiento. En el caso de que esta situación se diera inicialmente, la componente evolutiva de los ecosistemas haría que esa similitud durara un corto periodo de tiempo ya que cada ecosistema evolucionaría de manera diferente, igual que ocurre en la naturaleza. La evolución de un ecosistema tecnológico es más sencilla de controlar que en un ecosistema natural pero la cantidad de variables que se manejan es grande por lo que la clave para que el sistema evolucione lo mejor posible es definirlo sobre una sólida base teórica probada en un contexto real. (p. 55)

Egas y Jativa (2014) concluye que La evolución de las Metodologías de Desarrollo de la Ingeniería de Software en el Proceso de Ingeniería de Sistemas Software, permite dar a conocer la cronología desde el nacimiento de las primeras técnicas de desarrollo software como “Code and Fix” (Codificar y corregir) hasta el establecimiento de las Metodologías Tradicionales y Ágiles como instrumentos ordenados y sistemáticos para el desarrollo software, los cuales en la actualidad tienen una gran acogida en las pequeñas, medianas y grandes empresas a nivel mundial. Tales Metodologías nos enseñan que no existe una metodología universal para hacer frente con éxito a cualquier proyecto de desarrollo de software; esto irá de la mano de acuerdo al contexto del proyecto y las necesidades particulares de cada empresa o institución. (p. 152).

Moreiro, Sánchez-Cuadrado, Palacios, y Barra (2011) expresa que Las instituciones suelen disponer de una intranet para el intercambio y la gestión de información administrativa, pero el tiempo que se requiere para buscar y obtener la que parezca más

adecuada a veces es importante. Por ello pensamos que es útil la evaluación comparativa que presentamos en este artículo. Existen diferentes aplicaciones en el mercado, algunas genéricas y otras diseñadas para cada institución con archivos administrativos concretos o adaptados a la legislación de cada país, lo cual acarrea un coste económico. Katherine Wisser (2005) realizó un estudio sobre las aplicaciones para codificar y publicar documentos de acuerdo con EAD (encoded archival description). Los resultados concluyeron que según el tipo de organización existían dos clases: las grandes instituciones solían disponer de aplicaciones desarrolladas para sus propias necesidades, mientras que las pequeñas solían utilizar el software EAD Cookbook (Fox, 2002).

Por otra parte, pequeñas y grandes organizaciones consideran usar aplicaciones de software libre como una supuesta alternativa de menor coste. Existe por tanto la necesidad de evaluar su capacidad de configuración y adaptación a nuevos contextos, su coste a largo plazo y la existencia de comunidades de programadores y usuarios que ayuden en su mantenimiento y actualización. (p.2)

Asato España, Ramirez Gonzáles, Fierro Mendoza, Gutiérrez Vera, y Galván Morales (2017) explica que, como cualquier herramienta, SCRUM y Kanban no son ni perfectas ni completas. No indican todo lo que se debe hacer, sólo proporcionan ciertas restricciones y formas de trabajar (Zambrana, 2003). Es importante no perder de vista que sólo marcan directrices, pero no indican cómo hacer las cosas, por lo que es un error considerarlas literalmente como metodologías. De acuerdo a la experiencia, la combinación de ambas herramientas (y quizá otras más) potencia significativamente la productividad, calidad y control del trabajo en proyectos complejos aún en el ámbito escolar. Debe tenerse presente que con Scrum se guía la forma de trabajar y con Kanban se controla el flujo del proceso. (...) Como toda herramienta, las alternativas mostradas tienen ámbitos de aplicación, alcances y limitaciones, por ello resulta conveniente tener en claro sus características para buscar alternativas complementarias según sea el objetivo perseguido. En el campo educativo es importante aplicarlas cuando lo relevante sea fortalecer la manera de trabajar más que lograr un producto terminado (el cual con un poco más de tiempo podría obtenerse), por lo que es recomendable aplicarlo con mayor intensidad en asignaturas de gestión de proyectos más que de programación básica ya que en estas últimas se busca un dominio técnico con enfoque de carácter individual. Adicionalmente resulta más adecuado proponer proyectos complejos (aunque no alcancen a ser completamente terminados durante el ciclo escolar) pero que escapen a la

capacidad de un desarrollador individual o un equipo pequeño de trabajo, ya que es en ese tipo de experiencias donde resulta significativa la aplicación de estas interesantes maneras de trabajar colaborativamente. (p. 25)

Hilario (2018) concluye que Según los resultados que se obtuvieron en esta investigación, se concluye que existe un alto nivel de insatisfacción por parte de los encuestados con respecto al sistema actual y a su vez un alto nivel de aceptación de la necesidad de realizar una propuesta de mejora del actual sistema, esto a través de un sistema web que trabaje de forma rápida, eficiente y optimice la mayoría de los procesos del área de Biblioteca. La interpretación realizada coincide con la hipótesis general propuesta para la investigación donde se mencionó que el desarrollo de un sistema web, mejorará los procesos para el control de biblioteca en la Institución Educativa Santa Rosa de Santo Domingo de la ciudad de Huarmey, en el año 2017. A partir de ello concluyo indicando que la hipótesis general queda debidamente aceptada. (p. 136).

Arce (2016) cuyo objetivo principal es la de reducir el tiempo que se emplea en los trámites documentales dentro de la institución educativa básica María de Nazareth, explica que: “Actualmente la sociedad dispone de diversas herramientas tecnológicas que permite optimizar diferentes recursos, por lo tanto, las instituciones públicas y privadas constantemente están realizando cambios en la infraestructura tecnológica para así poder cubrir las necesidades de una sociedad globalizada donde la tecnología forma parte de la comunidad” y concluye en que “El uso de herramientas tecnológicas open source permitió reducir costos. La utilización de la base de datos permitió incrementar el nivel de seguridad y mejorar la conservación de los documentos. Con el Sistema se estima que el tiempo de la tramitación de los documentos se reduzca, ya que la información va a estar centralizada. La aplicación de esta propuesta permitirá reducir el uso del papel, mermando así el número de impresiones. El uso del sistema web incrementará el nivel de satisfacción en los usuarios gracias a los resultados obtenidos.” (p. 102)

Núñez Paula y Núñez Govín (2005) afirma que La infraestructura tecnológica debe facilitar los flujos de información y la comunicación interactiva entre las personas y grupos que integran la organización en todo momento. El problema consiste en saber cómo emplear las denominadas TICs para aprender más y mejor y cómo utilizarlas para realizar una gestión eficaz del conocimiento. Las tecnologías de la información dinamizan la transmisión de la información, pero la gestión eficaz del conocimiento depende de la cognición humana y de la comunicación. Para lograrla, se necesita conocer y desarrollar



una cultura de la información, así como ajustar los procesos de comunicación e información a las características específicas de las personas y grupos; a ello, se le conoce genéricamente como personalización de la información.

Si se analiza el universo de productos informáticos que permiten el desarrollo de los distintos subprocesos de GC, puede observarse que, muy dispersos en la red, existen miles de ellos, cientos de cada tipo, y que su número crece diariamente con el empuje de las compañías que los crean, con propósitos que van desde los más generales hasta los más específicos, para facilitar la gestión de los flujos de información, de trabajo y la colaboración en el interior de las organizaciones. Lograr estos resultados, depende, en gran medida, del dominio de la herramienta y de conocer exactamente a dónde se quiere llegar y cómo lograrlo. La primera cuestión consiste en determinar, entre todas las accesibles, las herramientas que conformarán la infraestructura tecnológica de la organización; para ello, es necesario estudiar a fondo las características de la organización, su medio, los presupuestos disponibles y elegir cuál utilizar. Esta tarea requiere de un estudio profundo y detallado, que se dificulta frecuentemente por la diversidad de clasificaciones que existen para agrupar estas herramientas. Los costos del software son los más altos en la implantación de un sistema; se dice que es más costoso adquirir y adaptar el software que desarrollarlo (p.36)

CAPÍTULO II

PLANTEAMIENTO DEL PROBLEMA

2.1 Identificación del problema

Uno de los principales problemas que atraviesa el ARP (Archivo Regional de Puno) es que el proceso de búsqueda de documentos se realizaba en una búsqueda manual de los libros y revisar esta documentación de los notarios es preocupantes por las siguientes razones: PRIMERO, estos “índices notariales” son un listado que el mismo notario prepara como parte de su trabajo notarial y en algunos casos estos índices datan de 1900 u otros del siglo 1800, y como se entenderá están en papel, en un mal estado de conservación y la manipulación constante, está deteriorando cada vez más; SEGUNDO, la cantidad de tiempo que se debe invertir en una petición de búsqueda de una persona, sin embargo, cada día el ARP recibe en promedio 30 solicitudes de búsqueda notarial diarias, y otras solicitudes de búsqueda de partidas de nacimiento, matrimonio y defunción, expedientes judiciales u otras, haciendo en algunos casos muy complicado atender a todos en el día, la cual es la política del ARP; TERCERO, la falta de espacio ha sido uno de los problemas que el ARP ha atravesado y estando en locales alquilados por muchos años, aunque ahora cuenta con una nueva infraestructura, que ha minimizado ciertos problemas.

Otro de los problemas detectados es la pobre capacitación en el uso de equipos informáticos y la resistencia a usar una nueva tecnología, has sido un impedimento a nuestra propuesta, no contar con la cantidad de computadoras necesarias, ni tampoco de un servidor que pueda almacenar la información, han sido problemas que han ido creciendo con el tiempo, tal como lo detalla los gráficos estadísticos mostrados más adelante, haciendo uso de la regresión lineal simple (análisis estadístico) para la

estimación de crecimiento de la cantidad de solicitudes diarias y atendidas, y este número tanto en personas como en solicitudes ha ido creciendo de manera progresiva, lo que en pocos años, se convertiría en algo inmanejable si se seguía trabajando de esta manera, por lo que realizó un trabajo de concientización, la adquisición que un servidor, equipos de redes y computadoras, y las capacitaciones al personal que se hicieron en transcurso del año, a todo el personas por varias etapas.

2.2 Enunciados del problema

En base a los argumentos vertidos anteriormente, es que nos formulamos la siguiente interrogante.

¿De qué manera el desarrollo de un Sistema de Registro Notariales usando Vagrant, acelerará la búsqueda y recuperación de información notarial en el ARP?

2.3 Justificación

Viviendo la era digital, es impensable que instituciones del estado estén en esta situación, sin embargo la realidad así lo confirma, y la mayoría de las instituciones públicas del estado no invierten en la capacitación a sus trabajadores y en el fortalecimiento del aparato estatal, lo que dificulta que esta pueda alcanzar estadales altos de competitividad con las empresas privadas.

Sin embargo el ARP es una institución “única” en su género y no tiene competencia con la empresa privada, y esto no solo pasa en nuestro país, las instituciones que velan por el patrimonio de la nación son únicas por país, existiendo organizaciones privadas, pero que no por su condición jurídica, no pueden otorgar copias de documentos legalizados que son los que tiene valor, por esa razón es que en muchos países ya se ha comenzado hace muchos años, la modernización del aparato archivístico, siendo una de los primos referentes el Archivo de España y el Archivo de Indias, seguido de los de Francia y otros.

En nuestro país, solo se siguen algunas pautas especialmente de conservación mas no de modernización, aunque el gobierno ha promulgado la (LEY N° 27658, 2002) la que indica en el capítulo 1, art. 1 “Declárase al Estado peruano en proceso de modernización en sus diferentes instancias, dependencias, entidades, organizaciones y procedimientos, con la finalidad de mejorar la gestión pública y construir un Estado democrático, descentralizado y al servicio del ciudadano.”. Posteriormente en el año 2013 aprueban la Política

Nacional de Modernización de la Gestión Pública por la Presidencia del Consejo de Ministros.

En base a estas leyes y otras más que el Gobierno del Perú ha dado, es que proponemos que se modernice la gestión pública en el ARP, y se propone el análisis y desarrollo de una aplicación que permita el rápido ingreso de la información notarial y que esta permita la rápida ubicación de los libros con las escrituras públicas y mejorar la atención al usuario.

La cantidad de usuarios atendidos es como se muestra en la siguiente Tabla

Tabla 1

Servicios archivísticos 2001 al 2013

Año	Usuarios atendidos en A. I.	Investigadores y usuarios particulares atendidos en A. H.	Supervisión asesoramiento y capacitación	Asesoría Jurídica	Total servicios prestados
2001	5,055	64			5,119
2002	4,427	249			4,676
2003	3,999	184			4,183
2004	3,798	200			3,998
2005	3,333	321			3,654
2006	3,524	208			3,732
2007	4,384	255	124		4,763
2008	5,805	209	85		6,099
2009	5,437	132	25		5,594
2010	4,986	336	22	710	6,054
2011	5,296	187	78	697	6,258
2012	5,289	172	29	1,099	6,589
2013	1,018	130	-	-	1,031

Fuente: Estadísticas del Archivo Regional de Puno

2.4 Objetivos

2.4.1 Objetivo general

Desarrollar un Sistema de Registros Notariales usando un sistema de virtualización Vagrant para acelerar la búsqueda y recuperación de información.

2.4.2 Objetivos específicos

- Identificar las variables que intervendrán en la elaboración del software y la base de datos, así como los requisitos del sistema
- Aplicar técnicas y metodologías ágiles, para el desarrollo del software, haciendo uso de la virtualización y modelado de las Bases de Datos
- Evaluar el software en diferentes condiciones e identificar y corregir fallos

2.5 Hipótesis

2.5.1 Hipótesis general

El desarrollo de un Sistema de Registro Notariales en el Archivo Regional usando un sistema de virtualización Vagrant, acelerará la búsqueda y recuperación de información.



CAPÍTULO III

MATERIALES Y MÉTODOS

3.1 Lugar de estudio

El lugar de estudio se ubica en el departamento de Puno, Provincia de Puno, distrito de Puno, en la ciudad de Puno, Av. El ejercito 645 Urb. Chanu Chanu en un terreno de 1100 metros cuadrados, local del Archivo Regional de Puno, oficina de informática.

El Archivo Regional de Puno en cumplimiento de sus funciones custodia documentación que constituye Patrimonio Documental de la Nación en la Región en dos locales, la Oficina principal ubicada en el Jr. Arequipa N° 1145 [antiguo local] y un repositorio ubicado en los talleres del Gobierno Regional Puno (Jr. Carabaya). Este repositorio alberga gran cantidad de documentación, en condiciones de hacinamiento, poniendo en riesgo de deterioro el Patrimonio Documental.(Archivo Regional de Puno, 2015, p.5)

3.2 Población

La población de personas que son permanentes entre personal contratado y personal nombrado es alrededor de 37 personas, aunque es número es variable, debido a que por necesidad por algunos meses es requerido la contratación de más personal lo que no afecta al proyecto.

Según el (Archivo Regional de Puno, 2015) en su POI 2015 indica que Actualmente la institución cuenta con 20 trabajadores nombrados, 01 cargo de confianza, 12 contratados por Contratos Administrativos de Servicios (CAS) y 4 por Servicios Diversos. El personal está distribuidos en las siguientes unidades orgánicas: Dirección 03 (8%), Asesoría Jurídica

(5%), Órgano de Administración de Archivos (5%), Informática (19%), Oficina Técnica Administrativa (19%), Dirección de Archivo Intermedio (33%) y Dirección de Archivo Histórico (11%). De los 37 trabajadores 19 son profesionales y 18 son técnicos. (p.7).

Tabla 2

Distribución del personal por unidades orgánicas

UNIDAD ORGÁNICA	Nº Trabajadores	%
DIRECCIÓN: Profesionales	31	8%
Técnicos	2	
	2	
ASESORÍA JURÍDICA: Profesionales	2	5%
Técnicos	2	
ÓRGANO DE ADMINISTRACIÓN DE ARCHIVOS: Profesionales	0	
Técnicos	2	5%
	2	
	0	
INFORMÁTICA: Profesionales	7	19%
Técnicos	2	
	5	
OFICINA TÉCNICA ADMINISTRATIVA: Profesionales	7	19%
Técnicos	34	
DIRECCIÓN DE ARCHIVO INTERMEDIO: Profesionales	12	33%
Técnicos	6	
	6	
DIRECCIÓN DE ARCHIVO HISTÓRICO: Profesionales	4	11%
Técnicos	3	
	1	
TOTAL	37	100%

Fuente: Elaborado por el área de Planificación y Presupuesto – ARP. (POI 2015)

Debemos considerar también el software está orientado hacia los documentos y también se muestra la población de libros (protocolos notariales) y otros documentos que custodia el ARP los cuales se presentan a continuación.

Tabla 3

Numero Notarios por Provincias

PROVINCIAS	N° de Notarios	Porcentaje (%)
Puno	61	46%
Azángaro	16	12%
Melgar	12	9%
San Román	11	8%
Yunguyo	2	2%
Huancané	6	5%
Lampa	5	4%
Carabaya	7	5%
Chucuito	11	8%
Sandia	2	2%
TOTAL	133	100%

Fuente: Archivo Regional de Puno

Tabla 4

Fondos documentales y colecciones - Archivo Intermedio

FONDO DOCUMENTAL	METROS LINEALES	FECHAS EXTREMAS
Fondo Notarial	535.1	1963 a 2007
Fondo Corte Superior de Justicia	522.78	1692 a 1993
Fondo Registro Civil Fondo Ministerio Público Fondo INEI	143	1890 a 2004
Fondo Expedientes Archivados	34.2	1949 a 1995
	32.5	1985 a 1995
	0.5	1995 a 2002
SUB TOTAL	1268.08	

Fuente: Archivo Regional de Puno

Tabla 5

Fondos documentales y colecciones - dirección Archivo Histórico

N°	FONDO DOCUMENTAL	UNIDAD MEDIDA	CANT. TOTAL	CANTIDAD ORGANIZ.	POR ORGANIZ.	FECHAS EXTREMAS
1	Notarial	M.l.	18,9	9		1663 - 1919
2	Corregimiento	M.l.	0,7			1703 - 1783
3	Intendencia	M.l.	6,5	1,8	4,7	1784 - 1824
4	Juzgado de Primera Instancia	M.l.	1,1			1824 - 1850
5	Corte Superior de Justicia:					
	Expediente penales	M.l.	486,8	139,76	347,04	1850 - 1994
	Copiadores, Investiga. Sentencias, otros	M.l.	85,16			1850 - 1996
	Juzgado de Paz No Letrado	M.l.	19,39	19,39	0	1825 - 1982
6	Prefectura de Puno	M.l.	81,19	81,19		1825 - 1995
7	Sub-Prefectura de Lampa	M.l.	8,44		8,44	1825 - 1995
8	Municipalidad Provincial de Puno	M.l.	16,72		16,72	1852 - 1996
9	Beneficencia Publica puno	M.l.	31		31	1841 - 1986
10	Tesorería Fiscal	M.l.	21		21	1826 - 1970
11	Contribuciones (Impuestos Sucesorios)	M.l.	15,93	15,93		1917 - 1981
12	SUNAT (Varios Documentos)	M.l.	15,43		15,43	1900 - 1974
13	Colecciones Familiares	M.l.	10,4		10,4	1830 - 1973
14	Periódicos e Impresos	M.l.	10,76	10,76		1838 - 1999
15	Planos					
16	Fotografías	unidad	1465		1465	
17	Audiovisuales	unidad				
	VHS					
	Casetes	unidad	22			1996- 1999
	CD	unidad	14			

Fuente: Archivo Regional de Puno.

3.3 Muestra

Para el cálculo de tamaño de muestra cuando el universo es finito, es decir contable y la variable de tipo categórica, primero debe conocer "N" ó sea el número total de casos esperados ó que ha habido en años anteriores (Por ejemplo, en el año 2009), para eso deben revisar los datos estadísticos del Departamento de Pediatría.

Si la población es finita, es decir conocemos el total de la población y deseásemos saber cuántos del total tendremos que estudiar la fórmula sería:

$$n = \frac{N * Z_{\alpha}^2 * p * q}{d^2 * (N - 1) + Z_{\alpha}^2 * p * q}$$

Donde:

- $N =$ Total de la población
- $Z_{\alpha} = 1.96$ al cuadrado (si la seguridad es del 95%)
- $p =$ proporción esperada (en este caso $50\% = 0.5$)
- $q = 1 - p$ (en este caso $1 - 0.5 = 0.5$)
- $d =$ precisión (en su investigación use un 5%).

Para nuestro caso consideramos a las 37 personas como una población pequeña y finita, y no todos usan el software, por lo que no se hará ninguna estimación o cálculo de la muestra para los trabajadores, por lo que se usará a toda la población de ellas para nuestra validación.

Se utilizó el tamaño de la muestra a los libros (protocolos notariales) que tiene un aproximado de 2780 libros, para lo cual, aplicando el tamaño de la muestra, esta quedaría así.

$$n = \frac{2780 * 1.96^2 * 0.5 * 0.5}{0.05^2 * (2780 - 1) + 1.96 * 0.5 * 0.5}$$

$$n = \frac{2669.91}{7.4375} = 358.98$$

$$n = 359 \text{ Libros}$$

El tamaño de la muestra nos indica que debemos considerar una muestra de los libros de 359 para nuestra validación del software.

3.4 Metodología de investigación

La metodología de desarrollo de software eXtream Programing (XP) para la codificación (fases de diseño y codificación) del proyecto y la metodología SCRUM para la gestión del equipo humano, es la que se utilizó para llevar a cabo este proyecto. Para esto nos preguntamos qué es SRUM así que la web proyectos agiles no da la siguiente definición:

Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

En Scrum se realizan entregas parciales y regulares del producto final, priorizadas por el beneficio que aportan al receptor del proyecto. Por ello, Scrum está especialmente indicado para proyectos en entornos complejos, donde se necesita obtener resultados pronto, donde los requisitos son cambiantes o poco definidos, donde la innovación, la competitividad, la flexibilidad y la productividad son fundamentales.(Proyectos Agiles.org, n.d.)

En el libro de Roger S. Pressman (2005) en el capítulo 4, 4.1 sobre agilidad, este cita a Ivar Jacobson indicando lo siguiente:

Agilidad se ha convertido actualmente en la palabra de moda en cuanto se describe un moderno proceso de software. Cualquiera es ágil. Un equipo ágil es un equipo rápido que responde de manera apropiada a los cambios. Éstos son, en gran parte, la materia del desarrollo de software. Cambios en el software que se va a construir, cambios entre los miembros del equipo, cambios debidos a las nuevas tecnologías, Cambios de todo tipo que pueden incidir en el producto que se construye o en el proyecto que crea el producto. En cualquier actividad de software se debe incluir un soporte para los cambios, esto es algo que adoptamos porque es el alma y el corazón del software. Un equipo ágil reconoce que el software lo desarrollan individuos que trabajan en equipo y que las aptitudes de esta gente, y su capacidad para colaborar, son esenciales para el éxito del proyecto. (p.79)



Pressman (2005) indica en su libro que la Programación Extrema “abarca un conjunto de reglas y prácticas que ocurren en el contexto de cuatro actividades del marco de trabajo: planeación, diseño, codificación y pruebas”. (p.84)

En el desarrollo de software, el material principal son los análisis del negocio, como está formado, cuantos actores van a intervenir el uso del software, así como las iteraciones y secuencias de trabajo que se desarrollaran. Una vez analizado estos temas se utilizará la metodología SCRUM, conformar el equipos de desarrollo, y demás. Como parte del equipamiento, será necesario contar con computadoras con sistema operativo Linux de preferencia (cualquier distribución), debido a que estos sistemas de virtualización como Vagrant, cargan en su interior un sistema operativo Linux con todos los programas necesarios como apache o Nginx, PHP, soporte JSON, Bases de datos MySQL, PostgreSQL, Mondodb, etc.

El método a utilizar para la gestión del software es SCRUM, y esta inicia primeramente, eligiendo un responsable del producto, se conforma el equipo de desarrollo y se elige el Scrum master, que es el encargado de ser el nexo entre los desarrolladores y el cliente. Se elabora y prioriza una lista de objetivos o backlog, que más que una lista, es todo lo que debe hacerse para convertir la visión del sistema a desarrollar en una realidad, revisando constantemente los avances del proyecto usando un tablero Kanban (también es parte de la metodología ágil). Se hace una estimación afinada de la lista de objetivos pendientes en tamaños pequeño, mediano o grande, seguidamente de planifican los Sprints, en base a la velocidad del equipo para lograr completar los sprint.

Todo el trabajo debe ser visible con una pizarra de Scrum y 3 columnas: Pendiente, en proceso y Hecho, esta ayudarán a que se vea como y a qué velocidad está avanzado el proyecto. Parte del éxito de SCRUM en el desarrollo de aplicaciones es el Scrum diario o la reunión diaria de pie, donde se hacen 3 preguntas al equipo: ¿Qué hiciste ayer? ¿Qué vas hacer hoy? ¿Qué problemas tienes?, esto ayuda mucho a que el equipo este enfocado y podamos darnos cuenta de los problemas de manera rápida (ósea, ágil). Luego es necesario una Revisión o demostración del sprint, un análisis de retrospectiva del sprint, que consideramos que es las más importante de todas las etapas que desarrollamos, las lecciones aprendidas. Terminado este pasó, se empieza inmediatamente el siguiente ciclo de Sprints.

Las Técnicas a utilizar son: diagramas de UML, normalización de bases de datos, uso de la metodología SCRUM en todas sus fases, entrevistas con los clientes, reuniones de trabajo constante. Capacitaciones sobre el uso de sistemas de control de versiones, como centralizar el código. Adicional a esto, la estandarización de la codificación, para obtener mejores resultado y material organizado para presentar al final, un manual de uso del sistema, así como sus características principales.

Para el análisis estadístico, se hará uso de un Tabla de mando integral. “La métrica más importante en un proyecto ágil como Scrum es el valor que se está dando al cliente. Mediante esta métrica, el cliente puede conocer la velocidad con que retorna su inversión y saber cuándo ya no es necesario seguir con el proyecto, porque los beneficios pendientes de obtener ya no compensan sus costes” (Proyectos Agiles.org, 2018). Entre las métricas de productividad y efectividad de la entrega, se mide la Velocidad con que se completan objetivos/requisitos en cada iteración, Tiempo de entrega de un requisito tras su petición o Lead Time, Urgencias y prioridad/valor de los requisitos completados. Además existen las métricas de resultados del proyecto y entre lo que se mide es la Velocidad con que se aporta valor al negocio, Número de requisitos completados respecto al total de requisitos, Días de trabajo ideales pendientes, Desviación de resultados de proyecto respecto a planificación inicial, etc. Son varios como las métricas de situación financiera, Retorno de Inversión (ROI) pendiente, Presupuesto disponible y/o presupuesto gastado, Desviación financiera respecto a la planificación inicial.

Estadísticamente hay muchos datos que se pueden analizar, desde los ya descritos y también están los que se registrarán en la base de datos, como por ejemplo: número de pasajeros, promedio de venta de boletos mensuales, número de unidades, distancias entre rutas, fecha y hora de salida y llega de vehículos, nombre de los trabajadores, planillas, etc., en fin son muchos a analizar dependiendo del análisis que se realizará y se priorizarán los más urgentes, tal como los especifica la metodología a seguir.

3.5 Descripción detallada de métodos por objetivos específicos

Se muestra en este apartado los recursos y metodologías utilizadas para la elaboración del trabajo

Los objetivos específicos planteados son los siguientes: a) Identificar las variables que intervendrán en la elaboración del software y la base de datos, así como los requisitos del sistema, b) Aplicar técnicas y metodologías ágiles, para el desarrollo del software, haciendo uso de la virtualización y modelado de las Bases de Datos, c) Probar el software en diferentes condiciones e identificar y corregir fallos.

3.5.1 Recursos

Tabla 6

Recursos necesarios para el Inicio del Proyecto: Fondo Notarial

RECURSOS: HARDWARE

Núm.	Descripción	Cantidad
1	Servidor IBM x3400	1
2	Switch Cisco 2900	1
3	Switch Básicos	6
4	Computadoras Core i5	15

RECURSOS: EQUIPOS Y ACCESORIOS

Núm.	Descripción	Cantidad
5	Cableado estructurado	1
6	Cableado Eléctrico	1
7	Cuchillas Termodinámicos	1
8	Proyector Multimedia	1

RECURSOS: MUEBLES

Núm.	Descripción	Cantidad
9	Escritorios de madera	6
10	Sillas	6
11	Otros	1

Fuente: Área de Informática

Para la vista de presentación física del despliegue del sistema, será necesaria la participación de tres elementos clave en la red:

a) La PC del Usuario

Este equipo de cómputo PC o Laptop es el que se utiliza en el laboratorio de cómputo del ARP, y que accederá a la aplicación vía la red interna. La PC del usuario debe estar conectada a la red LAN del ARP.

b) El Servidor de Aplicaciones

Equipo informático en donde se encuentra alojado el sistema, en la cual se encuentran además los archivos usando el modelo de datos MVC, clases y otros, para la ejecución de las transacciones e interacción entre la base de datos y el usuario.

c) El Servidor de Base de Datos

Equipo informático en donde se encuentra almacenada la Base de Datos. Este servidor está configurado usando un sistema operativo Linux, corriendo sobre el sistema Vagrant

Capa de Acceso a Datos (Física)

Esta capa controla todo lo referente al acceso a datos: Base de Datos y archivos en general; aquí se programan las clases que se encargarán de los accesos y toma de datos de los archivos.

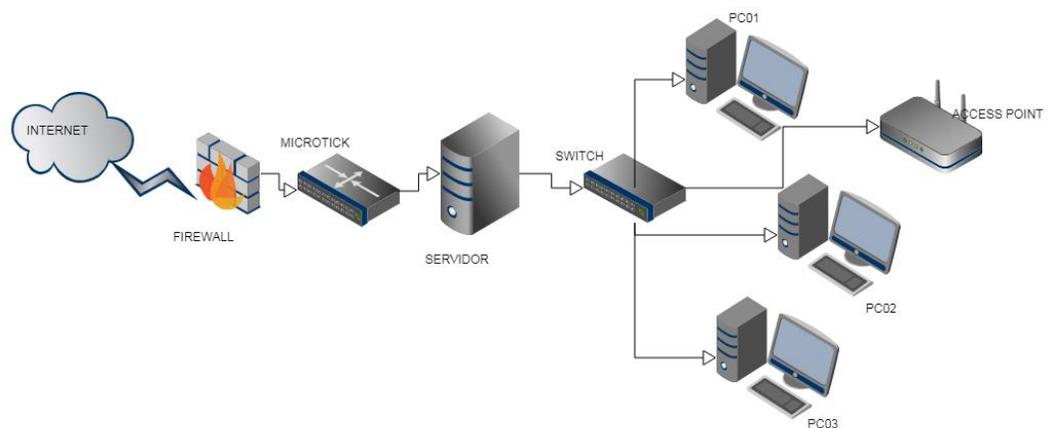


Figura 1. Modelo del Sistema de Red en el AR Puno

3.5.2 Interfaces del Sistema

A continuación, se presentarán las principales interfaces del Sistema Fondo Notarial que consta de 3 Módulos principales, que es: Modulo de Ingreso de Datos al sistema, Modulo Buscador de datos y Modulo de corrección de información (Control de Calidad).

La Figura siguiente muestra el diseño de la interfaz sencilla, solo en Figura sin ningún tipo de programación (Modelo gráfico)

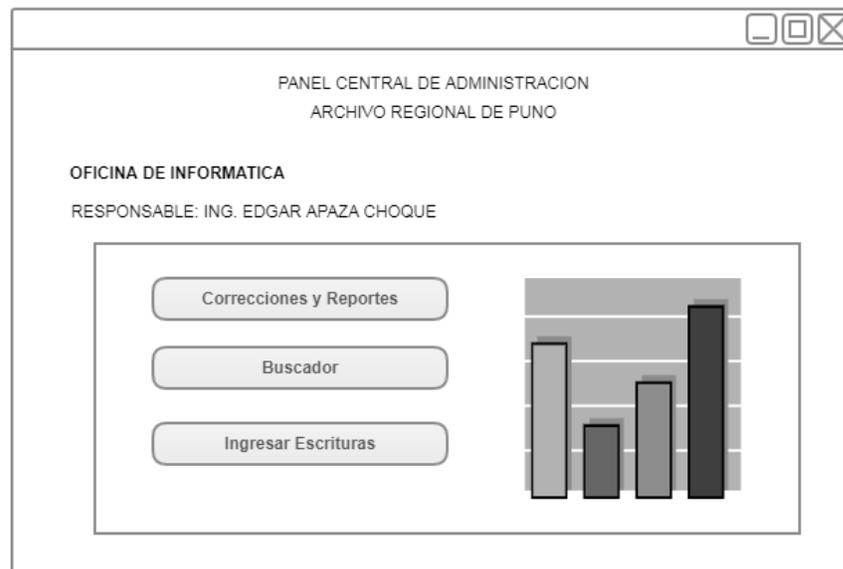


Figura 2. Modelo de la ventana de Ingreso – Panel de administración principal

Estas interfaces estarán divididas por los Módulos del Sistema que se han desarrollado. Entre estas se encuentran la vista de Corrección y reportes, el buscador, la ventana de ingreso de escrituras.

Los modelos y diseños iniciales se hicieron previamente en hojas de papel, revisados por el equipo de trabajo, dentro de los cuales los diseños se escribieron en HTML, CSS3 y JavaScript.

3.5.3 Implementación del Software

Instalación de Sistema Operativo Linux (Vagrant)

La parte inicial de la implementación de la solución ha consistido en la instalación del Sistema Operativo Linux en el equipo, seguidamente configurar el nombre del equipo y actualizar (con el parámetro `sudo apt-get update`). Para facilitar al administrador el manejo del Sistema Operativo se activará la interfaz gráfica instalando el gestor de ventanas GNOME (GNU Network Object Model Environment), con la línea de comando `sudo apt-get install gnome`.

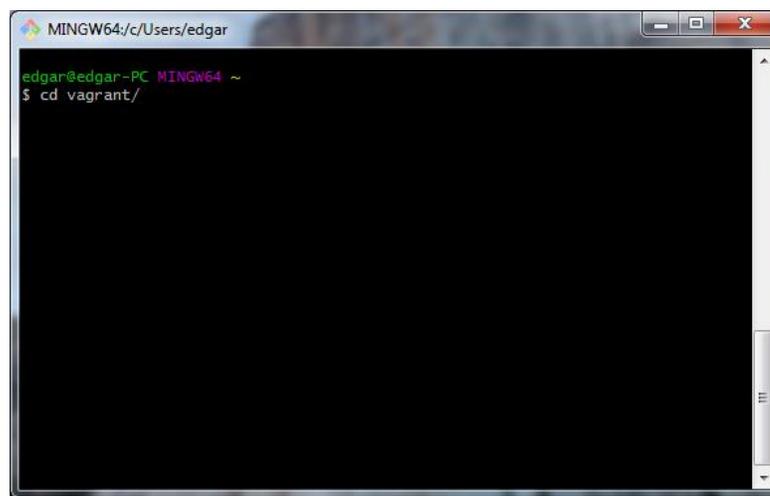


Figura 4. Ingreso a Vagrant en Windows 7, usando el software Git Bash

Una vez dentro de la carpeta de Vagrant, podemos ver las distintas máquinas de software, cada una con una configuración diferente. Para nuestro proyecto usaremos la carpeta “machine1”, que ya tiene instalado el Sistema Operativo Linux Ubuntu versión 16.04 LTS, servidor NGINX, apache2, PHP 7.0, MariaDB-server, entre otros.

```
MINGW64:/c/Users/edgar/vagrant
edgar@edgar-PC MINGW64 ~/vagrant
$ ls
machine1/  puppet.zip  server_apache/  servidor1/  software/
edgar@edgar-PC MINGW64 ~/vagrant
$ |
```

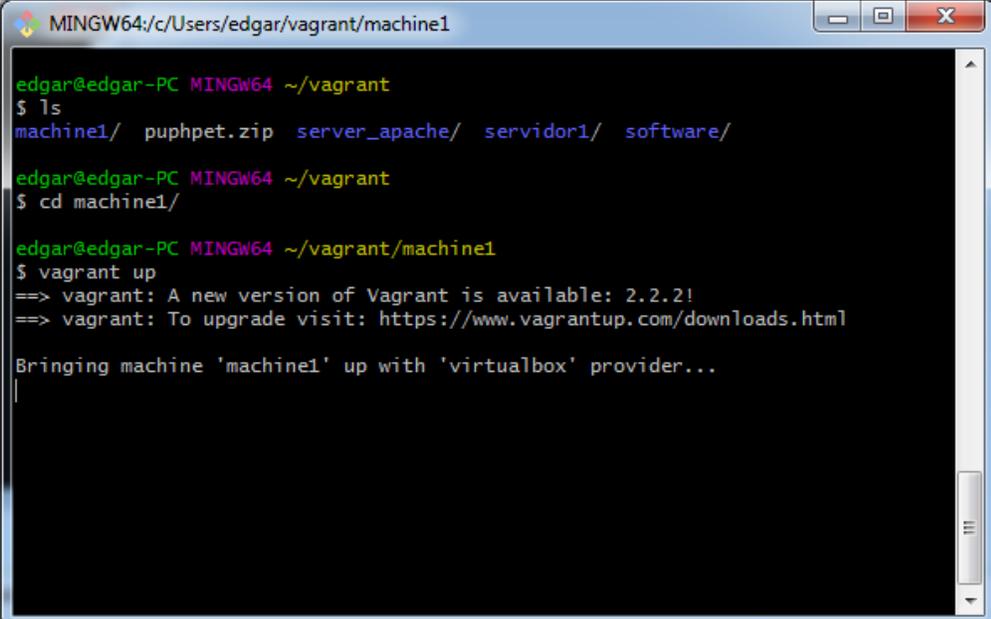
Figura 5. Cargando y arrancando Vagrant usando la máquina virtual “machine1”.

Debemos ingresar a la carpeta con el comando `cd machine1` y luego presionar la tecla enter.

```
MINGW64:/c/Users/edgar/vagrant/machine1
edgar@edgar-PC MINGW64 ~/vagrant
$ ls
machine1/  puppet.zip  server_apache/  servidor1/  software/
edgar@edgar-PC MINGW64 ~/vagrant
$ cd machine1/
edgar@edgar-PC MINGW64 ~/vagrant/machine1
$
```

Figura 6. Ingresando a la máquina virtual Machine1

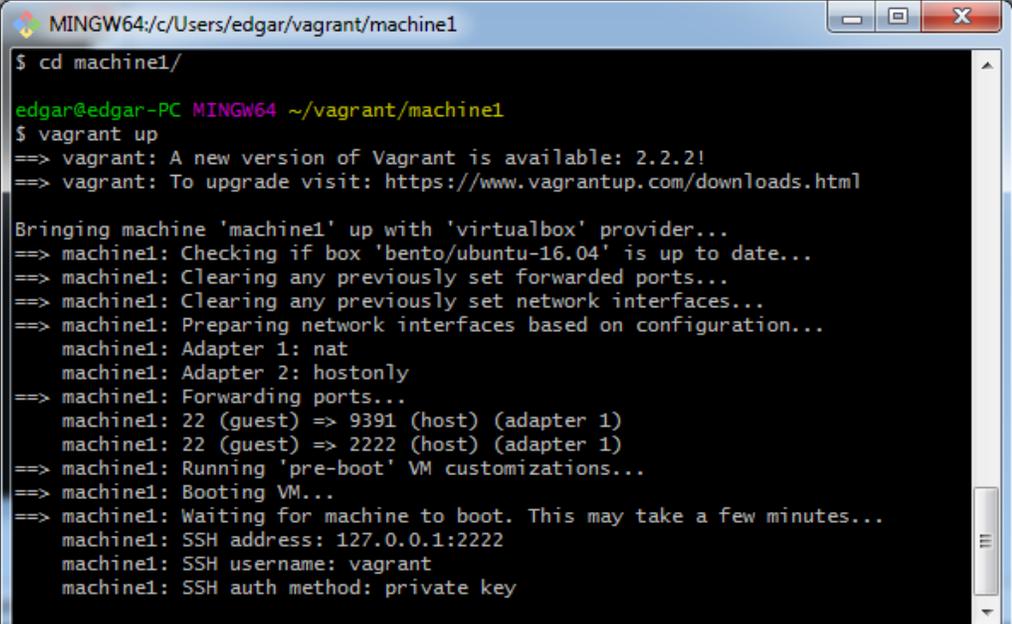
Como se puede ver, ya estamos dentro de la carpeta de `machine1`, y dentro de ella escribimos el comando: **Vagrant up** obteniendo la siguiente vista, que nos muestra como el servidor Vagrant está corriendo.



```
MINGW64:/c/Users/edgar/vagrant/machine1
edgar@edgar-PC MINGW64 ~/vagrant
$ ls
machine1/  puppet.zip  server_apache/  servidor1/  software/
edgar@edgar-PC MINGW64 ~/vagrant
$ cd machine1/
edgar@edgar-PC MINGW64 ~/vagrant/machine1
$ vagrant up
==> vagrant: A new version of Vagrant is available: 2.2.2!
==> vagrant: To upgrade visit: https://www.vagrantup.com/downloads.html
Bringing machine 'machine1' up with 'virtualbox' provider...
```

Figura 7. La Figura muestra como inicia el Vagrant

Comienza la configuración del SSH que lo utilizaremos para conectarnos a la máquina virtual.



```
MINGW64:/c/Users/edgar/vagrant/machine1
$ cd machine1/
edgar@edgar-PC MINGW64 ~/vagrant/machine1
$ vagrant up
==> vagrant: A new version of Vagrant is available: 2.2.2!
==> vagrant: To upgrade visit: https://www.vagrantup.com/downloads.html
Bringing machine 'machine1' up with 'virtualbox' provider...
==> machine1: Checking if box 'bento/ubuntu-16.04' is up to date...
==> machine1: Clearing any previously set forwarded ports...
==> machine1: Clearing any previously set network interfaces...
==> machine1: Preparing network interfaces based on configuration...
machine1: Adapter 1: nat
machine1: Adapter 2: hostonly
==> machine1: Forwarding ports...
machine1: 22 (guest) => 9391 (host) (adapter 1)
machine1: 22 (guest) => 2222 (host) (adapter 1)
==> machine1: Running 'pre-boot' VM customizations...
==> machine1: Booting VM...
==> machine1: Waiting for machine to boot. This may take a few minutes...
machine1: SSH address: 127.0.0.1:2222
machine1: SSH username: vagrant
machine1: SSH auth method: private key
```

Figura 8. Configurando los parámetros de inicio para la comunicación de datos

Una vez finalizada la carga en aproximadamente 3 minutos, es posible regresar a la terminal y escribir el comando para otras configuraciones. Actualmente el servidor ya se encuentra funcionando, ya que utiliza una máquina virtual con VirtualBox, la cual ya se encuentra disponible desde este preciso momento.

```
MINGW64:/c/Users/edgar/vagrant/machine1
- provision
==> machine1: flag to force provisioning. Provisioners marked to run always will
still run.
==> machine1: Running provisioner: shell...
machine1: Running: C:/Users/edgar/AppData/Local/Temp/vagrant-shell120181210-5
608-1gpbrer.sh
machine1: Running files in files/startup-once
machine1: Finished running files in files/startup-once
machine1: To run again, delete hashes you want rerun in /opt/puppet-state/s
tartup-once-ran or the whole file to rerun all
machine1: Running files in files/startup-always
machine1: Finished running files in files/startup-always
==> machine1: Running provisioner: shell...
machine1: Running: C:/Users/edgar/AppData/Local/Temp/vagrant-shell120181210-5
608-1pjzewb.sh
machine1: Running files in files/startup-once-unprivileged
machine1: Finished running files in files/startup-once-unprivileged
machine1: To run again, delete hashes you want rerun in /opt/puppet-state/s
tartup-once-unprivileged-ran or the whole file to rerun all
machine1: Running files in files/startup-always-unprivileged
machine1: Finished running files in files/startup-always-unprivileged

edgar@edgar-PC MINGW64 ~/vagrant/machine1
$
```

Figura 9. Figura que muestra e indica que la Máquina virtual está corriendo (Running)

Para comprobar su funcionamiento, debemos abrir cualquier navegador de nuestra preferencia, que para nuestro caso será Google Chrome, y escribimos <http://servidor.test> y nos mostrará el siguiente resultado.

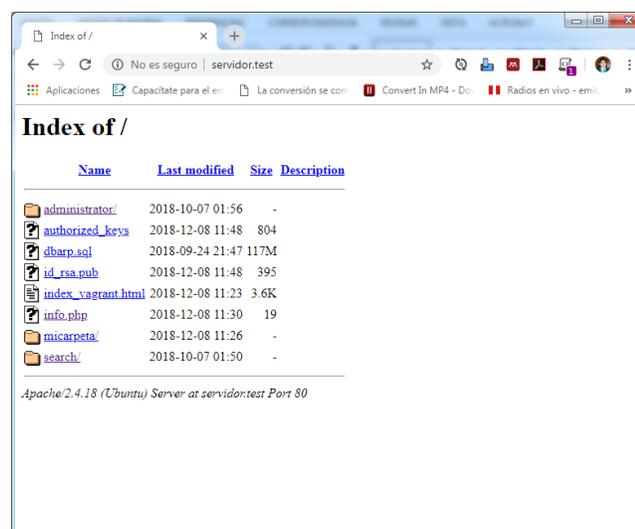


Figura 10. Figura que indica que el servidor está corriendo

Ahora si volvemos a escribir <http://buscador.test>, obtendremos el siguiente resultado.

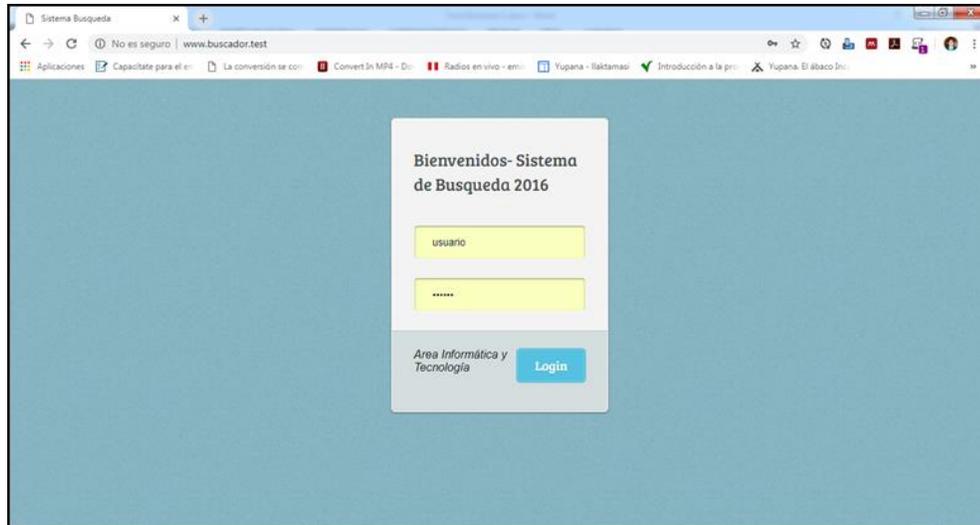


Figura 11. Pantalla de inicio del Buscador de Escrituras públicas.

Esta es la ventana inicial del sistema búsquedas que desarrollamos, siendo el usuario lo mismo, con la contraseña: “binter”.

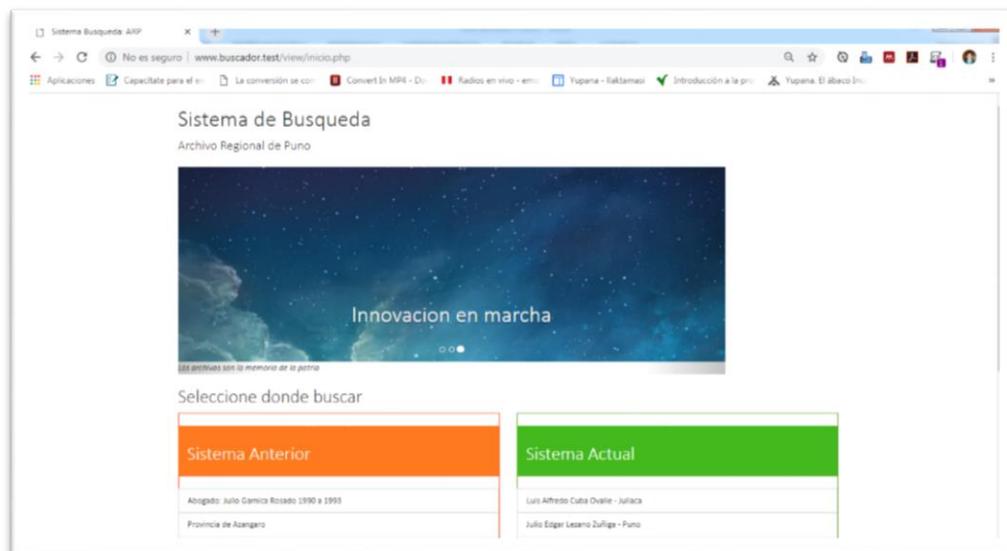


Figura 12. Figura que muestra como poder acceder al sistema de búsqueda. Con las credenciales listas, se pueden ver 2 botones que indican Sistema anterior, refiriéndose a una base de datos mas antigua que se ha modificado poco, y el sistema

Actual en verde, que contiene mejoras en el sistema y en las Tablas de la base de datos.

Ambos cuentan con acceso a la misma información, y la suma de ambos registros es aproximadamente unos 620 mil registros de escrituras públicas.

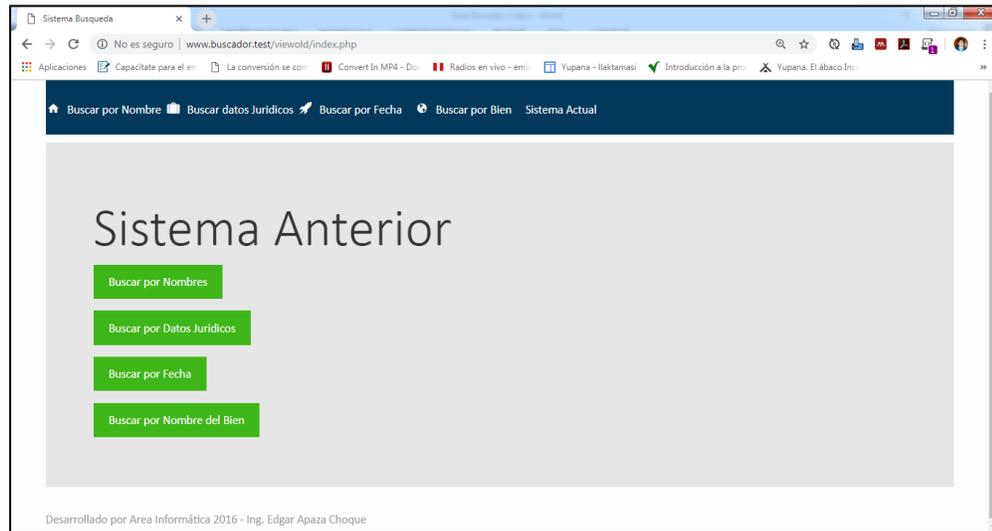


Figura 13. Botones para buscar, por nombre, datos jurídicos, búsqueda por fecha

El sistema permite la búsqueda por nombres de las personas, por nombres de empresas u organizaciones, por fecha o por nombre del bien.

La siguiente Figura muestra la búsqueda de todas las personas con apellido paterno “castellanos”, y que nos muestra en cuestión de segundos.

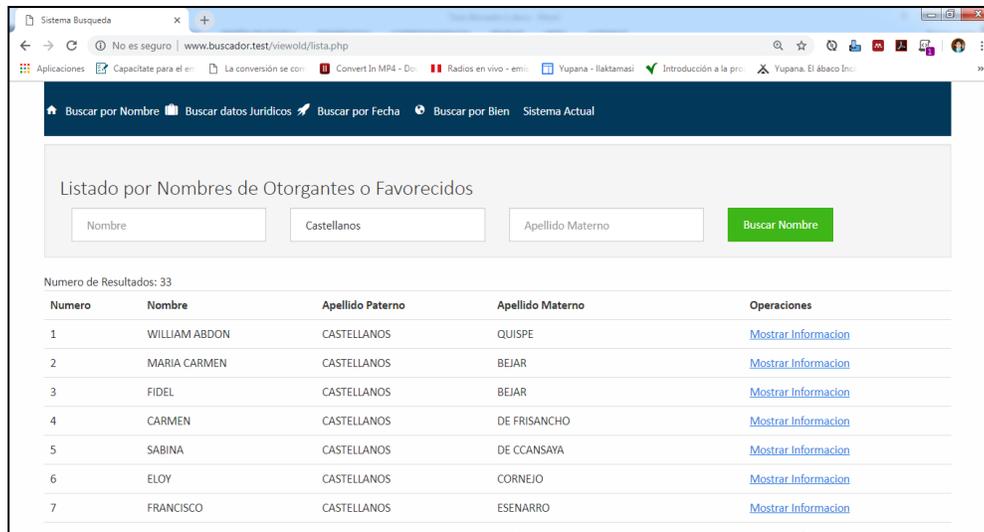


Figura 14. Resultado de la búsqueda (Ejemplo: Apellido paterno castellanos)

Al presionar en el link “mostrar información”, muestra información sobre la cantidad de escrituras que ha comprado o a vendido de forma ordenada, tal como se muestra en la figura.

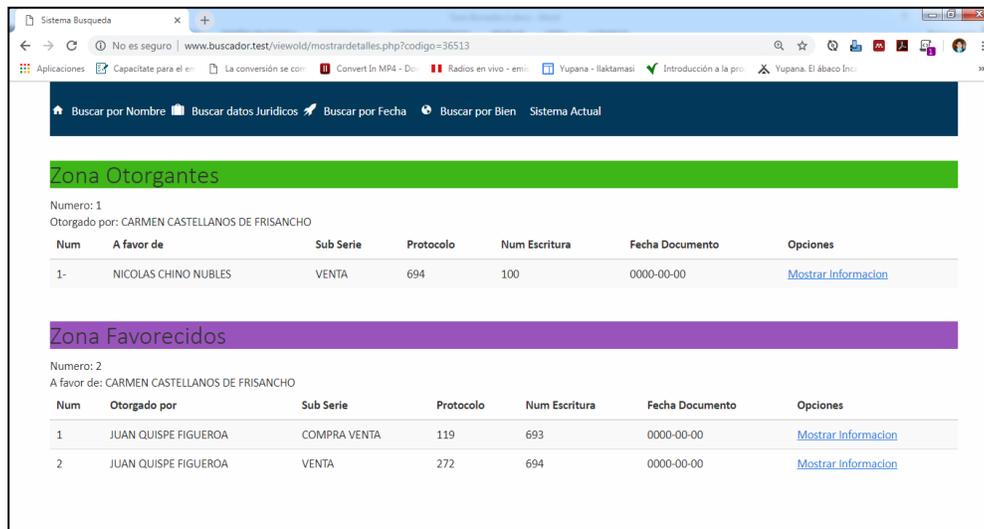


Figura 15. Lista de casos cuando estos Venden (Otorgantes) y cuando adquieren un bien (Favorecidos)

Lo mismo ocurre si buscamos por datos jurídicos, como en este ejemplo escribimos la palabra colegio, obtenemos el siguiente resultado.

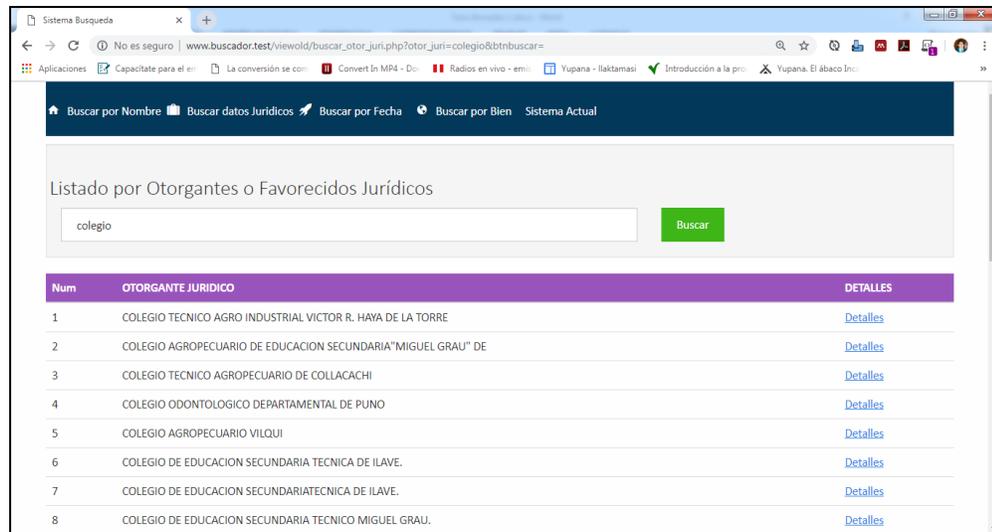


Figura 16. Búsqueda por nombres de Personas o empresas de carácter jurídico

Al presionar el botón detalles, obtenemos la siguiente información.

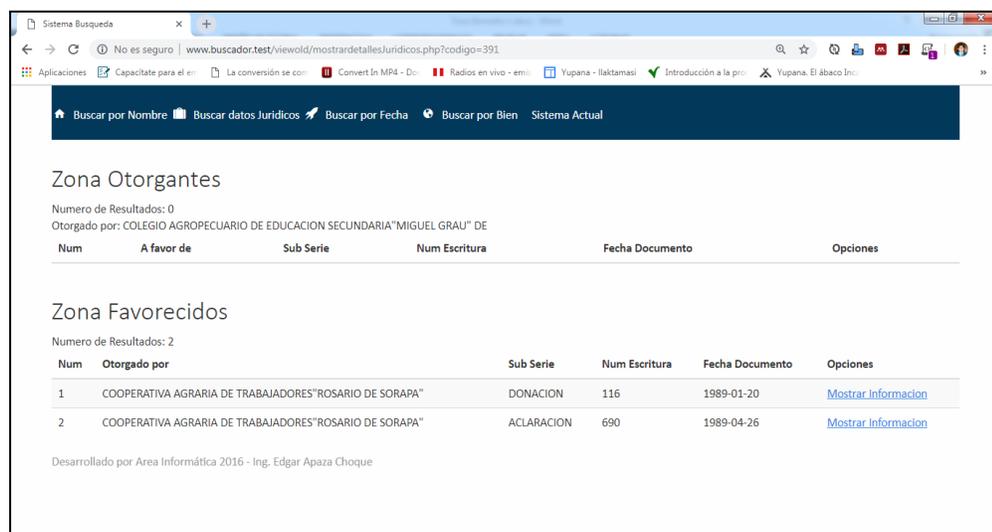


Figura 17. Muestra la información del resultado de la búsqueda por persona jurídica

La opción de búsqueda por fechas es también muy útil si se quiere buscar por una fecha específica o por los datos del mes o año, si se desea, aunque tarda más tiempo, la información es mucho más precisa.

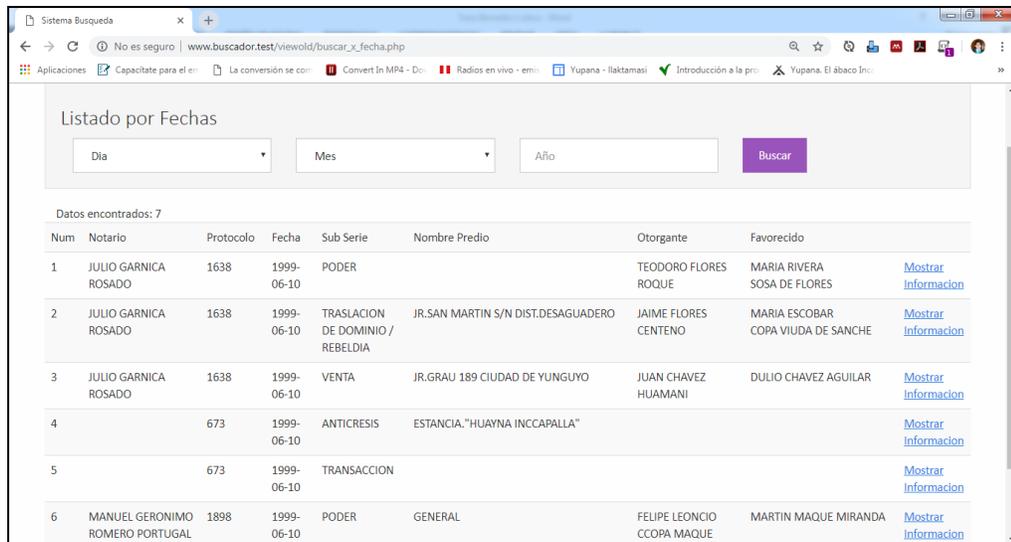


Figura 18. Ejemplo de búsqueda por fecha

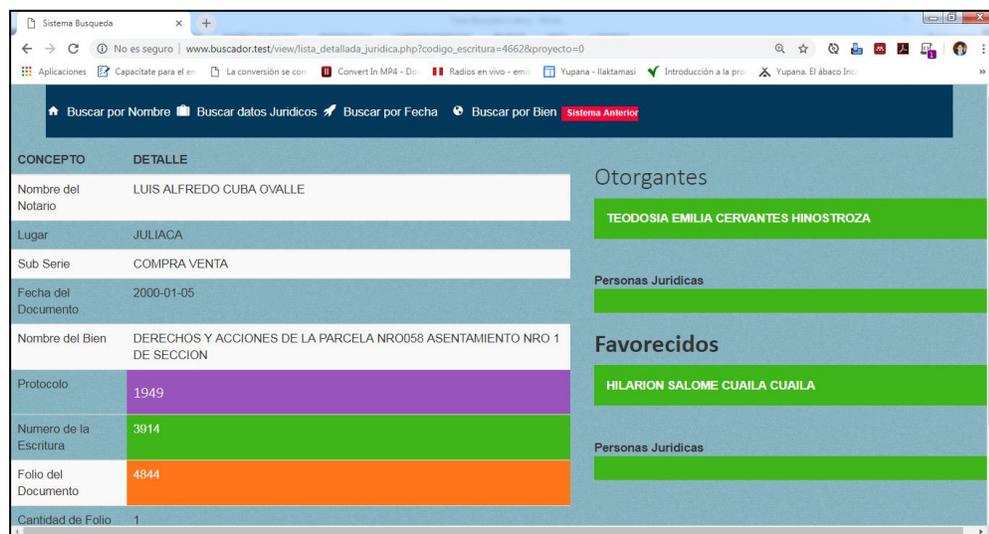


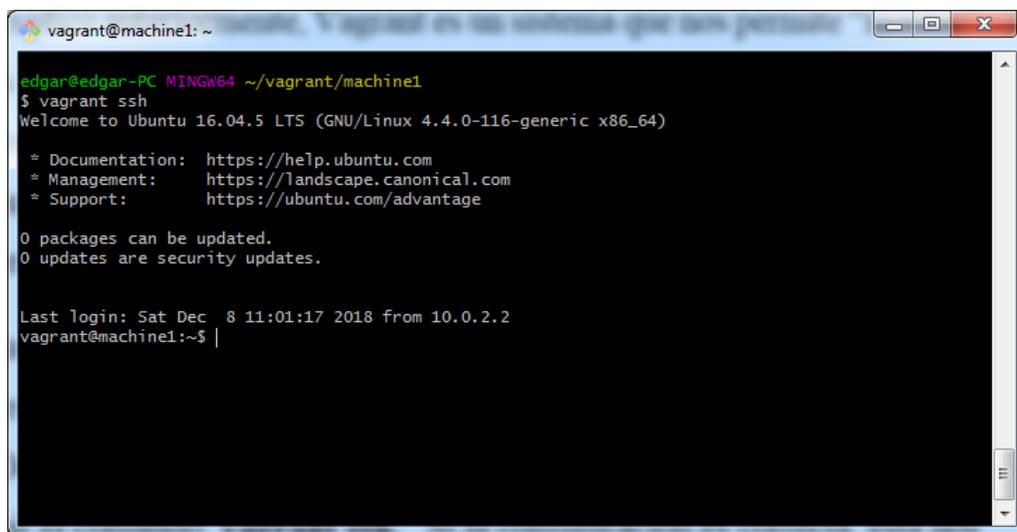
Figura 19. Detalle de los datos de la escritura

Esta es la ventana más importante, debido a que muestra información acerca de la escritura, por el ejemplo el nombre del notario, lugar, nombre del bien, numero de protocolo número de la escritura y el folio, quien lo vendió y quien lo compro.

Configuraciones avanzadas con Vagrant

Como ya explico anteriormente, Vagrant es un sistema que nos permite “montar” entornos (máquinas virtuales) de manera fácil, replicables y configurables, de acuerdo a nuestras necesidades. La facilidad que no proporciona Vagrant es que esta máquina se puede compartir con diferentes usuarios que usan distintos sistemas operativos, lo único que se necesita es tener instalado Vagrant y un VirtualBox para desplegar todo su poder.

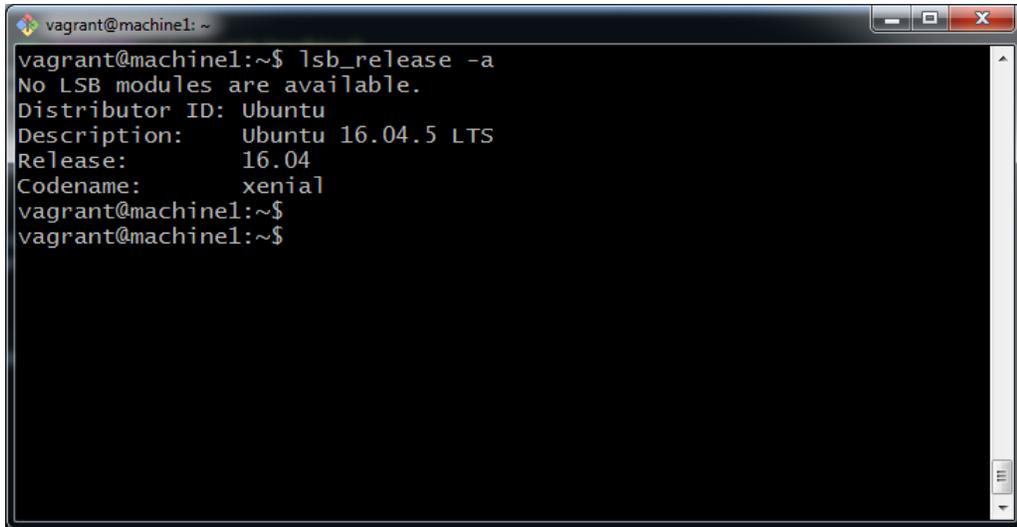
Ahora se mostrará cómo podemos ingresar a la máquina de Linux Ubuntu y realizar configuraciones, instalar paquetes, actualizar el sistema operativo o crear llaves SSL, entre otras cosas. De manera inicial, debemos tener el Vagrant corriendo, y luego de eso para ingresar escribir el comando: **Vagrant SSH**. Si la configuración es correcta, esta nos mostrará el siguiente resultado.



```
vagrant@machine1: ~  
edgar@edgar-PC MINGW64 ~/vagrant/machine1  
$ vagrant ssh  
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-116-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
0 packages can be updated.  
0 updates are security updates.  
  
Last login: Sat Dec  8 11:01:17 2018 from 10.0.2.2  
vagrant@machine1:~$
```

Figura 20. Ingresando a la máquina virtual usando el protocolo SSH

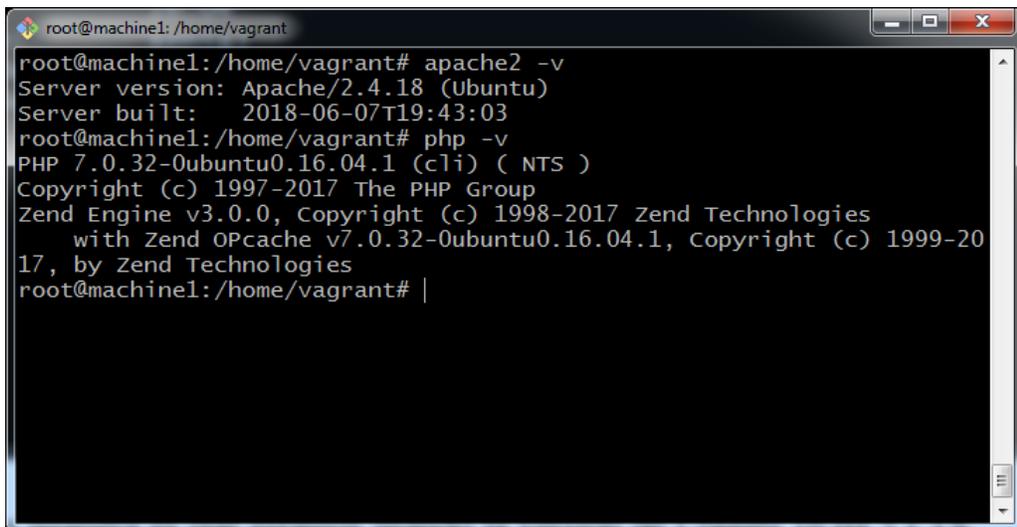
Como se puede apreciar ya estamos dentro del sistema operativo Linux, y para probarlo escribiremos el comando `lsb_release -a`



```
vagrant@machine1: ~  
vagrant@machine1:~$ lsb_release -a  
No LSB modules are available.  
Distributor ID: Ubuntu  
Description:    Ubuntu 16.04.5 LTS  
Release:        16.04  
Codename:       xenial  
vagrant@machine1:~$  
vagrant@machine1:~$
```

Figura 21. La Figura muestra la versión del servidor que está configurado y la versión de Ubuntu

Lo que nos muestra es que estamos usando la versión de Ubuntu 16.04.5 LTS, actualizado y con varios programas ya instalados, como se ve en las siguientes imágenes.



```
root@machine1: /home/vagrant  
root@machine1:/home/vagrant# apache2 -v  
Server version: Apache/2.4.18 (Ubuntu)  
Server built:   2018-06-07T19:43:03  
root@machine1:/home/vagrant# php -v  
PHP 7.0.32-0ubuntu0.16.04.1 (cli) ( NTS )  
Copyright (c) 1997-2017 The PHP Group  
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies  
    with Zend OPcache v7.0.32-0ubuntu0.16.04.1, Copyright (c) 1999-2017,  
    by Zend Technologies  
root@machine1:/home/vagrant# |
```

Figura 22. Muestra las versiones del servidor APACHE, PHP Y MySQL (MariaDB)

Aquí se ve la versión de apache2 que se está usando que es la versión 2.4.18, y la versión de PHP 7.0.32 para Ubuntu 16.04.

```
root@machine1: /home/vagrant
root@machine1: /home/vagrant# mysql --version
mysql Ver 15.1 Distrib 10.0.36-MariaDB, for debian-linux-gnu (x86_64)
using readline 5.2
root@machine1: /home/vagrant#
```

Figura 23. Vista del Servidor Ubuntu server dentro de vagrant

Aquí se muestra la versión de mariadb usada en este proyecto, que es la versión 10.0.36 para Linux.

```
root@machine1: /home/vagrant
• apache2.service - LSB: Apache2 web server
  Loaded: loaded (/etc/init.d/apache2; bad; vendor preset: enabled)
  Drop-In: /lib/systemd/system/apache2.service.d
           └─apache2-systemd.conf
  Active: active (running) since Tue 2018-12-11 03:52:30 UTC; 54min ago
  Docs: man:systemd-sysv-generator(8)
  Process: 1191 ExecStart=/etc/init.d/apache2 start (code=exited, status=0/SU
  Tasks: 10
  Memory: 36.3M
  CPU: 4.873s
  CGroup: /system.slice/apache2.service
          └─1546 /usr/sbin/apache2 -k start
             1717 /usr/sbin/apache2 -k start
             1718 /usr/sbin/apache2 -k start
             1719 /usr/sbin/apache2 -k start
             1720 /usr/sbin/apache2 -k start
             1721 /usr/sbin/apache2 -k start
             2793 /usr/sbin/apache2 -k start
lines 1-18
```

Figura 24. La Figura muestra que el servidor apache2 está activo y corriendo

Se muestra que el apache está corriendo y no tiene problemas. Para ver esta pantalla se escribe el comando: *systemctl status apache2.service*

Lo mismo ocurre si queremos ver a mariadb, Python, etc., comprobar su funcionamiento. Para salir de esta ventana se usa la letra “q”.

Las interfaces de red se muestran con el comando: *ifconfig*, y esta nos muestra la configuración de 2 Ethernet, con direcciones IPs distintas.

```
root@machine1: /home/vagrant
root@machine1: /home/vagrant# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:5f:82:b1
          inet addr:10.0.2.15  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe5f:82b1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2710 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1878 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:616042 (616.0 KB)  TX bytes:193104 (193.1 KB)

eth1      Link encap:Ethernet  HWaddr 08:00:27:d4:ae:18
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed4:ae18/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:742 errors:0 dropped:0 overruns:0 frame:0
          TX packets:364 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:83147 (83.1 KB)  TX bytes:122629 (122.6 KB)
          Interrupt:16 Base address:0xd240

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
```

Figura 25. La configuración de las interfaces de red en Ubuntu

La que utilizamos para el proyecto es la *eth1* con la dirección IP 192.168.56.101, que si escribimos esta dirección en cualquier navegador obtendremos el siguiente resultado.

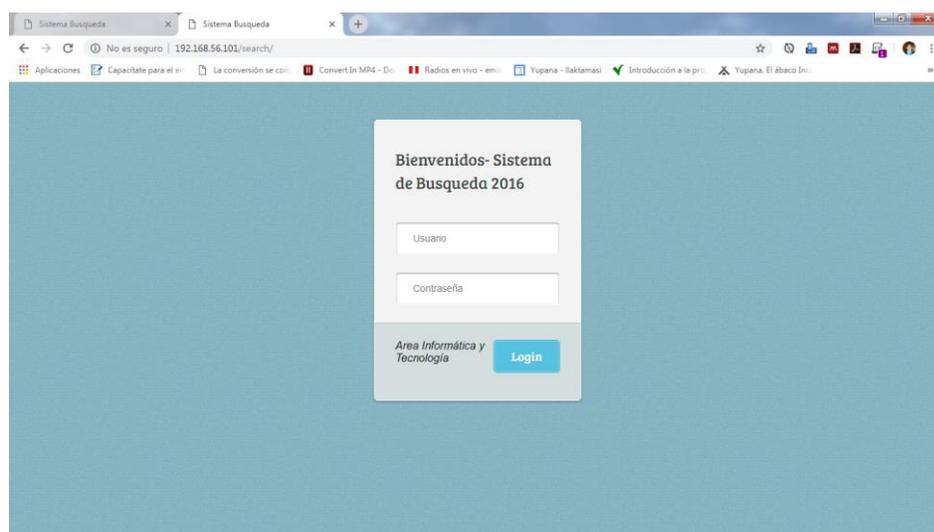


Figura 26. Acceso al software usando la dirección del servidor

3.5.4 Software de Ingreso al Fondo Notarial

Este software fue desarrollado usando la versión de Visual Studio .NET, para el ingreso de escrituras públicas. Este programa se desarrolló en reemplazo de la vieja versión web de ingreso de escrituras, que en este proyecto no se mostrarán.

Se usó las herramientas de empaquetado del mismo Visual Studio para el desarrollo de este aplicativo. Las siguientes capturas de pantalla muestran el trabajo desarrollado.

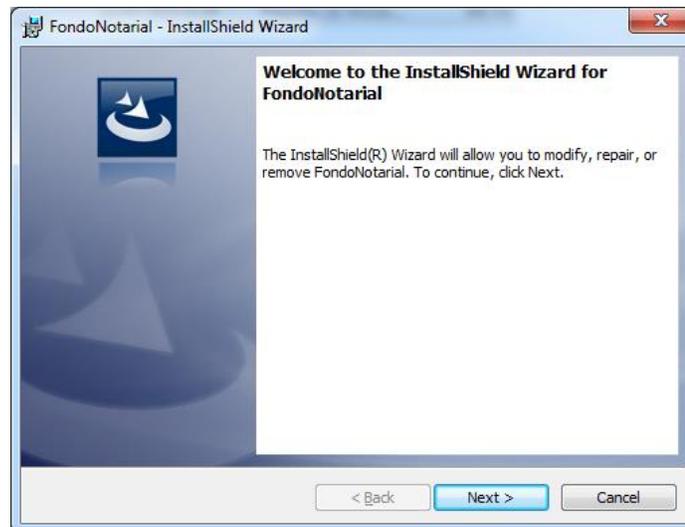


Figura 27. Ejecutando el instalador del Sistema Fondo Notarial

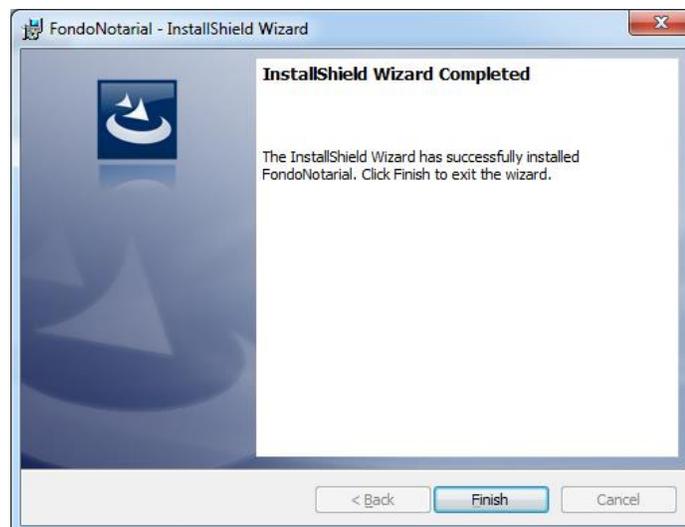


Figura 28. Vista del Asistente de Instalación del Software Notarial

Una vez instalado, procedemos a configurar la conexión, la que se guarda en un archivo de texto dentro del instalador.

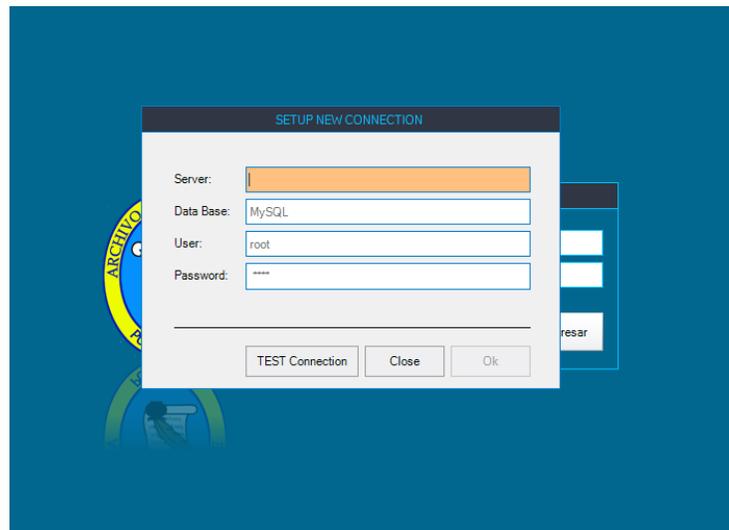


Figura 29. Ingreso de configuración para la aplicación

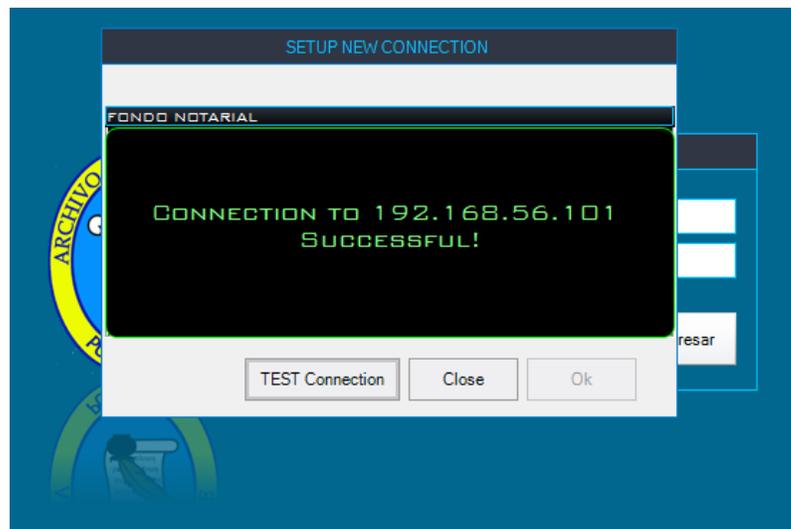


Figura 30. Vista de Establecer conexión al sistema

Una vez “logueado” en el sistema, podemos conectarnos a la base de datos, para poder ingresar la información de los libros notariales.

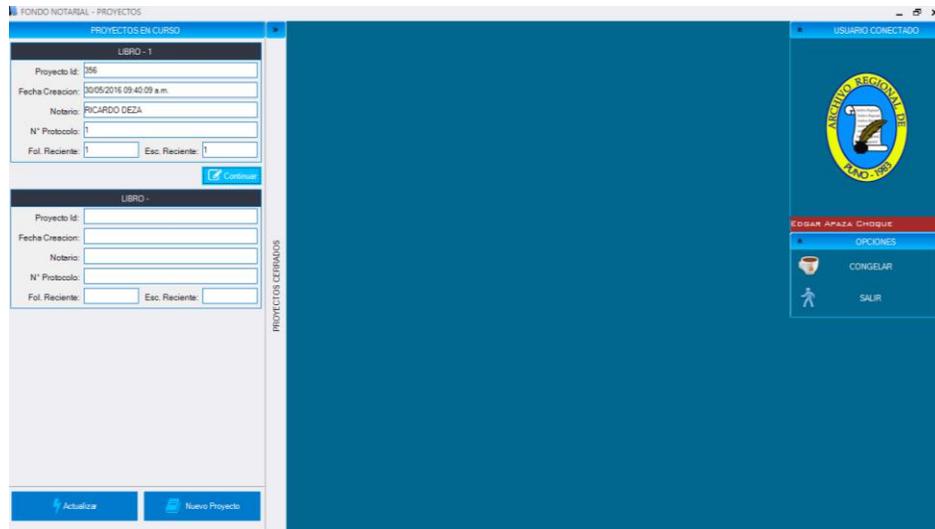


Figura 31. Pantalla inicial del software de ingreso de escrituras

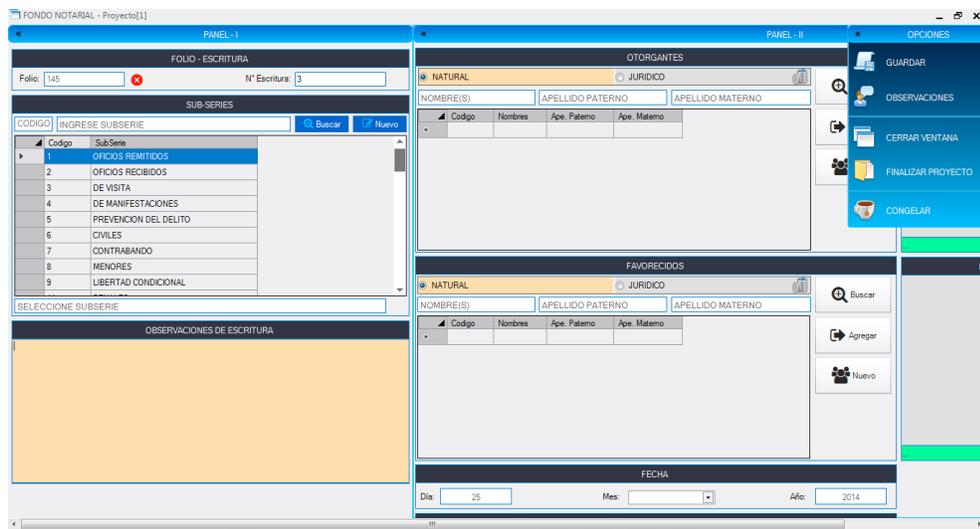


Figura 32. Pantalla de ingreso de escrituras públicas

3.5.5 Uso de sistema de control de versiones GIT

Tal como se describe en (Wikipedia, 2018) Git (pronunciado "guit") es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia y la confiabilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. Su propósito es llevar registro de los cambios en archivos de computadora y coordinar el trabajo que varias personas realizan sobre archivos compartidos.

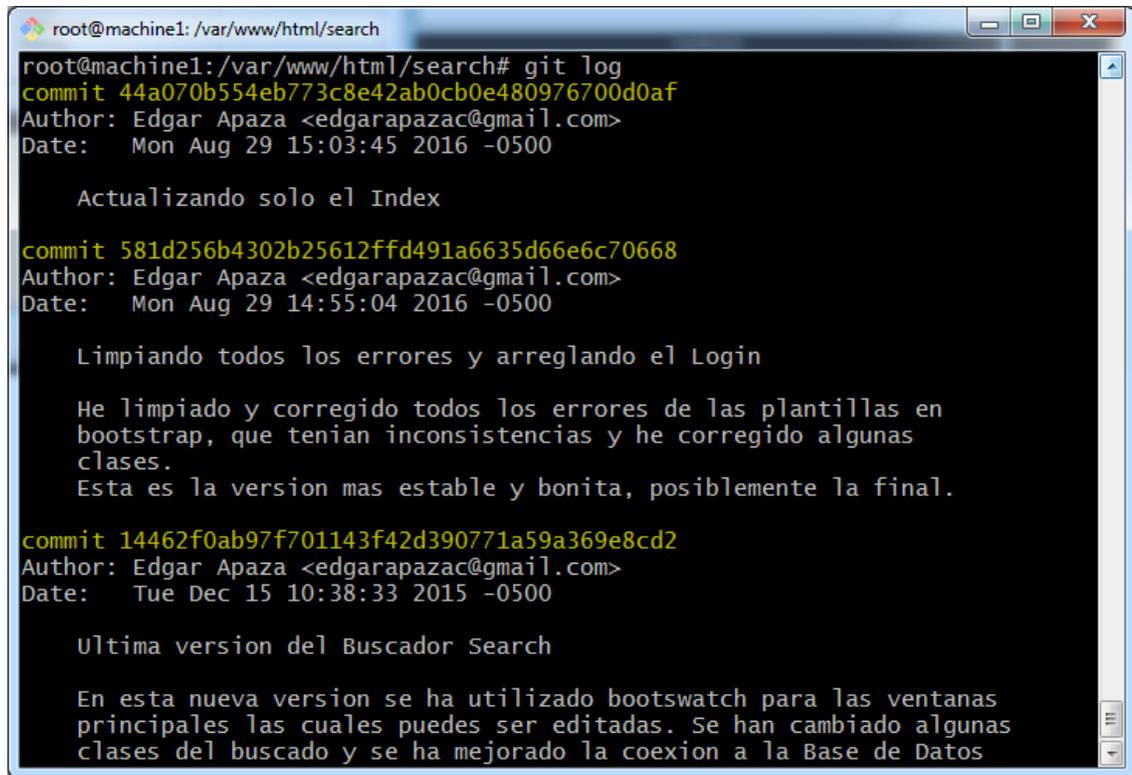
Sin duda uno de los programas que se empezó a usar por muchos desarrolladores y que desplazo a subversión y mercurial, que eran los sistemas de control de versiones de otras empresas. Uno de los proyecto en web más importantes fue GitHub, según Platzi (2018) “GitHub es un servicio de alojamiento que ofrece a los desarrolladores repositorios de software usando el sistema de control de versiones, Git”. Actualmente Git es propiedad de Microsoft y su popularidad ha caído bastante, y se han eliminado muchos repositorios migrando a Git Labs, que es el que actualmente estamos utilizando. GitHub permite que alojemos proyectos en repositorios de forma gratuita.

Aquí presentamos una muestra sobre el trabajo que se realizó en nuestro servidor de pruebas y desarrollo, el cual se puede observar el historial de procesos que se hicieron en Git, cuando se estaba en proceso de desarrollo. Para obtener esta información, basta con escribir en la consola, dentro de la carpeta del proyecto: **git log** esto mostrará un listado detallado de cuando hicimos y subimos cambios importantes en el sistema.

Aunque muchas veces, hemos destruido por completo todo el código fuente, una de las ventajas mas importantes que logramos con Git es la de crear ramas.

Una rama es una copia fiel del proyecto original, pero que se puede trabajar de manera independiente en ella, sin que esta afecte al código original que no tiene fallas. Por ejemplo, si se quiere probar una nueva funcionalidad, no lo probamos en la rama original (código funcionando), sino que creamos una rama a partir de la que funciona, le ponemos un nombre como prueba1, y en esa rama hacemos los cambios, probamos la nueva funcionalidad, y si algo falla, no tenemos problemas, simplemente eliminamos la rama creada, y volvemos a crear otra.

Sin embargo, si todo el código funciona bien, y vale la pena incluirlo en el código fuentes, entonces de proceder a realizar un **merge** (fusión en inglés) de la rama prueba1 con la rama **master**, y de esa manera se consigue una nueva versión del programa. Es nos resultó muy útil a la hora de escribir y mejorar el programa.



```
root@machine1: /var/www/html/search
root@machine1:/var/www/html/search# git log
commit 44a070b554eb773c8e42ab0cb0e480976700d0af
Author: Edgar Apaza <edgarapazac@gmail.com>
Date:   Mon Aug 29 15:03:45 2016 -0500

    Actualizando solo el Index

commit 581d256b4302b25612ffd491a6635d66e6c70668
Author: Edgar Apaza <edgarapazac@gmail.com>
Date:   Mon Aug 29 14:55:04 2016 -0500

    Limpiando todos los errores y arreglando el Login

    He limpiado y corregido todos los errores de las plantillas en
    bootstrap, que tenían inconsistencias y he corregido algunas
    clases.
    Esta es la version mas estable y bonita, posiblemente la final.

commit 14462f0ab97f701143f42d390771a59a369e8cd2
Author: Edgar Apaza <edgarapazac@gmail.com>
Date:   Tue Dec 15 10:38:33 2015 -0500

    Ultima version del Buscador Search

    En esta nueva version se ha utilizado bootswatch para las ventanas
    principales las cuales puedes ser editadas. Se han cambiado algunas
    clases del buscado y se ha mejorado la coexion a la Base de Datos
```

Figura 33. Vista del sistema GIT para el control de Versiones – Sistema Fondo Notarial

Git es muy poderoso y cuenta con gran cantidad de funcionalidades desde las mas básicas, hasta muy complejas.

Planificación del proyecto

A pesar de ser un proyecto ágil, no hemos renunciado a una mínima planificación, inicialmente identificamos las siguientes etapas:

Investigación tecnología

Se dedicó un tiempo razonable para el análisis, diseño y obtención de los requisitos del sistema, los cuales nos facilitaron el trabajo de codificación. También se utilizó los test unitarios, aunque en esta parte no nos dedicaremos a mostrar estos test unitarios, debido a que se perdieron, sin embargo, fruto de estos test unitarios tenemos las clases en el sistema creados y funcionando, que era para lo que inicialmente servía.

Instalación de entorno

Antes de empezar, instalaremos las siguientes herramientas y software requerido para nuestro proyecto.

- Vagrant (<https://www.vagrantup.com>)
- VirtualBox (<https://www.virtualbox.org/>) version 5.0
- PuPHPet (<https://puphpet.com/>)
- Git bash (<https://git-scm.com/downloads>)

Para el servidor Linux (después de instalar y correo Vagrant)

Aquí se presentan los códigos para la instalación de los paquetes necesarios en el servidor Linux Ubuntu 16.04 LTS. (Los comandos están ejecutados como súper usuario - root)

- apt-get update yy apt-get upgrade
- apt-get install apache2
- apt-get install php php-mysql php-fpm
- apt-get install mariadb-server
- apt-get install git git-core
- apt-get install vim rar
- etc.

Establecemos también un sistema de backup automatizado a una unidad interna, en otro disco duro con un sistema de archivos NTFS haciendo uso de CRON en Linux.

3.5.6 Desarrollo de la aplicación por casos de uso (sprints)

Scrum como lo describe Schwaber y Sutherland (2013) indica que “Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del

riesgo. Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación”. (p.4)

Schwaber y Sutherland (2013) añade que “el corazón de Scrum es el sprint”, y tiene mucho sentido, ya que este “es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento del producto”, en el cual se utilizan las métricas de análisis, diseño e implementación, pero limitado por el sprint. “Cada nuevo sprint comienza inmediatamente después de la finalización” del sprint anterior, lo que permite una flujo constante de desarrollo y que los entregables se hagan a tiempo.

Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective). (Schwaber y Sutherland, 2013,p.9)

Sprint1. Ingreso de Escrituras

Es la parte más crítica, la de la gestión del módulo de ingreso de escrituras públicas, debido a que se realizó un análisis de los datos, una estimación de costos y tiempo por cada campo que se debe registrar, los test de prueba para comprobar la consistencia de la base de datos planteada, así, como las interacciones entre las maquinas.

Uno de los principales desafíos fue el sistema operativo que se debía usar. Inicialmente se instaló una Pc con sistema operativo Windows Server 2003, a la que se configuró algunas partes, pero por el bajo conocimiento de las características y funcionalidad que Windows ofrece, es que se decidió usar una distribución de Linux.

La primera versión en probar fue FEDORA, después por incompatibilidad con algunos componentes de hardware se instaló CentOS la versión 5.6 la cual trajo nuevas herramientas y una mejor configuración de los componentes del equipo.

La computadora (servidor) fue de la marca IBM x3400, con un solo procesador y 4 GB de RAM. Originalmente de fábrica tenía el Windows server, el cual se cambió a Linux. Con 2 Discos duros SAS de 500GB cada uno, inicialmente se unió a estos dos discos duros separados, como si fuera un solo disco duro de 1 TB. Esto nos ocasionó muchos problemas, debido que no podíamos generar copias de seguridad sobre el mismo sistema operativo, por



lo que se procedió nuevamente a separar los discos en unidades distintas, una con ext3 y el otro disco en NTFS.

Una vez instalado el sistema operativo, se procedió al análisis de la información por libros (protocolos) y realizar una tarea denominada particionamiento vertical y horizontal en bases de datos.

También se utilizó la normalización de bases de datos, lo que nos permitió probar algunas particularidades del sistema, así como la utilización de vistas, store procedure e inclusive triggers. Estos nuevos elementos, nos dan la posibilidad de que se pueda automatizar algunas funciones como el pedir datos y registros, actualizar, crear vistas a partir de otras Tablas, etc.

Sprint2. Búsqueda de Datos

Se desarrolló un módulo de búsqueda la cual fue escrita en PHP con un servidor APACHE, inicialmente con AppServer, después con XAMPP, y finalmente hasta la fecha con una distribución Linux server. Aunque el sistema ha pasado por diversas versiones como Fedora, RedHat, CentOS, Ubuntu Server, Zentyal, entre otros, en todas las distribuciones de Linux el sistema han corrido muy bien. Se han experimentado algunos inconvenientes, mucho más por el desconocimiento de algunas funcionalidades, sin embargo, estas se fueron superando poco a poco en base a las necesidades.

En esta parte se mostrará la forma y código fuente de cómo es que se desarrolló el sistema.

Sprint 3. Login y seguridad de la aplicación

Con el sistema funcionando, se realizó un sistema de login muy básico la cual contempla el uso de sesiones (SESSION) en PHP, el que nos proporciona una consulta a la base de datos y recupera de ella el tipo de usuario, el nivel y los coloca en una sesión de PHP.

Posteriormente a este desarrollo, en este Sprint, se ha creado una página de “Login” mas pulida usando CSS3, JavaScript, y el modelos aplicado fue el MVC, asegurando la aplicación mediante envíos de datos por POST. Mejorado esta parte, la accesibilidad mejoró y también la creación de nuevos usuarios y nuevas funcionalidades.

Sprint 4. Correcciones y Control de Calidad

Este Sprint es el más largo de todos, debido a que se debió analizar la información introducida y mediante una serie de códigos, la idea principal, es la de revisar escritura por escritura lo ya desarrollado y comparado con el libro, para validar su integridad y confiabilidad de los datos. Junto con el Sprint 1, este Sprint 4 es uno de los que consideramos nosotros que es el más valioso, debido a que depende de esta instancia, que se publiquen y muestren los datos de manera correcta.

Se han planteado una serie de mecanismos para que este control sea lo más ágil posible, el cual permite que se introduzca el número del libro (Protocolo) y este automáticamente realiza una consulta a la base de datos, y coloca los resultados de esta consulta en un array que podemos recorrer con JavaScript, lo que agiliza este proceso.

3.5.7 Alcance del proyecto final

El rigor con el que hemos llevado los test, ha hecho que nos tengamos que replantear el alcance del proyecto, algo que es perfectamente plausible siguiendo la filosofía del Desarrollo Ágil.

A título personal, hasta la fecha y en base a las mejoras que se la efectuado – gracias a las aportaciones de los mismos trabajadores –, este software se ido puliendo cada vez más, y está a punto de convertirse en un sistema experto (si no lo es en la actualidad), debido a que es único en su clase, y posee formas y características que la hacen extensible a mas mejoras y cambios.

3.5.8 Metodología de desarrollo del software

Para el análisis, diseño y desarrollo de este proyecto se utilizó la combinación de las metodologías Scrum y eXtreme Programming, la que nos proporciona el marco de trabajo para el desarrollo de las funcionalidades, requeridas.

Alcance del sistema

Este proyecto Software Sistema Notarial, realizado para el archivo regional, la oficina de Archivo Intermedio, y no es para toda la institución, de eso modo es que limitamos nuestro accionar solo para archivo intermedio. Aunque podemos indicar que se ha realizado mejoras en el área de recepción de documentos, la oficina de cobranzas (caja), aunque este tema no afecta a nuestro proyecto final, sin embargo, ayuda de forma “interesante”, para mantener un orden y una mejor manera para efectuar las búsquedas de documentos, de tipo notarial.

Para llevar a cabo estas tareas, se separaron los ítems por los siguientes módulos.

Tabla 7

Funciones del sistema Búsqueda

FUNCIONES	DESCRIPCIÓN
Módulo de Seguridad	El objetivo de este es administrar los permisos que tendrán los usuarios y controlar su acceso a cada una de las ventanas del sistema.
Búsqueda y muestra de resultados	Una vez logueado, se puede realizar la búsqueda de documentos y visualizar su información
Módulo informes y consultas	Generará la impresión de reportes solicitados por la institución

El modelo de búsqueda depende del Módulo de ingreso de datos, el cual se detallará más adelante.

Limitaciones del sistema

Este sistema tiene como principal limitación la digitalización de los documentos. Inicialmente se había planteado la idea de ingresar la información y al mismo tiempo digitalizarla, esto no se dio debido a que no se contaba con las maquinas necesarias ni el equipamiento, condiciones de trabajado, capacitación para llevar a cabo este proyecto, sin embargo, la base de datos creada, contempla esta posibilidad, permitiendo enlazar el registro de la escritura a una ruta con la Figura de esta.

Este es un proyecto bastante ambicioso, que demandará mucho tiempo y un gasto de presupuesto considerable.

3.5.8 Análisis del sistema

Análisis del sistema Actual

En el ARP no existe un software de esta naturaleza, con tales características, las cuales están centradas en la optimización del tiempo de respuesta que hay entre que entra el documento hasta el momento que sale. El tiempo de atención solo se puede observar desde adentro de la oficina de archivo intermedio, mas no se puede reflejar a los usuarios, debido a las siguientes razones.

Los usuarios se acercan al módulo de solicitud de expedientes, y este entrega los datos a la persona, y este los ingresa al sistema de recepción de solicitudes, que también desarrollamos. Una vez impreso 2 copias, una para el usuario y otra para archivo intermedio, histórico u otra oficina, se paga y registra la información en el software de caja (también desarrollado por nosotros).

El sistema de búsqueda ayuda especialmente a los trabajadores del ARP, a ubicar los archivos de manera mas rápida, y ubicar los protocolos, sin embargo, la principal limitación es que se tiene a una sola personas denominada “calificador” el cual es encarga de leer el documento completo, identificar las firmas y problemas que pueda tener una escritura pública, desde que el notario no ha firmado en la escritura, error en el folio, borroneos, espacios en blanco, etc., las cuales es otra limitante que presenta este software, que claramente requiere un mayor trabajo y que sirva para una nueva versión que pueda incorporar estas mejoras, que hasta la

fecha no es posible, dejando a un trabajador del archivo realizar las veces de “verificador de escrituras”, y esto se efectúa en base a la normativa dada por el Archivo General de la Nación.

El sistema del fondo notarial, funciona y cumple con su objetivo, pero se ve limitado a la intervención humana en el proceso de expedición de documentos, para lo cual ahora se mostrará la forma en que este proceso ocurre.

MODULO DE BÚSQUEDA DE ESCRITURAS

Diagramas de Casos de Uso.

Grafico 01. Ingreso al Sistema de FONDO NOTARIAL

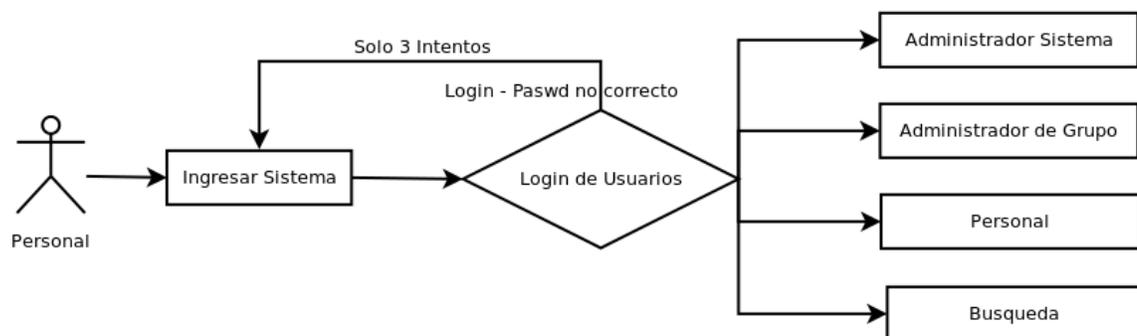


Figura 34. Diagrama de ingreso al sistema (Modulo de Seguridad)

Como se puede apreciar en el grafico 1, el usuario que se conecta a nuestro sistema debe pasar por un proceso de validación de credenciales –aunque hay mucho que explicar en este proceso–, solo nos limitaremos a indicar que se utiliza la encriptación como medio de seguridad, el pasar las variables de usuario y password mediante el modelo MVC, y asegurar que las carpetas y archivos no estén disponibles para nadie que no sea el administrador. Es uno de los métodos de seguridad básicos, al cual podemos incrementar su seguridad, pero hasta la fecha no habido esta necesidad.

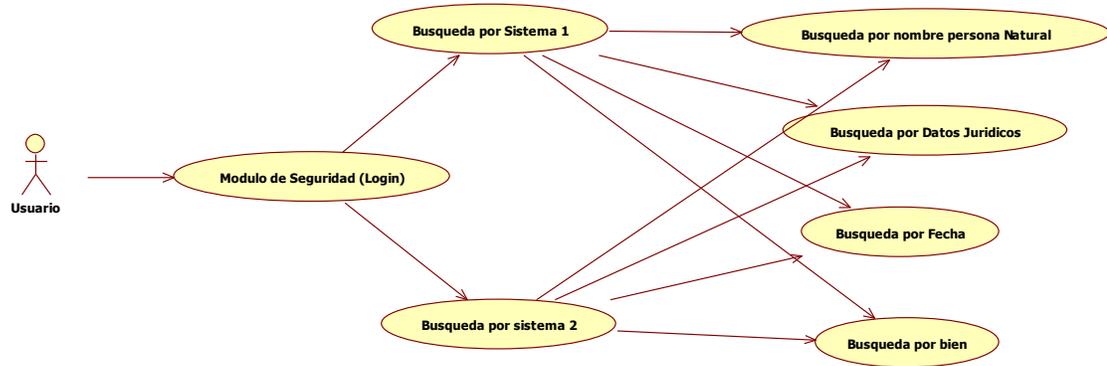


Figura 35. Casos de uso del módulo de búsquedas

Como se aprecia en el diagrama 2, el personal tiene acceso a las siguientes acciones, después de loguearse en el sistema: Búsqueda en los 2 sistemas que se tiene.

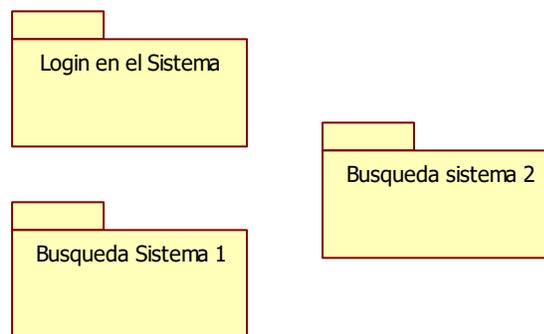


Figura 36. Diagrama de paquete. Buscar Escrituras

MODELOS PARA BÚSQUEDA DE ESCRITURAS

Grafico 04. Personal - SISTEMA FONDO NOTARIAL
Viene del Grafico 01.

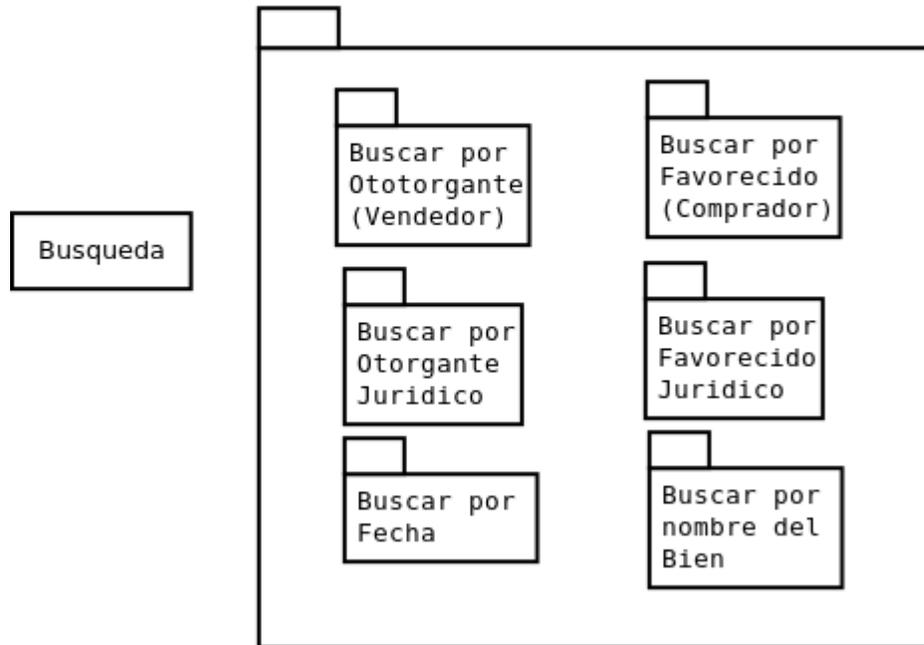


Figura 37. Diagrama de paquetes para búsqueda

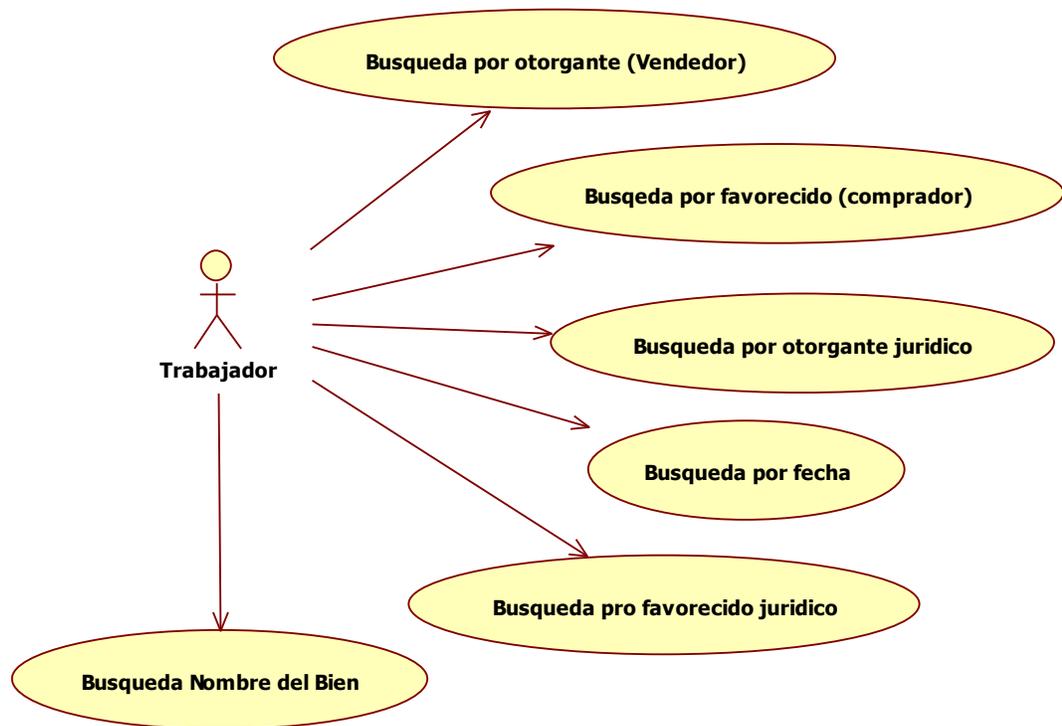


Figura 38. Diagrama de Casos de Uso – Trabajador búsquedas

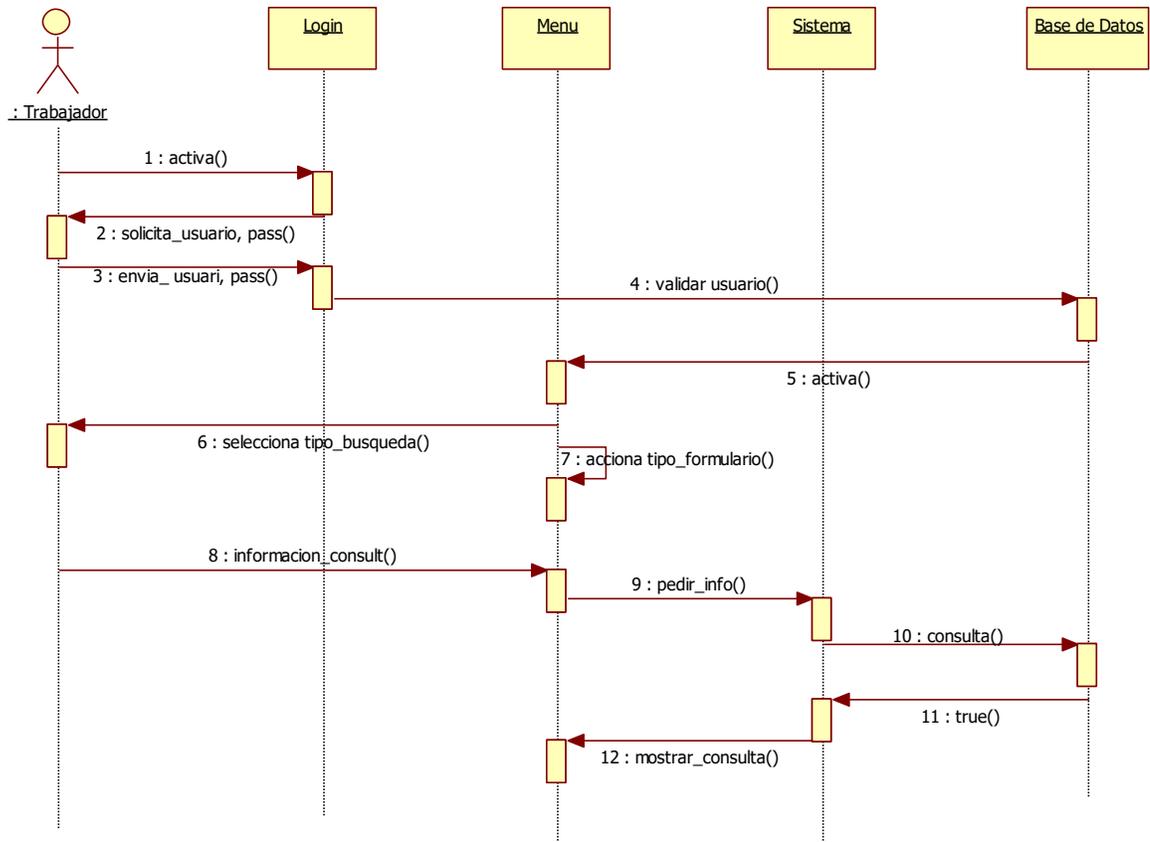


Figura 39. Diagrama de Secuencia. Búsqueda por Comprador

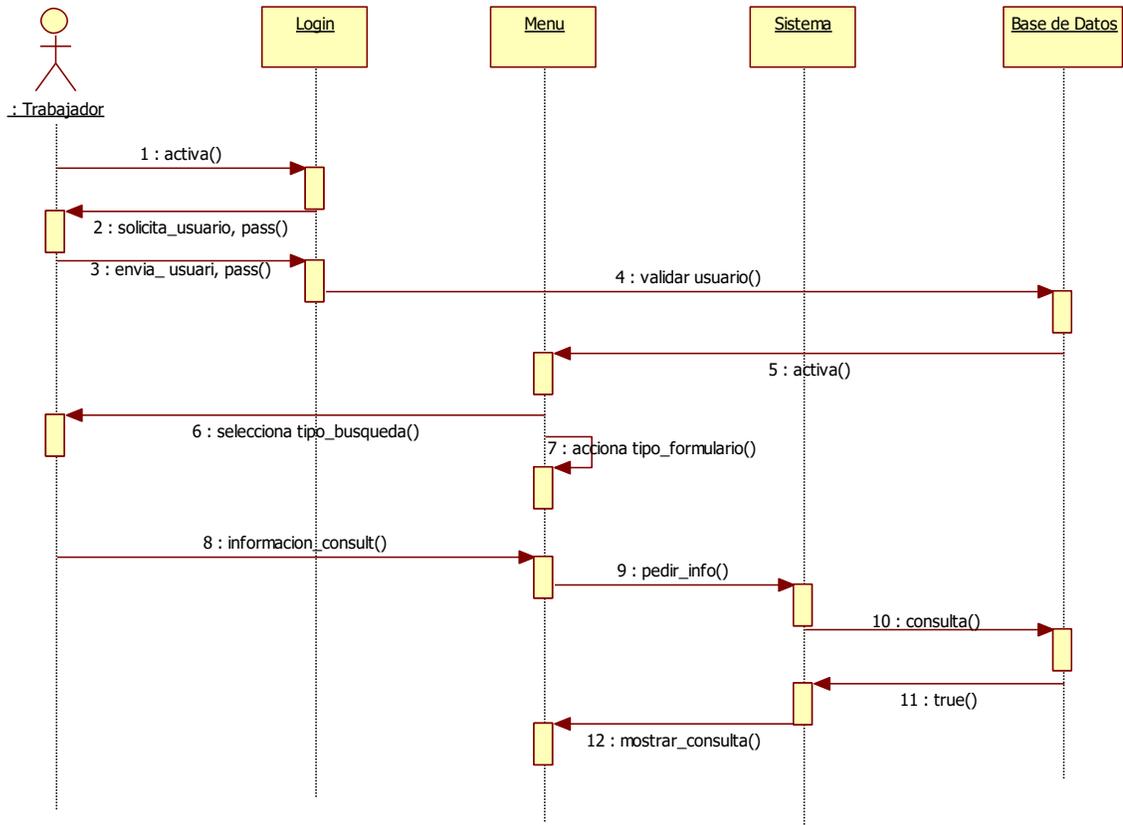


Figura 40. Diagrama de Secuencia. Búsqueda por Vendedor

3.5.9 Planificación del proyecto utilizando la metodología SCRUM

Una vez obtenidas las especificaciones (requerimientos), se procede a armar la pila de producto, que es el documento más relevante en el desarrollo de proyectos con Scrum. Esencialmente es una lista priorizada de los requerimientos principales implicados en el presente proyecto. A cada ítem de la pila se les denomina historias de usuario.

Según Schwaber y Sutherland (2013) El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se desarrolla un incremento de producto, y se cataloga como “Terminado”, disponible y potencialmente desplegable. Es más provechoso si la duración de los Sprints es permanente a lo largo del esfuerzo de desarrollo de la aplicación. Cada nuevo Sprint comienza prontamente después de la finalización del Sprint antepuesto.

Un Sprints consiste de una Reunión de Planificación del Sprint (Sprint Planning Meeting), los Scrums Diarios (Daily Scrums), el trabajo de desarrollo, la Revisión del Sprint (Sprint Review), y la Retrospectiva del Sprint (Sprint Retrospective). (Takeuchi et al., 2014, p.9).

De los diferentes campos recomendados para identificar cada historia de usuario, se ha seleccionado los siguientes:

- ID: Identificado único de la historia de usuario.
- Nombre: Es una descripción corta de la historia
- Importancia: Definido por el propietario del producto, es el grado de prioridad para el desarrollo de cada historia de usuario.
- Estimación Inicial: Es la estimación inicial del equipo sobre la cantidad de trabajo que es necesario para implementar la historia, comparada con otras historias.
- Como Probarlo: Una descripción a alto nivel de cómo se demostrará esta historia en la Demo al final del Sprint
- Notas: Cualquier otra información, clarificación, referencia a otras fuentes de información.

En la Tabla 3, se puede observar que se tiene únicamente 10 requerimientos iniciales que engloban las diferentes tareas o actividades, las cuales se las detalla en una pila de tareas



(sprint backlog). De acuerdo a la metodología, el sprint backlog contiene todas las tareas a realizar y las cuales tienen un peso así como un tiempo estimado de desarrollo.

Para el desarrollo del presente proyecto, se especifica cada tarea del sprint backlog de su correspondiente requerimiento, así como toda la documentación de implantación o desarrollo de la misma.

Tabla 8

Pila de productos Inicial para el ingreso Notarial

PILA DE PRODUCTO INICIAL

ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
1	Ingreso al Sistema	15	4	1	Entrar a la aplicación, ingresar usuario y contraseña, si el usuario es correcto ingresa al menú inicial caso contrario se presentará un mensaje de error	El usuario debe estar ingresado en la base de datos.
2	Ingresa a uno de los buscadores	13	4	1	Entrar a la aplicación, y escoger a cuál debe ingresar	Esta información puede ser utilizada por todos los que lo necesiten.
3	Escoge como efectuar la búsqueda	13	1	2	Ingresa cualquier valor, numérico o de texto y comprobar su exactitud en base a los resultados que se muestra Escoger a una persona de la lista y verificar si los detalles de la aplicación, coinciden con los detalles del libro.	
4	Escoge al usuario y visualiza la información de manera detallada	12	4	2		Esto se hace para verificar cuan útil es el sistema y cuantos aciertos se obtiene y cuantos desaciertos se tiene.
5	Efectuar una nueva búsqueda	11	3	2	Existen 2 botones: 1 que la búsqueda fue satisfactoria y 2 que no es la información requerida	

Tabla 9

Pila de productos para el desarrollo del sistema de Búsquedas

PILA DE PRODUCTO INICIAL

ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
1	Crear la Tabla para el acceso al sistema	1	3	1	Revisar el diseño físico en la base de datos	
2	Recuperar el código de acceso, según tipo de trabajador	2	3	1	Verificar según el código de usuario recuperado de la Tabla usuarios, que tipo de código devuelve y re direccionarlo a la ventana respectiva.	1: Administrador 2: Personal Ingreso 3: Personal Búsqueda
3	Muestra Ventana de Inicio	3	3	1		
4	Elegir la forma de búsqueda	4	3	1	Verificar los resultados de la búsqueda y si coinciden con la información previamente ingresada	1: por nombre 2: por Datos jurídicos 3: por Fecha 4: por Nombre del bien
5	Revisar los detalles de la Escritura	5	3	1	Muestra información detallada del notario, lugar, número de protocolo, escritura, folio y coincide con la información del libro.	
6	Registra si encontró la búsqueda o no					

Las siguientes Tablas dan una muestra sobre la utilización de la metodología SCRUM y de cómo se han establecido la pila de productos.

Tabla 10

Historia de usuarios para búsqueda de información

Núm.	<i>Como Trabajador quiero que Muestre información cuando se escribe un nombre para poder Ver detalles de las escrituras y visualizar los datos de la escritura</i>	
#001		
CRITERIOS DE ACEPTACIÓN	PRIORIDAD: <u> 2 </u>	SPRINT: <u> 1 </u>
1	Ingresar datos	Ingresar un nombre y muestra todas las coincidencias
2	Limita acceso de datos	Al ingresar algún valor extraño, no muestra nada y manda un mensaje en pantalla
3		

Tabla 11

Validación de usuarios en el sistema

Núm.	<i>Como Administrador del sistema quiero que Valide los datos de usuario y contraseña y le permita ingresar al sistema para poder Efectuar las búsquedas por comprador o vendedor</i>	
#002		
CRITERIOS DE ACEPTACIÓN	PRIORIDAD: <u> 2 </u>	SPRINT: <u> 1 </u>
1	Loguin de acceso	Verificar si el usuario está activo en el sistema y recuperar el código de acceso según el tipo de usuario
2		
3		

Tabla 12

Redireccionamiento de usuarios según el código de acceso

Núm.	<i>Como Administrador del sistema quiero que Realice una consulta a la base de datos y recupere el código de acceso al personal para poder Re direccionar a la ventana de acceso según el tipo de Acceso</i>	
#003		
CRITERIOS DE ACEPTACIÓN	PRIORIDAD: <u> 2 </u>	SPRINT: <u> 1 </u>
1	Código de acceso al sistema	Recuperar el código de acceso, según tipo de trabajador
2		
3		

Este modelo lo dejaremos ahí como prueba, y en los anexos se presentarán los detalles de todas las historias de usuario y su desarrollo, llevando un control de estos avances mediante el sistema de control de versiones de GIT.

Aquí se muestra una pequeña parte de cómo se lleva a cabo el control por git. Primero debemos arrancar Vagrant, esto lo hacemos usando la terminal de Git Bash, y escribimos el comando Vagrant up.

```
MINGW64:/c/Users/edgar/vagrant/machine1
Pictures/
Plantillas@
Reciente@
'Saved Games'/
Searches/
SendTo@
Sti_Trace.log
vagrant/
Videos/
'VirtualBox VMs'/

edgar@edgar-PC MINGW64 ~
$ cd vagrant/

edgar@edgar-PC MINGW64 ~/vagrant
$ ls
machine1/  puppet.zip  server_apache/  servidor1/  software/

edgar@edgar-PC MINGW64 ~/vagrant
$ cd machine1/

edgar@edgar-PC MINGW64 ~/vagrant/machine1
$ vagrant up
```

Figura 41. Ejecutando el Git Bash y ubicarnos en la carpeta Vagrant/machine1

Esperar a que el programa termine de iniciar e ir mediante la línea de comandos hasta la ruta de donde está el proyecto, en nuestro caso que Vagrant está corriendo una máquina virtual con Linux Ubuntu, nos dirigiremos a la siguiente ruta.

```
MINGW64:/c/Users/edgar/vagrant/machine1
$ cd machine1/

edgar@edgar-PC MINGW64 ~/vagrant/machine1
$ vagrant up
==> vagrant: A new version of Vagrant is available: 2.2.2!
==> vagrant: To upgrade visit: https://www.vagrantup.com/downloads.html

Bringing machine 'machine1' up with 'virtualbox' provider...
==> machine1: Checking if box 'bento/ubuntu-16.04' is up to date...
==> machine1: Clearing any previously set forwarded ports...
==> machine1: Clearing any previously set network interfaces...
==> machine1: Preparing network interfaces based on configuration...
machine1: Adapter 1: nat
machine1: Adapter 2: hostonly
==> machine1: Forwarding ports...
machine1: 22 (guest) => 9391 (host) (adapter 1)
machine1: 22 (guest) => 2222 (host) (adapter 1)
==> machine1: Running 'pre-boot' VM customizations...
==> machine1: Booting VM...
==> machine1: Waiting for machine to boot. This may take a few minutes...
machine1: SSH address: 127.0.0.1:2222
machine1: SSH username: vagrant
machine1: SSH auth method: private key
```

Figura 42. Se muestra como inicia el Vagrant con la máquina virtual en Linux

```
MINGW64:/c/Users/edgar/vagrant/machine1
edgar@edgar-PC MINGW64 ~/vagrant/machine1
$ vagrant ssh
```

Figura 43. Uso de Vagrant SSH, para ingresar a la maquina en Linux

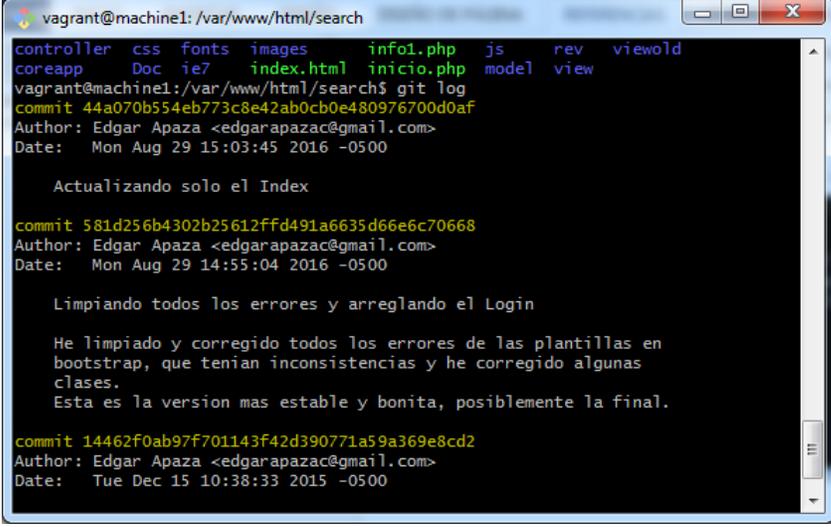
```
vagrant@machine1: /var/www/html/search
edgar@edgar-PC MINGW64 ~/vagrant/machine1
$ vagrant ssh
Welcome to Ubuntu 16.04.5 LTS (GNU/Linux 4.4.0-116-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

0 packages can be updated.
0 updates are security updates.

Last login: Tue Dec 11 04:40:17 2018 from 10.0.2.2
vagrant@machine1:~$ cd /var/www/html/search/
vagrant@machine1:/var/www/html/search$
```

Figura 44. Dentro del servidor e ingresando a la carpeta del proyecto



```
vagrant@machine1: /var/www/html/search
controller  css  fonts  images  info1.php  js  rev  viewold
coreapp     Doc  ie7    index.html  inicio.php  model  view
vagrant@machine1:/var/www/html/search$ git log
commit 44a070b554eb773c8e42ab0cb0e480976700d0af
Author: Edgar Apaza <edgarapazac@gmail.com>
Date:   Mon Aug 29 15:03:45 2016 -0500

    Actualizando solo el Index

commit 581d256b4302b25612ffd491a6635d66e6c70668
Author: Edgar Apaza <edgarapazac@gmail.com>
Date:   Mon Aug 29 14:55:04 2016 -0500

    Limpiando todos los errores y arreglando el Login

    He limpiado y corregido todos los errores de las plantillas en
    bootstrap, que tenian inconsistencias y he corregido algunas
    clases.
    Esta es la version mas estable y bonita, posiblemente la final.

commit 14462f0ab97f701143f42d390771a59a369e8cd2
Author: Edgar Apaza <edgarapazac@gmail.com>
Date:   Tue Dec 15 10:38:33 2015 -0500
```

Figura 45. Muestra del control de código y de cumplimiento de los Sprint con Git

Según esta muestra, es que todas las historias de usuarios que realizan, se escriben (registran) al completarse la tarea y que el código ya esté funcionando y sin errores.

Con respecto al sistema de control de versión Git, este solo se sube cuando el código fuente está probado, o para eliminar un código que no funciona y regresar a la versión anterior, que son alguna de las funcionalidades que nos ofrece Git. Otra herramienta que este software nos ofrece es que podemos subir nuestros repositorios locales a internet. Una de las páginas web más usadas fue la de GitHub (www.github.com), aunque actualmente, muchos proyectos se han cerrado ya que la compañía ahora es de Microsoft. Otra buena alternativa es usar Git Lab, pero existen muchas otras, que dejamos al lector a su elección.

3.5.10 Desarrollo del sistema

Una vez que se han determinado todos los requerimientos que satisfacen la documentación de la pila del producto inicial para el presente proyecto, se abordará con el desarrollo del software y su respectiva codificación.

Tomando en cuenta que la metodología Scrum hace una crónica a todas las actividades que el grupo de desarrollo realizará para alcanzar la solución a las exigencias obtenidas, por lo tanto, adicionalmente a los requerimientos del ARP, se deberán incluir los requerimientos que implican la utilización de las herramientas seleccionadas.

En este punto se abordará cada uno de los elementos de la pila del producto y su respectivo sprint backlog (pila de tareas), así como la documentación del proceso de desarrollo inmerso en cada actividad del Sprint. Se iniciará el proceso con la pila de tareas priorizadas, detalladas en la Tabla 3 (pila del producto inicial). De acuerdo a la metodología, por cada uno de los elementos de la pila de producto, se debe tener una estimación de la importancia y del esfuerzo inicial. De la estimación de la importancia deberá encargarse el dueño del producto (product owner), quien a su vez coloca una valoración a cada uno de dichos elementos de acuerdo al peso que observa dentro del proyecto.

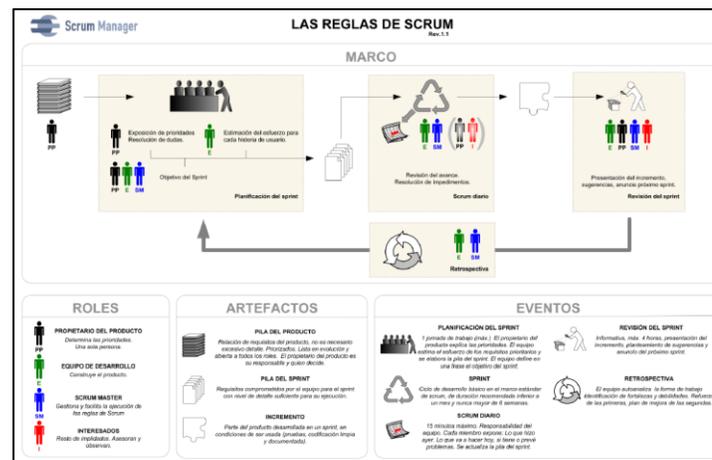


Figura 46. Reglas de SCRUM

Schwaber y Sutherland (2013) en su libro la guía de Scrum Indica “Scrum se basa en la teoría de control de procesos empírica o empirismo. El empirismo asegura que el conocimiento procede de la experiencia y de tomar decisiones basándose en lo que se conoce. Scrum emplea un enfoque iterativo e incremental para optimizar la predictibilidad y el control del riesgo. Tres pilares soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación”

Por otro lado, para la estimación del esfuerzo inicial, se debe encargar el Scrum manager, de acuerdo a las reuniones con el grupo de desarrolladores (Scrum Team), para delimitar el esfuerzo requerido. Así mismo en la Tabla 1, se puede apreciar la asignación del esfuerzo inicial requerido para cada uno de los 15 requerimientos señalados. Esto es especialmente importante, para poder tener una idea clara de cuáles actividades serán las que más recursos humanos demandarán.

Schwaber y Sutherland (2013) p.6 indica “El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. Los Equipos Scrum son autos organizados y multifuncionales. Los equipos auto organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad. Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación”. Las entregas incrementales de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

Una vez definida la pila del producto con sus respectivas valoraciones, se debe especificar por cada uno de los elementos (requerimientos), las correspondientes acciones o tareas a realizar o lo que en Scrum se denomina pila de tareas.

SISTEMA FONDO NOTARIAL (Diagramas UML)

ADMINISTRADOR DEL SISTEMA

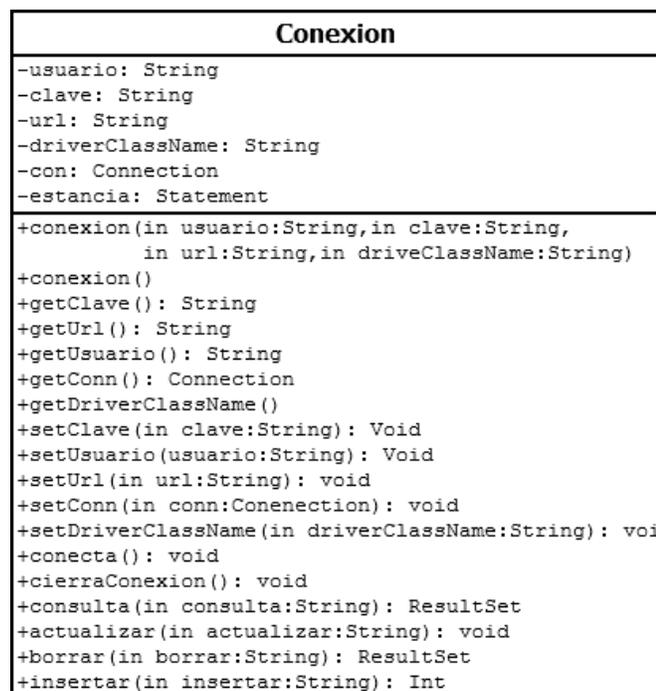


Figura 47. Diagrama de Clases. Clase Conexión

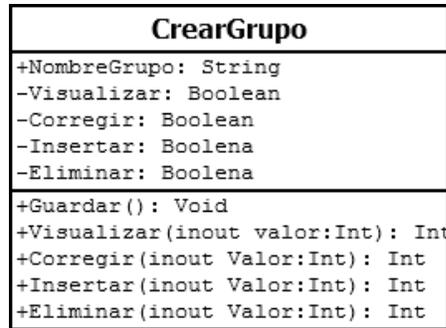


Figura 48. Diagrama de Clase. Crear Grupos de Ingreso

Diagramas de Casos de Uso.

Grafico 01. Ingreso al Sistema de FONDO NOTARIAL

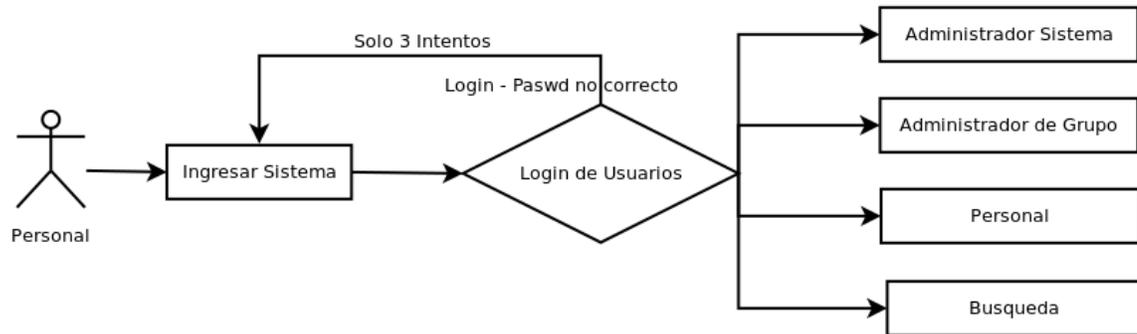


Figura 49. Sistema de Ingreso Fondo Notarial

Grafico 02. Administrador del Sistema - SISTEMA
FONDO NOTARIAL

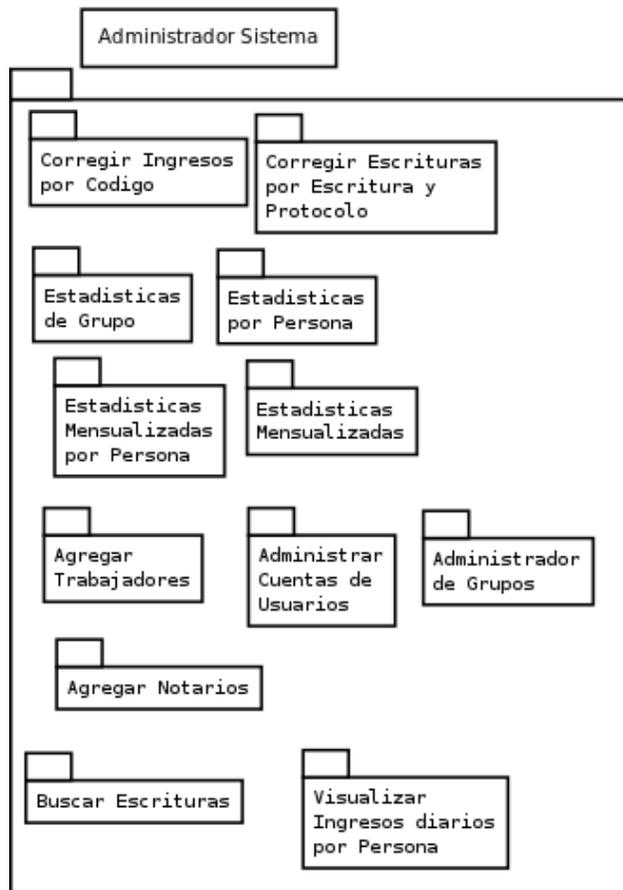


Figura 50. Diagrama de Paquetes. Administrador del Sistema

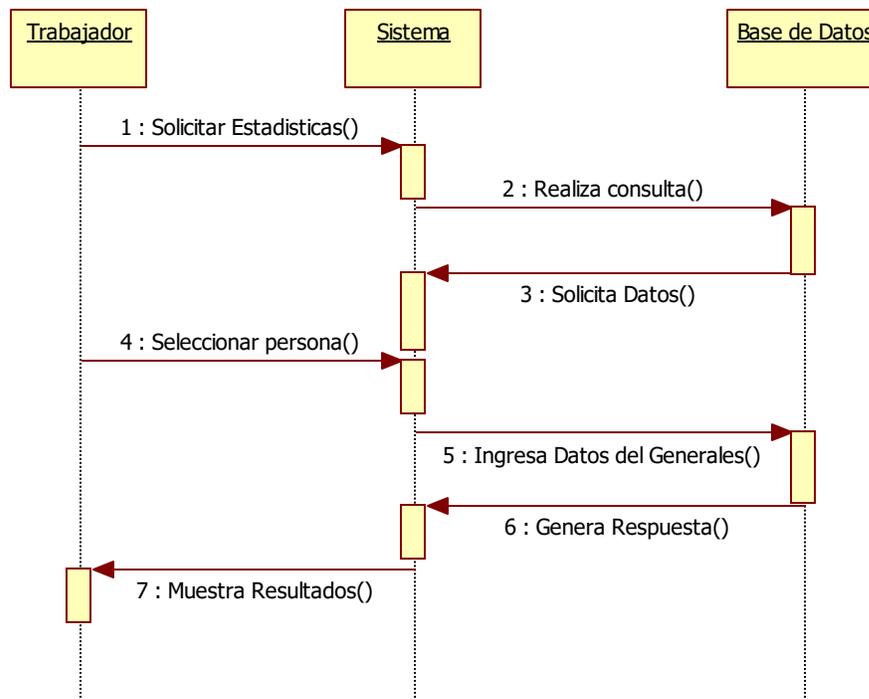


Figura 51. Diagramas de Secuencia. Generar Estadísticas Mensuales

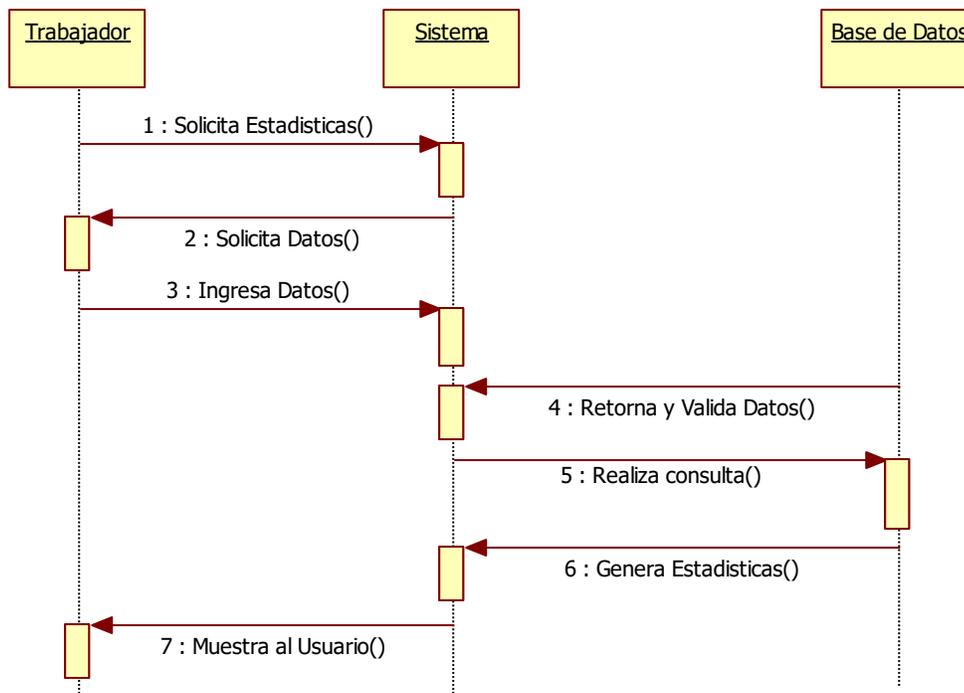


Figura 52. Diagramas de Secuencia. Generar Estadísticas Mensualizadas por Persona

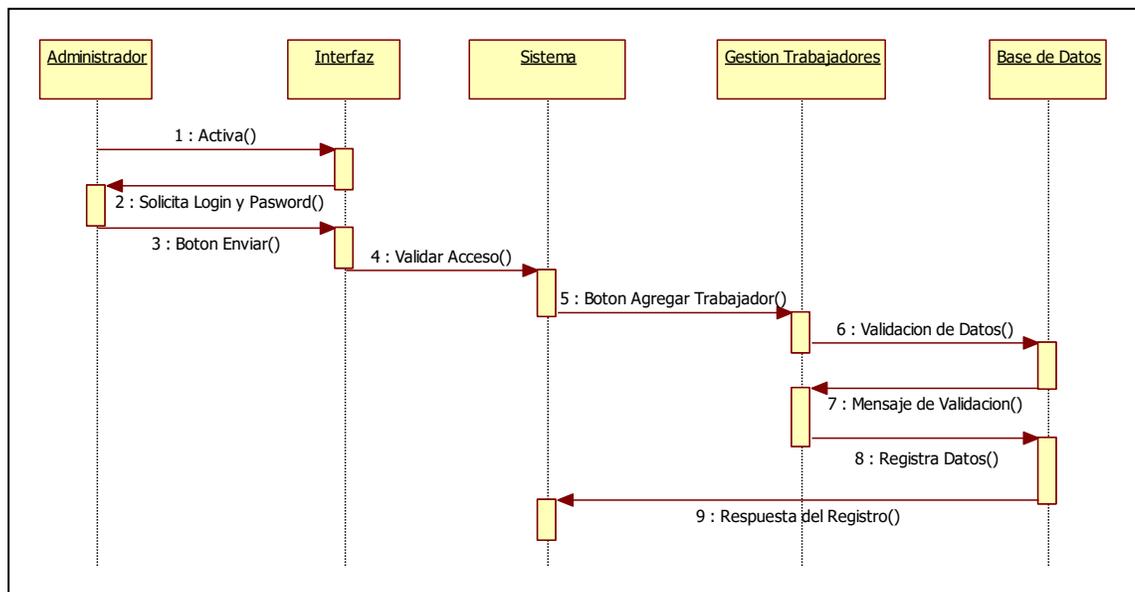


Figura 53. Diagrama. Estadísticas Mensualizadas por persona

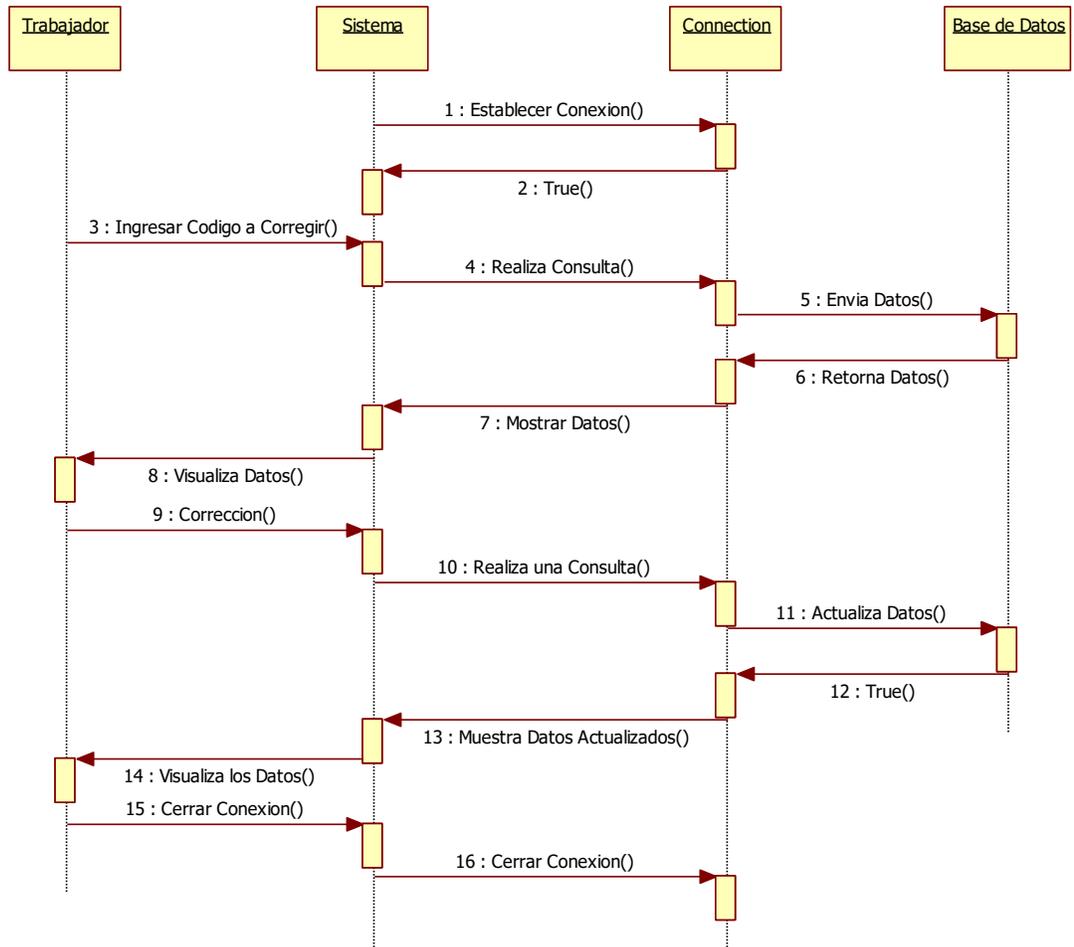


Figura 54. Diagramas de Secuencia. Corrección de Ingresos a Base de Datos por Código

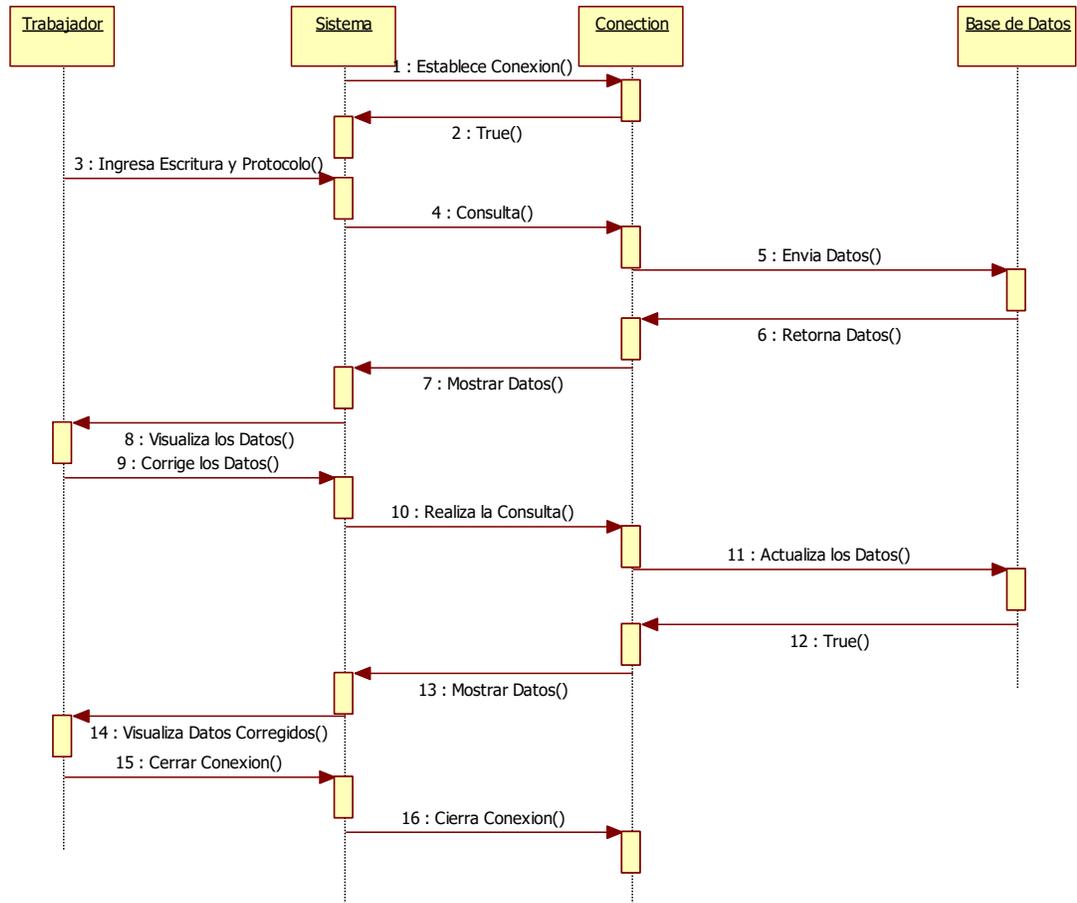


Figura 55. Diagramas de Secuencia. Corrección de Ingreso a Base de Datos por Escrituras y Protocolo

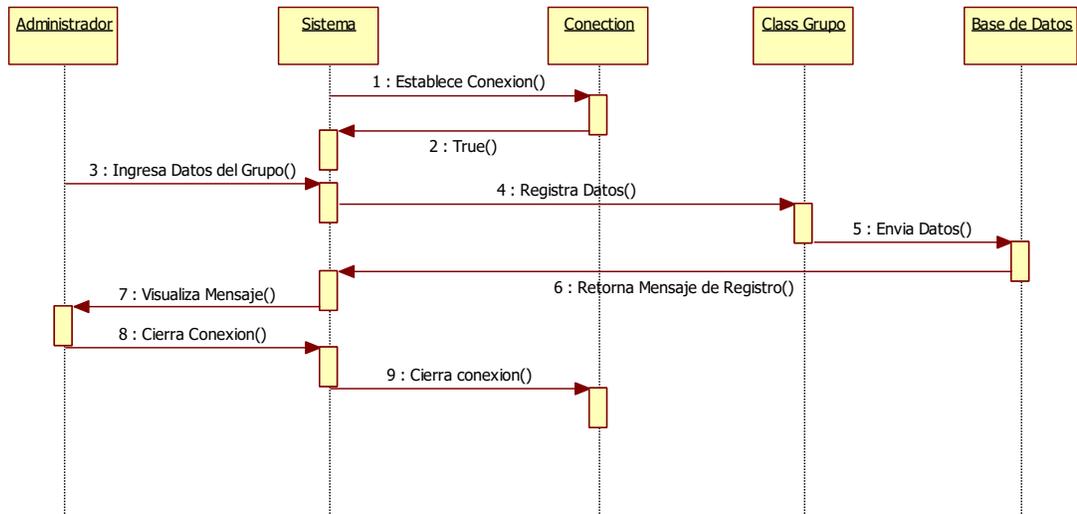


Figura 56. Diagramas de Secuencia. Administración de Grupos

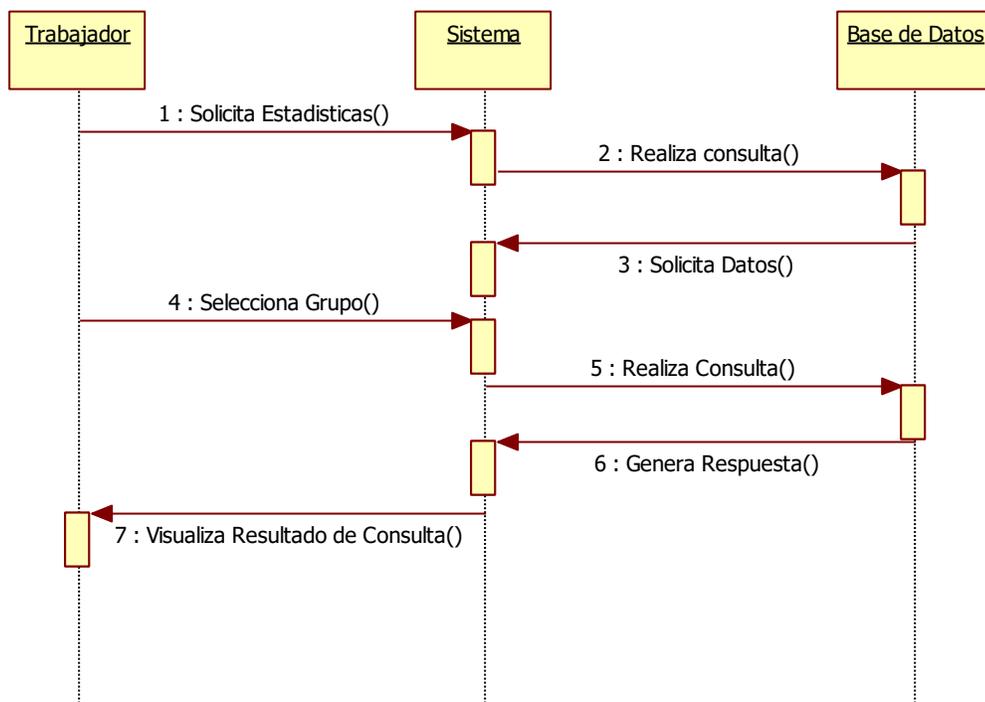


Figura 57. Diagramas de Secuencia. Estadísticas de Grupos

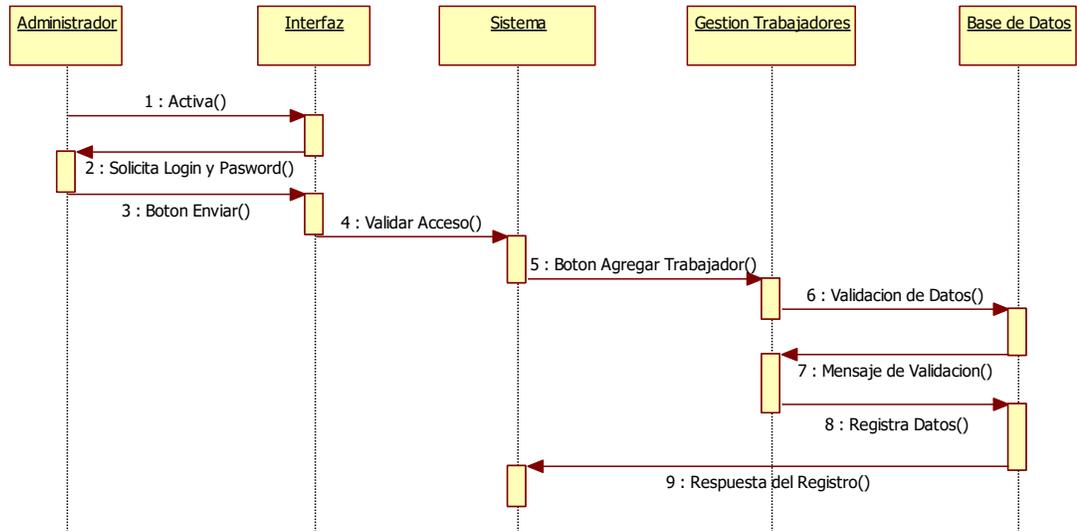


Figura 58. Diagramas de Secuencia. Agregar Trabajador

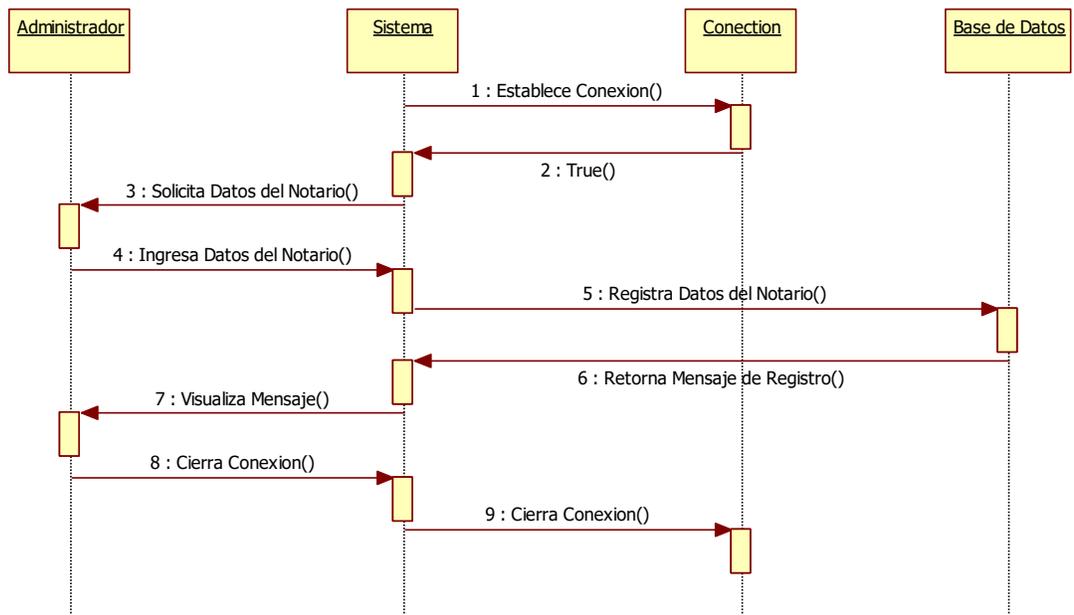


Figura 59. Diagrama de Secuencia. Agregar Notario

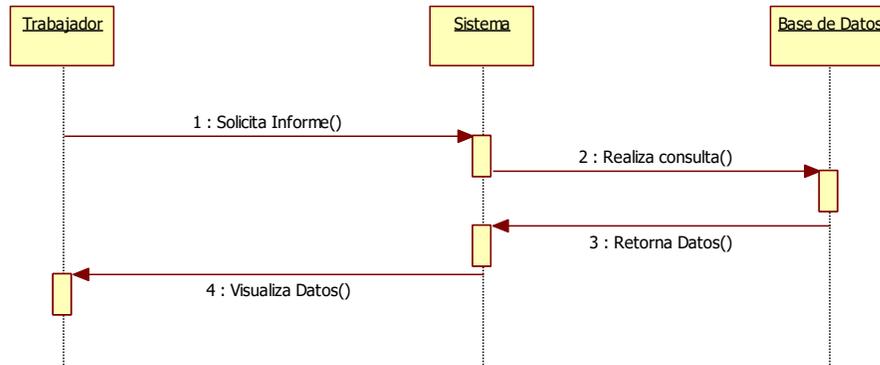


Figura 60. Diagramas de Secuencia. Visualizar Ingresos Diarios por Persona

ADMINISTRADOR DE GRUPO

Grafico 03. Administrador de Grupo - SISTEMA FONDO NOTARIAL

Viene del Grafico 01.

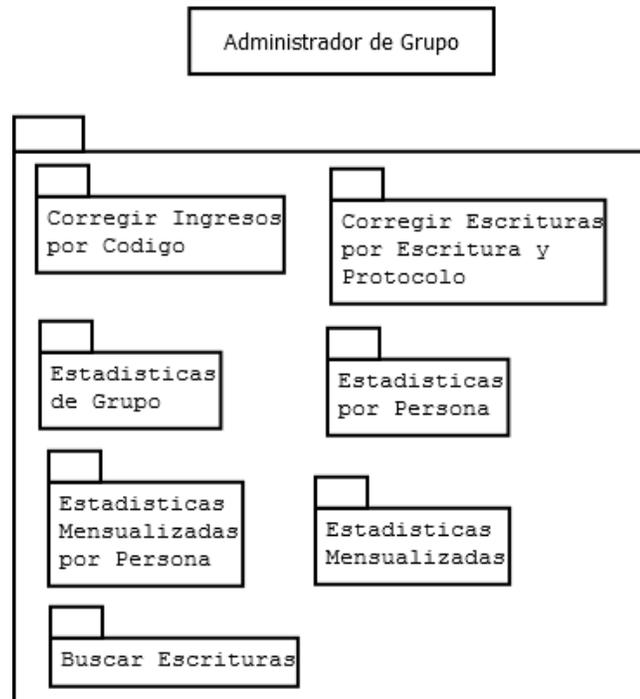


Figura 61. Módulos de Administración de Grupos

Diagramas de Secuencia. Corrección de datos

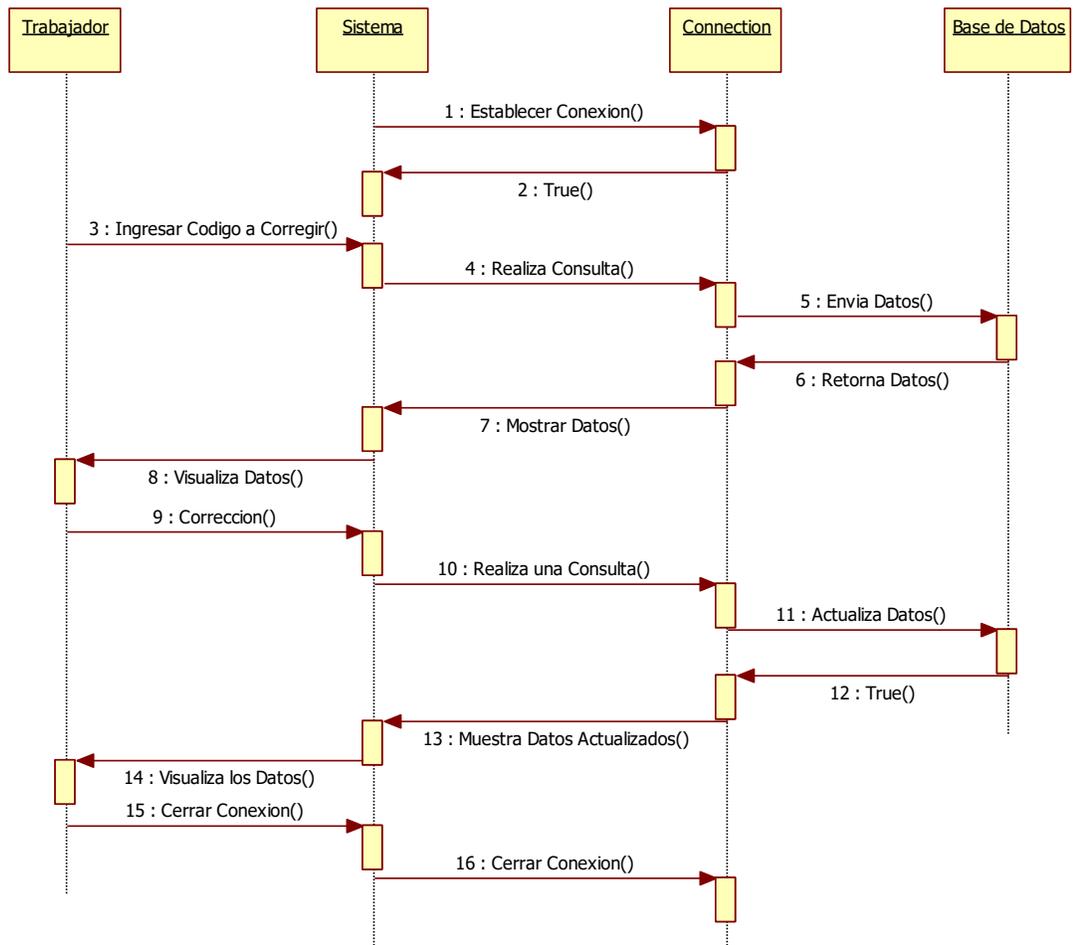


Figura 62. Diagramas de Secuencia. Corrección de Ingresos a Base de Datos por Código

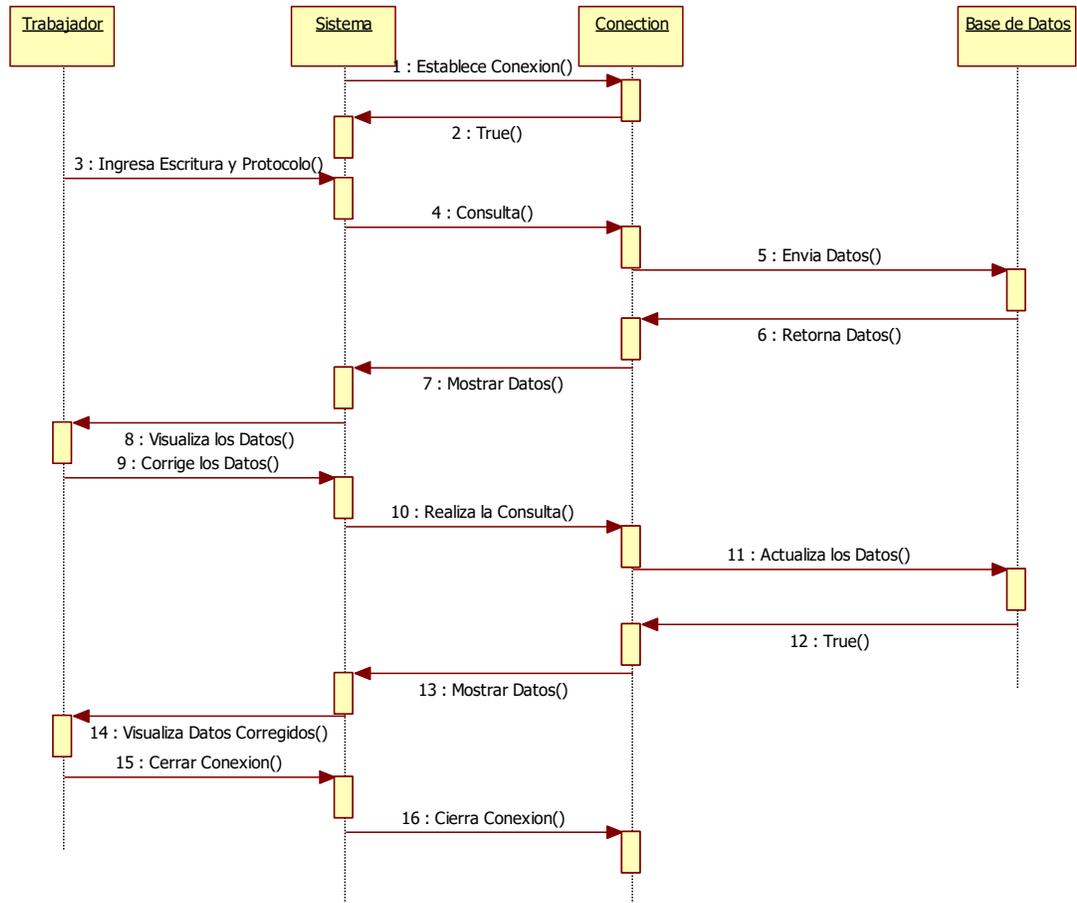


Figura 63. Diagramas de Secuencia. Corrección de Ingreso a Base de Datos por Escrituras y Protocolo

Grafico 04. Personal - SISTEMA FONDO NOTARIAL
Viene del Grafico 01.

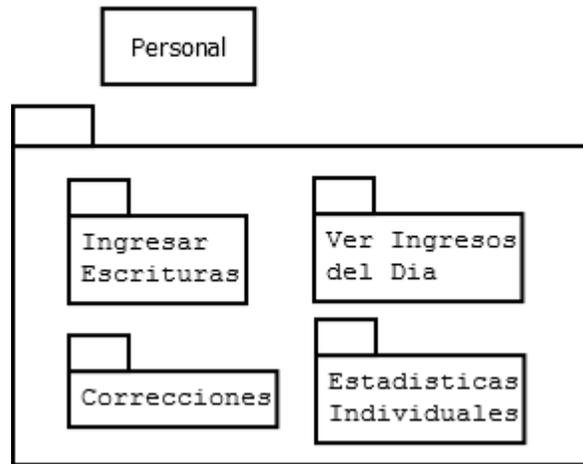


Figura 64. Diagramas de Secuencia. Estadísticas de Grupos

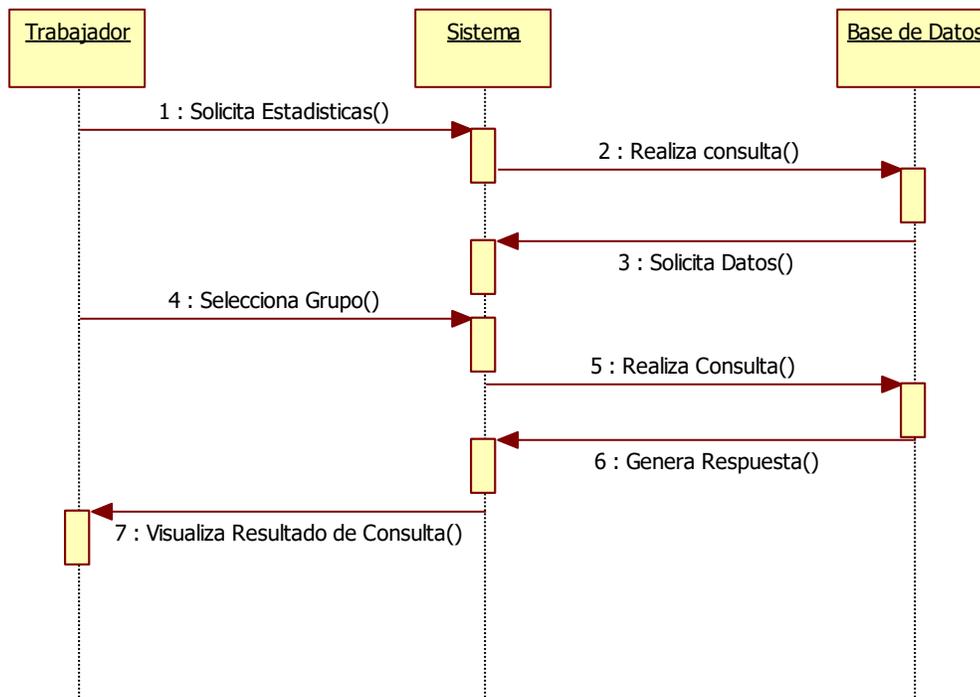


Figura 65. Diagrama de secuencias, Consulta de Escrituras

3.5.11 Estándares de programación

Los estándares proveen un orden y una metodología para la programación, los mismos que son utilizados para evitar cometer errores. Particularmente, se han seguido algunos parámetros como las variables es minúscula, nombres en Camel Case, constantes solo en mayúsculas, la indentación del código del programa, entre otras. El seguir estas reglas, no solo favorece la fácil lectura de un código fuente, además permite que el código fuente sea legible y entendible para otros programadores.

La codificación del sistema también con lleva a una estandarización para definir a las clases, funciones, esto hace los programas más entendibles haciéndolos más fáciles de leer, además de dar información sobre la función de un identificador, por ejemplo cuando es una clase, constante, paquete, puede ser útil para entender el código fuente del sistema. La Tabla 3 muestra la estandarización de código fuente que se usará en el desarrollo del proyecto.

Tabla 13

Estandarización de código

Descripción	Reglas para nombres
Paquetes	Los subsecuentes elementos del nombre del paquete variarán de acuerdo a las conveniencias de nombres internas de cada organización.
Clases	Los nombres de las clases deben ser sustantivos, cuando son compuestos tendrán la primera letra de cada palabra que lo forma en mayúsculas. Intentar mantener los nombres de las clases simples y descriptivas. Usar palabras completas, evitar acrónimos y abreviaturas
Interfaces	Los nombres de las interfaces siguen la misma regla que las clases. Los métodos deben ser verbos, cuando son compuestos tendrán la primera letra en minúscula, y la primera letra de las siguientes palabras que lo
Métodos	forma en mayúscula.
Variables	Excepto las constantes, todas las instancias y variables de clase o método empezarán con minúscula.
	Los nombres de las variables deben ser cortos pero con significado.
Constantes	Los nombres de las variables declaradas como constantes deben ir totalmente en mayúsculas separando las palabras con un subguión ("_").

Tabla 14

Pila de producto sprint 1- ventana del administrador

PILA DE PRODUCTO SPRINT 1

ID	Nombre de la historia	Importancia	Estimación	Sprint	Como probarlo	Notas
1	Ingreso al sistema	15	4	1	Entrar a la aplicación, ingresar usuario y contraseña, si el usuario es correcto ingresa al menú inicial caso contrario se presentara un mensaje de error.	El usuario debe estar registrado como administrador en la base de datos.
2	Creación de usuario en el sistema	14	4	1	Entrar a la aplicación, ir a seguridad, ir a usuario, ingresar los datos solicitados, realizar el ingreso de la información, verificar en el tracking que se haya creado en nuevo usuario.	Para la creación de usuario previamente deben estar creados los respectivos perfiles de acuerdo a los roles que maneja la institución.

SPRINT BACKLOG ITERACIÓN 1

En este periodo se enumeran las tareas y sub-tareas comprendidas dentro de la Iteración 1 que engloba la historia de usuario ingreso al sistema y creación de usuario en el sistema. En la Tabla 6 se puntualiza la pila de tareas para la Iteración 1.

A la hora de organizar, repartir y trabajar sobre las historias de usuario de la pila de tareas, se suele plasmar una división de cada historia en diferentes acciones como se ve en la Tabla 6. Es substancial observar la discrepancia entre tarea e historia, mientras que una historia es un entregable y es responsabilidad del dueño del producto, una tarea no es entregable y la responsabilidad es del equipo de desarrollo. Una historia suele dividirse en diferentes tareas.

REVISIÓN DEL SPRINT

Se presentan las listas de tareas realizadas y pendientes de cada desarrollador del sistema para seguir midiendo su avance para llegar a la fecha límite planteada.

TAREAS COMPLETADAS

Se en listan las tareas que los desarrolladores ha terminado hasta el momento de la revisión del sprint en la reunión.

Tabla 15

Sprint 1 – Módulo de Ingresos al sistema y Seguridad

Id	Tarea	Delegado	Estado	Hora	Días	Elemento del Product Backlog								
						Sprint	Inicio	Duración (días)	1 al 4 Abril 2014	7 al 11 abril 2014	14 al 19 Abril 2014	21 al 25 Abril 2018		
						1	1 abril 2014	20	Ingreso al Sistema					
1	Instalación y configuración del Servidor	Edgar	Completado						X					
1	Diseño de interfaces gráficas de usuario	Edgar	Completado						X					
1	Elaboración del modelado lógico y físico de la base de datos	Edgar	Completado							X				
2	Codificación de los módulos en PHP	Edgar	Completado							X				
2	Elaboración de consultas	Edgar	Completado							X				
2	Codificación de Interface y Sesión (CRUD) de la página de logueo	Edgar	Completado							X				
2	Codificación de la lógica del negocio para la página de logueo	Edgar	Completado							X				
3	Configuración de las seguridad	Edgar	Completado								X			
3	Elaboración de consultas usando MVC	Edgar	Completado								X			
3	Codificación de Interface y sesión para la creación de usuarios	Edgar	Completado								X			
4	Codificación de la lógica del negocio para la creación de usuarios	Edgar	Completado											X
4	Mejora de la seguridad para la creación de usuarios	Edgar	Completado											X

Tabla 16

Sprint 2 – Módulo de Ingresos de Escrituras con MVC

Sprint	Inicio	Duración (días)	Elemento del Product Backlog		Hora	Días	28 Abr al 2 mayo	5 may al 9 2014	12 may al 16 may	19 may al 23 mayo	26 myo al 30 mayo
2	28 abril al 30 Mayo 2014	25	Sistema de Ingreso de Escrituras MVC								
1	Análisis de ingreso de Escrituras UML	Edgar	Completado			X					
1	Codificación del módulo de ingresos por sistema	Edgar	Completado			X	X				
1	Codificación y pruebas para el ingreso de escrituras	Edgar	Completado				X				
1	Pruebas y corrección de errores	Edgar	Completado				X				
1	Codificación del módulo de ingreso para escrituras con otorgantes y favorecidos	Edgar	Completado				X				
2	Control de calidad del ingreso	Edgar	Completado				X				
2	Codificación del ingreso con datos de fechas , nombre de bien y otros	Edgar	Completado				X				
2	Diseño y codificación de ventanas para separación de datos de ingresos con códigos de sesión	Edgar	Completado					X			
2	Uso de SESSION para guardar el código del persona que usa el sistema	Edgar	Completado					X			
2	Separación de las fases de ingreso por tipo de usuario: otorgantes en una ventana; Favorecidos en otra ventana	Edgar	Completado					X	X		X
3	Codificación del módulo de ingreso de los datos de la escritura y guardado en la base de datos.	Edgar	Completado						X		X

Tabla 17

Sprint 3 – Módulo de Reportes diarios de ingreso de escrituras

Sprint	Inicio	Duración (días)	Elemento del Product Backlog		Horas	Días	2 – jun al 6-jun	9 – jun al 13-jun	16-jun al 20-jun	23-jun al 27-jun
			Id	Tarea						
3	2-Jun-2014 al 27-jun-2014		Elaboración de Reportes diarios							
1	Análisis del modelo de negocio para la obtención de requerimientos	Edgar	Completado		8	X				
1	Análisis y diseño del módulo de reportes	Edgar	Completado		8	X				
	Codificación del sistema de reportes	Edgar	Completado		8	X	X			
1	Maquetación del sistema en HTML	Edgar	Completado		8		X			
1	Diseño de las consultas en SQL	Edgar	Completado		8		X	X	X	
2	Creación de Vistas y procedimientos almacenados	Edgar	Completado		8			X		
2	Acondicionamiento de JavaScript en los reportes	Edgar	Completado		8			X		
2	Pruebas de funcionalidad	Edgar	Completado		8			X		
2	Permisos para usuarios	Edgar	Completado		8			X		
2	Creación de Vistas para trabajadores	Edgar	Completado		8			X	X	
3	Creación de reportes para Jefe/administrador	Edgar	Completado		8			X	X	

Cada aspecto del software desarrollado cumple exactamente con las mismas métricas y desarrollo, haciendo uso de estos modelos de Sprints, en base a la metodología SCRUM.

3.5.12 Product backlog final

Al cumplir el proceso de desarrollo se obtiene el listado de tareas basadas en las historias de usuario pero ya delimitadas, asignadas y terminadas

Tabla 18

Finalización de procesos de desarrollo

Pila de Producto Final					
ID	Nombre	Importancia	Estimación	Estado	Sprint
1	Ingreso al sistema	15	4	Completado	1
2	Creación de usuario en el sistema	14	4	Completado	1
3	Ingreso de información del sistema	13	4	Completado	2
4	Ingreso de parámetros generales del trabajador en el sistema	12	4	Completado	2
5	Ingreso de parámetros generales de libros en el sistema	11	3	Completado	2
6	Ingreso de parámetros en la base de datos	10	3	Completado	2
7	Administración Personal	9	4	Completado	3
8	Inscripción Personal	8	4	Completado	3
10	Administración Personal	6	3	Completado	4
12	Reporte de ingresos por personal	4	4	Completado	5
13	Generar Reportes Diarios Mensuales	3	3	Completado	5
15	Reportes y Consultas	1	4	Completado	6

3.5.13 Modelo de datos

Para este proyecto se utilizó las técnicas de particionamiento vertical y horizontal, así como también, la normalización de bases de datos para el análisis lógico de la misma; una vez finalizado este proceso, se procedió a crear la base de datos física, haciendo uso del lenguaje SQL, y las pruebas de registro (Inserción, actualización, borrado y vistas), las cuales han sido necesarias para nuestro software.

El modelo lógico se basa más en la teoría relacional y aplica los principios de normalización apropiados a los datos. El modelo físico, por otro lado, se refiere a la aplicación real, ya que los datos se almacenan en un RDBMS. Basado en el diseño lógico, el diseño físico trae la estructura de datos al mundo real de la implementación. (Oppel y Sheldon, 2010 p.33)

Oppel y Sheldon (2010) indica que SQL respalda tres tipos de Tablas: Tablas base, Tablas derivadas y Tablas vistas. La mayoría de las Tablas base son objetos de esquema que tienen los datos de SQL. Las Tablas derivadas son los resultados que se observan cuando se solicitan (consultan) datos de una base de datos.

La clave primaria para una Tabla se escoge por el diseñador de la base de datos de claves de candidato disponibles. Una clave de candidato es un conjunto de una o más columnas que identifican de forma exclusiva a cada fila.

3.5.14 Diseño de la base de datos

Para la construcción de la base de datos se emplea el modelo entidad – relación. Oppel y Sheldon (2010) escribe “La parte central de los principios del modelo relacional es el concepto de normalización, una técnica para producir un conjunto de relaciones que poseen un conjunto de ciertas propiedades que minimizan los datos redundantes y preservan la integridad de los datos almacenados tal como se mantienen (añadidos, actualizados y eliminados). El proceso fue desarrollado por E. F. Codd en 1972, y el nombre es un chiste político debido a que el presidente Nixon estaba “normalizando” relaciones con China en ese momento. Codd imaginó que si las relaciones con un país pueden normalizarse, entonces seguramente podría normalizar las relaciones de la base de datos. La normalización se define por un conjunto de normas, que se conocen como formas normales, que proporcionan una directriz específica de cómo los datos son organizados para evitar anomalías que den lugar a inconsistencias y pérdida de los datos tal como se mantienen almacenados en la base de datos. Cuando Codd presentó por primera vez la normalización, incluía tres formas normales. A pesar de que formas normales adicionales se han agregado desde entonces, las tres primeras cubren la mayoría de las situaciones que se encontrarán en las dos bases de datos personales y empresariales, y ya que la intención principal es presentar el proceso de normalización, sólo se analizarán esas tres formas.”

La base de datos se construye en base a los requerimientos plasmados en las historias de usuarios con el fin de obtener la visión que tiene los usuarios sobre la manipulación de los datos a través del sistema de Fondo Notarial.

El siguiente diagrama representa las entidades requeridas para la satisfacción de las necesidades del cliente y las relaciones entre ellas.

Tabla 19

Tabla Usuarios de la Base de Datos

Campo	Tipo	Descripción
cod_usu	INT	Código único del usuario
nom_usu	VARCHAR(80)	Nombre de la persona
pat_usu	VARCHAR(100)	Apellido paterno de la persona
mat_usu	VARCHAR(100)	Apellido materno de la persona
log_usu	VARCHAR(30)	Login del usuario
psw_usu	VARCHAR(40)	Clave del usuario
niv_usu	VARCHAR(1)	El Nivel del usuario *
chk_usu	VARCHAR(1)	Si el usuario está Activo o No
dir_usu	VARCHAR(150)	Dirección del usuario
tlf_usu	VARCHAR(15)	Teléfono del Usuario

* Existen: 1 = Administrador del Sistema, 2=Personal (Acceso restringido), 3=Búsqueda (Solo búsqueda y acceso al buscador)

Tabla 20

Tabla Proyecto de la Base de Datos

Campo	Tipo	Descripción
Proy_id	INT	Código único de la Tabla proyecto
Proy_nombre	VARCHAR(100)	Nombre del proyecto
Not_id	INT	Código del Notario relacionada con la Tabla Notarios
Num_protocolo	INT	Numero de protocolo
Folio_inicial	VARCHAR(10)	Folio inicial
Folio_final	VARCHAR(10)	Folio final
Escritura_inicial	INT	Escritura Inicial
Escritura_final	INT	Escritura final
observaciones	VARCHAR(500)	
Cod_usu	INT	Código de usuario
Fecha_creacion	Datetime	
Fecha_final	Datetime	
estado	TINYINT	Valor booleano TRUE/FALSE

Tabla 21

Tabla de Notarios de la Base de Datos

Campo	Tipo	Descripción
cod_not	INT	Código único de la Tabla Notarios
nom_not	VARCHAR(60)	Nombre del Notario
pat_not	VARCHAR(60)	Apellido Paterno del Notario
mat_not	VARCHAR(60)	Apellido Materno del Notario
fec_ini	CHAR(4)	Año que inició el Notariado
fec_fin	CHAR(4)	Año que finalizó el Notariado
provincia	VARCHAR(30)	Nombre de la provincia donde ejerció como Notario
obs	VARCHAR(255)	Observaciones

Tabla 22

Tabla escrituras de la Base de Datos

Campo	Tipo	Descripción
cod_sct	int(10)	Código de la Escritura
cod_not	int(10)	Código del Notario
num_sct	int(10)	Numero de Escritura
cod_dst	int(10)	Código del Distrito al que pertenece
fec_doc	date	Fecha de creación del documento
cod_sub	int(10)	Código de sub serie
nom_bie	varchar(255)	Nombre del bien
can_fol	int(10)	Cantidad de Folios
cod_pro	int(10)	Número del Libro (PROTOCOLO)
obs_sct	varchar(255)	Observación de la escritura
num_fol	varchar(255)	Número de Folio
cod_usu	int(10)	Código del Usuario
hra_ing	timestamp	Fecha y hora de ingreso

Tabla 23

Tabla otorgantes – escritor de la Base de Datos

Campo	Tipo	Descripción
cod_rel	varchar(6)	Código del registro Tabla otorgantes
cod_sct	varchar(6)	Código de la Escritura
cod_inv	varchar(6)	Código de la Escritura
flg_per	varchar(1)	Código del Personal

Tabla 24

Tabla Favorecidos – escritavor de la base de datos

Campo	Tipo	Descripción
cod_rel	varchar(6)	Código del registro Tabla favorecidos
cod_sct	varchar(6)	Código de la Escritura
cod_inv	varchar(6)	Código de la Escritura
flg_per	varchar(1)	Código del Personal

Tabla 25

Tabla de personas involucradas en las escrituras

Campo	Tipo	Descripción
Cod_inv	int(10)	Código único para una persona
Pat_inv	varchar(255)	Apellido paterno
Mat_inv	varchar(255)	Apellido Materno
Nom_inv	varchar(255)	Nombres del involucrado

Tabla 26

Tabla Involucrados Jurídicos

Campo	Tipo	Descripción
Cod_inv	int(10)	Código único -personas jurídicas
Raz_inv	varchar(255)	Nombre o Razón social
otros_juri	varchar(255)	Otros involucrados

Tabla 27

Tabla Serie

Campo	Tipo	Descripción
cod_sre	int(10)	Código auto incrementable
cod_sec	int(10)	Código de Sección
des_sre	varchar(100)	Descripción de la Serie

Tabla 28

Tabla Sub series

Campo	Tipo	Descripción
cod_sub	int(10)	Código auto incrementable
cod_sre	int(10)	Código de la Serie
des_sub	varchar(300)	Descripción de la Sub serie

Una vez definido la estructura básica de las Tablas, es necesario realizar un análisis previo, y se efectuaron tareas de normalización de base de datos, quedando las Tablas listas para el próximo proceso, que es la creación física de la base de datos.

Haciendo uso de MySQLWorkbench, proporcionado por MySQL es que presentamos a ustedes los modelos de las Tablas haciendo uso del software modelador, quedando la Tabla final de la siguiente forma.

3.5.16 Diseño Físico de la Base de Datos

Para el diseño físico de la Base de Datos se utilizó MySQL, y también se ha utilizado MySQLWorkbench, para la creación física, sin dejar de utilizar phpmyadmin.

La creación física de la base de datos, se realizó usando los códigos SQL que es el lenguaje universal para Bases de Datos, en el cual solo colocaremos una muestra del código en esta parte del documento y la totalidad del código se encontrará en los anexos.

Primeramente debemos iniciar el software de MySQL que para el caso de Windows se utilizó la versión de XAMPP 5.6, que a parte que tiene MySQL contiene el software Apache , PHP y Perl, PhpMyadmin, entre otros.

En las siguientes imágenes, mostramos los datos y registros ingresados por medio de la aplicación, los cuales han servido para que el ingreso de este importante fondo documental, sea ingresado a una base de datos en Tablas, como se puede apreciar.

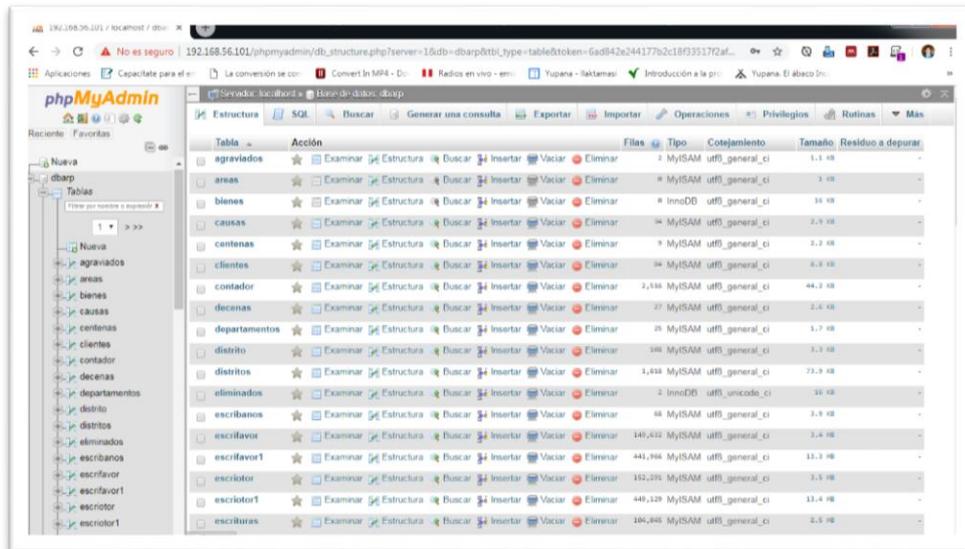


Figura 67. En la Figura se muestra todas las Tablas de la Base de Datos “dbarp”



Figura 68. La figura muestra una vista en MySQL para el listado de Notarios

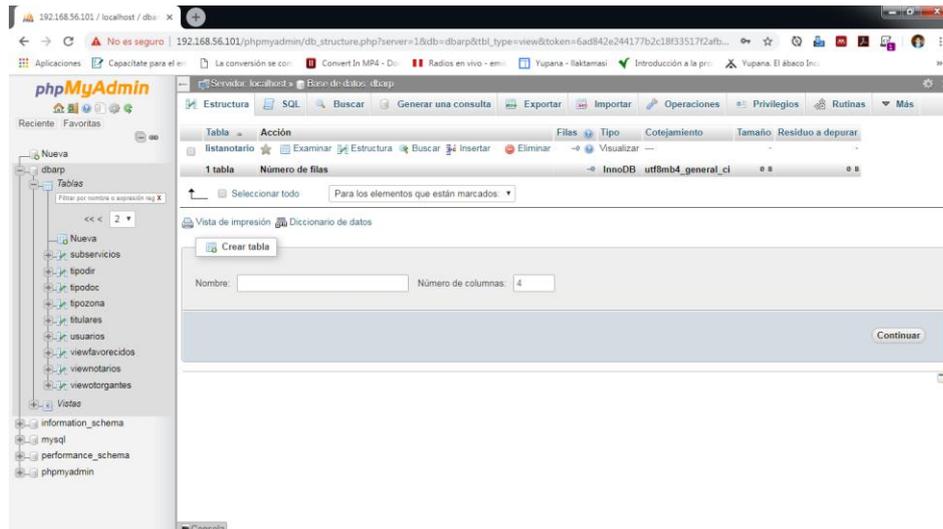


Figura 69. La figura muestra las otras vistas utilizadas en MySQL

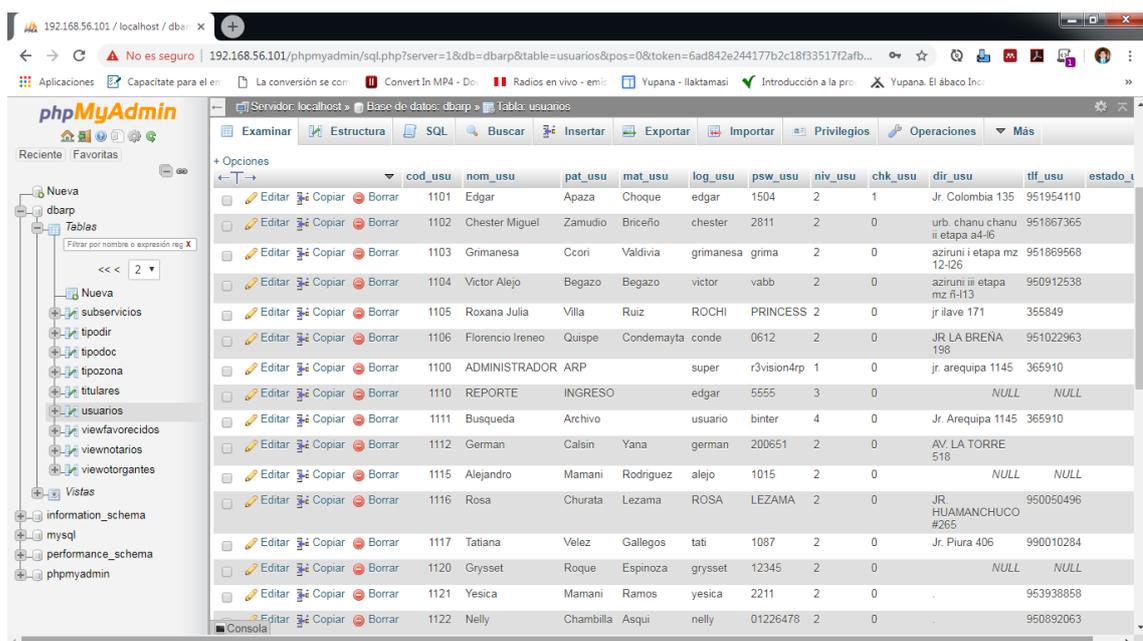


Figura 70. La Figura muestra la lista de los usuarios registrados

cod_sct	cod_not	num_sct	cod_dst	fec_doc	cod_sub	nom_bie	can_fol	cod_pro	obs_sct	num_fol	cod_usu	hra_ing	proy_id
1	112	1	211101	1998-08-08	48	FUNDO RUSTICO	3	1934	ESCALERANI-ANCCO-MOCCO	01	1104	2010-08-01 14:23:09	0
2	112	1015	211101	1998-09-30	48	SECTOR LO LO COMUNIDAD YOCARA	1	1937		1501	1105	2010-08-02 08:08:53	0
3	112	269	211101	1998-07-18	49	SOCIEDAD COMERCIAL INHUMACIONES S.R.L	3	1935		501	1103	2010-08-02 08:08:07	0
4	112	300	211101	1998-07-18	48	URB. VILLA SAN JUAN MZ A, LOTE 12	2	1935		503v	1103	2010-08-02 08:08:37	0
5	112	2	211101	1998-08-08	48		2	1934		04	1104	2010-08-02 08:08:24	0
6	112	301	211101	1998-07-18	49	ASOCIACION DE DETECTIVES PRIVADOS DEL PERU Y DEL M...	7	1935		504v	1103	2010-08-02 08:08:24	0
7	112	3	211101	1998-08-08	48	LOTE DE TERRENO URB. MZ "D" DE LA URBANIZACION ...	3	1934	ERROR EN EL NOMBRE DEL FABRICATION EN LA SUMILLA AP...	06	1104	2010-08-02 08:08:51	0
8	112	302	211101	1998-07-20	48	URB. PROLONGACION SANTA CELEDONIA MZ-C4, LOTE 1	2	1935		510VTA	1103	2011-07-20 10:54:48	0
9	112	303	211101	1998-07-20	48	URBANIZAC. MUNICIPAL TAPARACHI MZ- 9-S, LOTES 4-S...	2	1935		511v	1103	2010-08-02 09:08:55	0
10	112	4	211101	1998-08-08	48	LOTE UBICADO EN LA MZ "D" DE LA URBANIZACION JESUS ...	2	1934	LOS FOLIOS VUELTAS ESTAN EN BLANCO CON LA ESCRIPCI...	09	1104	2010-08-02 09:08:08	0
11	112	5	211101	1998-08-09	88		2	1934		011	1104	2010-08-02 09:08:53	0
12	112	1010	211101	1998-08-30	48	JR. MARIANO NUÑEZ 899 JULIACA	2	1937		1502	1105	2010-08-02 09:08:55	0
13	112	304	211101	1998-07-20	41	JR BENIGNO BALLON 1015-1017	2	1935		512v	1103	2010-08-02 09:08:02	0
14	112	6	211101	1998-08-10	48	INMUEBLE URBANO UBIC LA RINCONADA 204 ESTAR MCE L...	2	1934		013	1104	2010-08-02 09:08:27	0
15	112	305	211101	1998-07-16	48	JR LAMBAYEQUE 812-818	2	1935		512v	1103	2010-08-02 09:08:34	0
16	112	842	211101	1998-08-29	48	LOTE EN URB. SAN JULIAN	2	1936	EL PRIMER FOLIO ESTA ROTO	1001	1102	2010-08-02 09:08:28	0
17	112	843	211101	1998-08-29	48	LOTES EN URB. SANTA OLIMPIA	2	1936		1002v	1102	2010-08-02 09:08:38	0
18	112	1018	211101	1998-09-30	156	JR. APURIMAC NRO.	2	1937		1504	1105	2010-08-02 09:08:05	0

Figura 71. Muestra de la Tabla escrituras

Cod_not	Pat_not	Mat_not	Nom_not	Lugar	Fec_ini	Fec_fin
1	GARNICA	ROSADO	JULIO	PUNO	11/19/58	03/30/99
2	CUBARUBIAS	MULL	ANDRES	PUNO	23/02	12/31/00
3	DIAZ	PAREJA	DIEGO	PUNO	25/21	12/31/99
4	NIETO	MULL	JOSEPH	PUNO	25/23	12/31/99
5	CORONEL	GARCIA	VICENTE	PUNO	27/30	12/31/07
6	ZAMORA	MULL	JOSE MARIA	PUNO	12/16	01/01/01
7	CUENTAS	MULL	MANUEL	PUNO	14/86	12/31/47
8	ASCENCIO	MULL	MANUEL	PUNO	17/44	12/31/00
9	PONCE	MULL	MANUEL JORGE	PUNO	18/42	12/31/82
10	CACERES	MULL	MANUEL	PUNO	07/27/52	32/70
11	CORRALES	MULL	NICOLAS	PUNO	23/43	12/31/98
12	CANO	MULL	ALEJANDRO	PUNO	24/53	12/31/88
13	MANRIQUE	MULL	CLEMENTE	PUNO	24/87	02/01/98
14	JIMENEZ	MULL	MANUELE	PUNO	28/29	12/31/80
15		MULL	JEAN S. MARTIN	PUNO	01/25/60	03/21/67
16	TORANZOS	MULL	CARLOS	PUNO	04/17/60	08/25/00
17	GONZALES	MULL	SIMON	PUNO	04/17/66	12/31/18
18	DEZA	MULL	RICARDO	PUNO	30/87	05/21/19
19	ARAMAYO	MULL	ALFREDO	PUNO	42/48	10/24/38
20	GARNICA	MULL	MARIAHO C.	PUNO	12/28/07	07/17/11
21	GARNICA	MULL	GUILLERMO	PUNO	12/16	08/16/78
22	GARCIA DEARRIAGA	MULL	FRANCISCO	PUNO	28/21	12/31/96

Figura 72. La Figura muestra la lista de los NOTARIOS

The screenshot shows the phpMyAdmin interface for a database named 'dbarp'. The 'fondos' table is selected, and its structure and data are displayed. The table has two columns: 'cod_fon' and 'des_fon'. The data is as follows:

cod_fon	des_fon
1	CORTE SUPERIOR DE JUSTICIA - PUNO
2	MINISTERIO PUBLICO
3	REGISTRO CIVIL
4	INSTITUTO NACIONAL DE ESTADISTICA E INFORMATICA
5	NOTARIAL
6	CORREJIMIENTO
7	INTENDENCIA
8	JUZGADO DE PRIMERA INSTANCIA
9	SOCIEDAD DE BENEFICENCIA PUBLICA
10	TESORERIA FISCAL
11	PREFECTURA
12	MUNICIPALIDAD PROVINCIAL DE PUNO
13	COLECCIONES
14	ESCRIBANIAS
15	ARCHIVO REGIONAL DE PUNO

Figura 73. La Figura muestra la Tabla “fondos documentales”

The screenshot shows the phpMyAdmin interface for a database named 'dbarp'. The 'escritor' table is selected, and its data is displayed. The table has six columns: 'cod_rel', 'cod_sct', 'cod_inv', 'cod_per', and 'cod_inv_ju'. The data is as follows:

cod_rel	cod_sct	cod_inv	cod_per	cod_inv_ju
1	1	1	1104	NULL
2	2	7	1105	NULL
3	3	11	1103	0
4	4	12	1103	0
5	5	5	1104	4
6	6	14	1103	3
7	7	15	1104	0
8	8	17	1103	0
9	9	20	1103	0
10	10	21	1104	0
11	11	25	1104	0
12	12	19	1105	0
13	13	26	1103	0
14	14	29	1104	0
15	15	31	1103	0
16	16	33	1102	0

Figura 74. La Figura muestra el listado de la Tabla “escritor” (Otorgantes Escrituras)

La escritura de la Tabla “escrifavor” es la misma, sin embargo, los códigos de los involucrados cambian

	Cod_inv	Raz_inv	otros_juri
<input type="checkbox"/>	1	INHUMACIONES S.R.L	NULL
<input type="checkbox"/>	2	SOCIEDAD COMERCIAL INHUMACIONES S.R.L	NULL
<input type="checkbox"/>	3	ASOCIACION CIVIL SIR ARTHUR CONAN DOYLE	NULL
<input type="checkbox"/>	4	INMOBILIARIA BIENES Y RAICES OLIMPO E.I.R.LTDA.	NULL
<input type="checkbox"/>	5	"TRANSPORTE BAYER'S SOCIEDAD ANONIMA CERRADA" OTOR...	NULL
<input type="checkbox"/>	6	COMUNIDAD CAMPESINA LICAS	NULL
<input type="checkbox"/>	7		
<input type="checkbox"/>	8	ADRA OFASA DEL PERU REGIONAL SUR ESTE DE JULIACA	NULL
<input type="checkbox"/>	9	INMOBILIARIA BIENES RAICES OLIMPO E.I.R.LTDA..	NULL
<input type="checkbox"/>	10	BANCOSUR - JULIACA REPRESENTADO POR VICTOR ARAGON ...	NULL
<input type="checkbox"/>	11	ASOCIACION PRO PRO VIVIENDA MAGISTERIAL AMAUTA	
<input type="checkbox"/>	12	LA MUNICIPALIDAD PROVINCIAL DE SAN ROMAN - JULIACA	
<input type="checkbox"/>	13	URBANIZACION MUNICIPAL TAPARACHI-SECTOR ALTO CERRO...	NULL
<input type="checkbox"/>	14	ASOCIACION DE PRODUCTORES AGROPECUARIOS LOS PINOS ...	NULL
<input type="checkbox"/>	15	"TRANSPORTES BAYERS SOCIEDAD ANONIMA CERRADA"OTORG...	NULL
<input type="checkbox"/>	17	SOCIEDAD DE BENEFICENCIA	NULL
<input type="checkbox"/>	18	"I A I G I F S I A C R I S T I A N A M A R A N A T A"	NULL

Figura 75. La Figura muestra la lista de la Tabla “involjuridicas1” (Involucrados Jurídicos)

	proy_id	proy_nombre	not_id	num_protocolo	folio_inicial	folio_final	escritura_inicial	escritura_final
<input type="checkbox"/>	1	2612	114	2612	11501	11999	4138	4327
<input type="checkbox"/>	2	2794	115	2794	1	500	1	274
<input type="checkbox"/>	3	2614	114	2614	12501	13000	4523	4720
<input type="checkbox"/>	4	2	115	2797	1551	2050	763	1060
<input type="checkbox"/>	5	2616	114	2616	13501	14148	4918	5161
<input type="checkbox"/>	6	2800	115	2800	3051	3550	1776	2306
<input type="checkbox"/>	7	2618	114	2618	501	1000	185	380
<input type="checkbox"/>	8	2798	115	2798	2051	2550	1071	1411
<input type="checkbox"/>	9	2799	115	2799	2551	3050	1412	1775
<input type="checkbox"/>	10	3551	115	2801	3551	4050	2307	2559
<input type="checkbox"/>	11	2617	114	2617	1	500	1	184
<input type="checkbox"/>	12	2802	115	2802	4051	4550	2560	2816

Figura 76. Las Figura presenta la lista de proyectos creados con el software

3.5.17 Interfaces y Prototipos del Sistema

Para el desarrollo de las interfaces y del prototipo del sistema se utilizaron los siguientes programas:

- StartUML, para la elaboración de los modelos UML presentados antes
- Sublime Text 3, para la codificación de los archivos HTML, CSS, PHP y otros.
- Navegador Google Chrome y Mozilla Firefox, con extensiones para las pruebas del sistema.

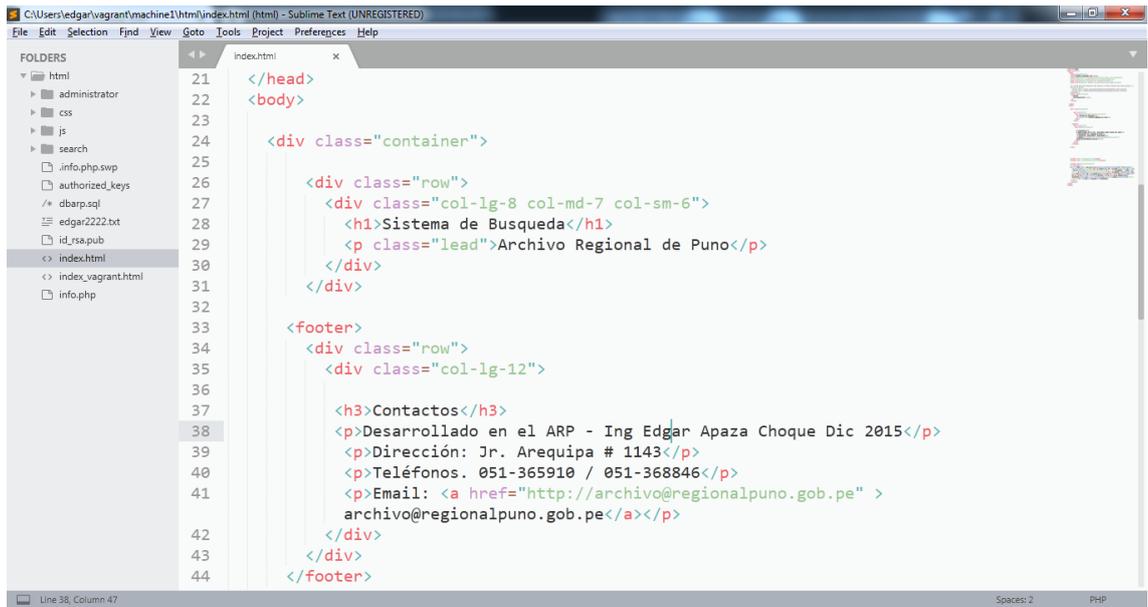
Como parte de la maquetación principal, se utilizó HTML5 con BOOTSTRAP que es un framework desarrollado y liberado por Twitter que tiene como objetivo facilitar el diseño web. Permite crear de forma sencilla webs de diseño adaptable, es decir, que se ajusten a cualquier dispositivo y tamaño de pantalla y siempre se vean igual de bien.

Algunas capturas de pantalla fueron tomadas en Ubuntu Linux, ya que la mayoría que trabaja en el área de informática del ARP, desarrollan aplicaciones usando software libre como Ubuntu Linux, CentOS, entre otros. Por ese motivo algunas de las capturas de pantalla se encuentran en este sistema operativo.

3.5.18 Implementación

En el proceso de implementación se efectuaron diferentes acciones, como el desarrollo de aplicaciones en PHP, sin la utilización de ningún framework, de los existentes, como Laravel, Symfony, Phalcon, etc., aunque no descartamos la utilización de algunos de estos.

Por lo tanto, se utilizó el modelo MVC, muy utilizado por muchos software y framework, aunque en otros, ha cambiado de MVC a MVT en el caso de Python y otras variantes, pero sigue siendo la base de la programación, especialmente en PHP. En la siguiente Figura se muestra el árbol de carpetas utilizado para la elaboración, usando sublime text 3, en su versión gratuita.



```
21 </head>
22 <body>
23
24 <div class="container">
25
26 <div class="row">
27 <div class="col-lg-8 col-md-7 col-sm-6">
28 <h1>Sistema de Busqueda</h1>
29 <p class="lead">Archivo Regional de Puno</p>
30 </div>
31 </div>
32
33 <footer>
34 <div class="row">
35 <div class="col-lg-12">
36
37 <h3>Contactos</h3>
38 <p>Desarrollado en el ARP - Ing Edgar Apaza Choque Dic 2015</p>
39 <p>Dirección: Jr. Arequipa # 1143</p>
40 <p>Teléfonos. 051-365910 / 051-368846</p>
41 <p>Email: <a href="http://archivo@regionalpuno.gob.pe" >
  archivo@regionalpuno.gob.pe</a></p>
42 </div>
43 </div>
44 </footer>
```

Figura 77. Vista del código del programa escrito en SublimeText3, archivo index.html

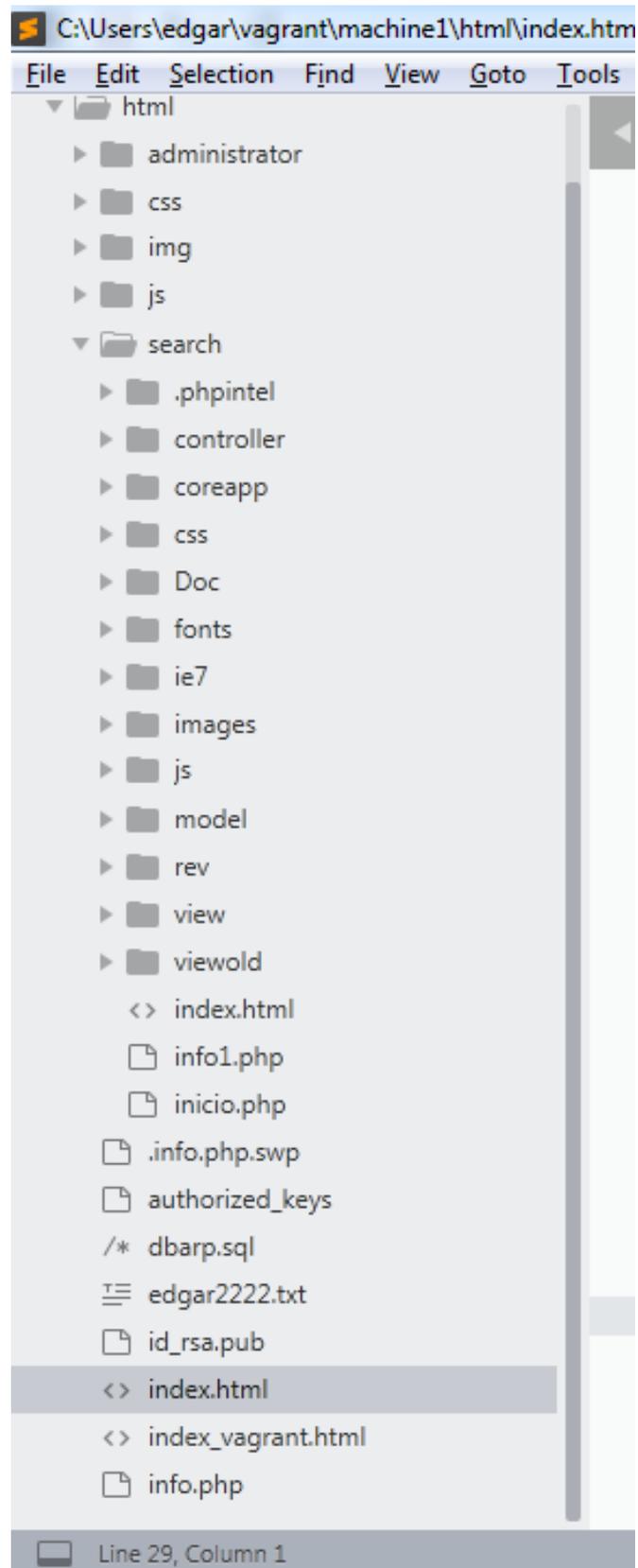


Figura 78. Árbol de carpetas utilizadas y se aprecia el MVC (Model, View, Controller)

La siguiente Figura muestra los permisos de cada carpeta, visto en el servidor Vagrant, siguiendo la nomenclatura de Linux.

```

vagrant@machine1: /var/www/html/search
authorized_keys  edgar2222.txt  index.html      js
css              id_rsa.pub      index_vagrant.html  search
vagrant@machine1:/var/www/html$ cd search/
vagrant@machine1:/var/www/html/search$ ls
controller  css  fonts  images  info1.php  js  rev  viewold
coreapp     Doc  ie7    index.html  inicio.php  model  view
vagrant@machine1:/var/www/html/search$ ls -lh
total 48K
drwxrwxr-x 1 vagrant vagrant 4.0K Dec  8 11:28 controller
drwxrwxr-x 1 vagrant vagrant  0 Dec  8 11:28 coreapp
drwxrwxr-x 1 vagrant vagrant 8.0K Dec  8 11:28 css
drwxrwxr-x 1 vagrant vagrant  0 Dec  8 11:28 Doc
drwxrwxr-x 1 vagrant vagrant  0 Dec  8 11:28 fonts
drwxrwxr-x 1 vagrant vagrant  0 Dec  8 11:28 ie7
drwxrwxr-x 1 vagrant vagrant 4.0K Dec  8 11:28 images
-rwxrwxr-- 1 vagrant vagrant 945 Oct  7 01:50 index.html
-rwxrwxr-- 1 vagrant vagrant 22 Dec 22 17:04 info1.php
-rwxrwxr-- 1 vagrant vagrant 5.7K Dec 22 17:06 inicio.php
drwxrwxr-x 4.0K Dec  8 11:28 js
drwxrwxr-x 1 vagrant vagrant 4.0K Dec  8 11:28 model
drwxrwxr-x 1 vagrant vagrant 8.0K Dec  8 11:28 rev
drwxrwxr-x 1 vagrant vagrant 4.0K Dec  8 11:28 view
drwxrwxr-x 1 vagrant vagrant 4.0K Dec  8 11:28 viewold
vagrant@machine1:/var/www/html/search$

```

Figura 79. Vista del árbol de carpetas y los permisos por cada carpeta y archivo

```

C:\Users\edgar\vagrant\machine1\html\search\coreapp\Conection.php (html) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
html
├── administrator
├── css
├── img
├── js
├── search
│   ├── .phpintel
│   ├── controller
│   └── coreapp
│       ├── Conection.php
│       └── config.php
├── css
├── Doc
├── fonts
├── ie7
├── images
├── js
└── model
index.html x Conection.php x config.php x
1 <?php
2
3 class Conection {
4
5     public function Conectar() {
6         require_once ("config.php");
7         $mysqli = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
8         $mysqli->set_charset("utf8");
9         if ($mysqli->connect_errno) {
10            echo "Error al contencar a MySQL: (".$mysqli->
11                connect_errno.") ".$mysqli->connect_error;
12            exit();
13        }
14
15        //echo $mysqli->host_info. " Test eb Conection.php";
16        return $mysqli;
17    }
18 }?>
Line 17, Column 2 Tab Size: 4 PHP

```

Figura 80. La carpeta Coreapp, conexión directa con la Base de Datos

```
1 <?php
2 /**
3  * Creado por Edgar Apaza
4  */
5 class Buscar
6 {
7
8
9
10 public function buscarsql($nombre,$paterno,$materno)
11 {
12     if(!is_null($nombre)){
13         if(!is_null($paterno))
14         {
15             if(!is_null($materno))
16             {
17                 //echo "sql (nombre, paterno, materno)";
18
19                 $nexo = "%";
20                 $sinEspacios = trim($nombre);
21                 $nom_temp = explode(" ", $sinEspacios);
22                 $nom_corregido = implode($nexo, $nom_temp);
23                 $query="SELECT Cod_inv, Nom_inv, Pat_inv, Mat_inv, otros FROM involucrados1 WHERE Nom_inv LIKE
24                 '%$nom_corregido%' AND Pat_inv LIKE '%$paterno%' AND Mat_inv LIKE '%$materno%' ORDER BY
25                 Pat_inv ";
26                 //$rpta = mysql_query($query);
27                 return $query;
28             }
29         }
30     }
31     //echo "sql (nombre, paterno)";
32
33     $nexo = "%";
34     $sinEspacios = trim($nombre);
35     $nom_temp = explode(" ", $sinEspacios);
36     $nom_corregido = implode($nexo, $nom_temp);
37     $query="SELECT Cod_inv, Nom_inv, Pat_inv, Mat_inv, otros FROM involucrados1 WHERE Nom_inv LIKE
38     '%$nom_corregido%' AND Pat_inv LIKE '%$paterno%' OR
39     //$rpta = mysql_query($query);
40 }
```

Figura 81. Vista del código de búsqueda usando Clases en PHP

```
1 <?php
2
3 class Listado {
4
5     private $con;
6
7     function __construct() {
8         require_once "../coreapp/Conexion.php";
9         $conexion = new Conexion();
10        $this->con = $conexion->Conectar();
11    }
12
13    public function EscriturasOtorgantes($codigo) {
14
15        //echo $this->con->host_info." Dentro de Notarios";
16        $sql = "SELECT flg_per FROM escritor WHERE cod_sct = ".$codigo;
17        $datos = $this->con->query($sql);
18        $fila = $datos->fetch_assoc();
19
20        if ($fila["flg_per"] == 0) {
21            $dat = self::ListadoOtorgantes($codigo);
22        }
23
24        if ($fila["flg_per"] == 1) {
25            $dat = self::ListadoOtorgantesJuridicos($codigo);
26        }
27
28        return $dat;
29    }
30
31    public function ListadoOtorgantes($codigo) {
32        //echo "Natural";
33        $lista_nombres = array();
34    }
```

Figura 82. Vista de la Clase Listado

```
1 <?php
2
3 $codigo = $_REQUEST["codigo_escritura"];
4 $lista = array();
5
6 function NombresOtorgantes($codigo) {
7     require_once '../model/ListadoNew.php';
8     $listado = new Listado();
9     $lista = $listado->EscriturasOtorgantes($codigo);
10
11     return $lista;
12 }
13
14 function NombresFavorecidos($codigo) {
15     require_once '../model/ListadoFavNew.php';
16     $listado = new ListadoFav();
17     $listaf = $listado->EscriturasFavorecidos($codigo);
18
19     return $listaf;
20 }
21 ?>
```

Figura 83. Vista del archivo CheckListadoNew.php

```
13
14
15 /*
16 *
17 * ZONA DE OTORGANTES
18 *
19 */
20
21 <h2 class="sub-header">Zona Otorgantes</h2>
22 <div class="table-responsive">
23     <table class="table table-striped">
24         <thead>
25             <tr>
26                 <th>Num</th>
27                 <th>Sub Serie</th>
28                 <th>Num Escritura</th>
29                 <th>Fecha Documento</th>
30                 <th>Proyecto</th>
31                 <th>Opciones</th>
32             </tr>
33         </thead>
34     </table>
35 </div>
36 <?php
37 /*+
38 * Obtener todos Los datos de Las escrituras a mostrar
39 */
40
41 $i=1;
42 while($fila1 = $result1->fetch_assoc())
43 {
44     $sqlEscritura1 = "SELECT num_sct, fec_doc, cod_sub, proy_id FROM escrituras1 WHERE cod_sct = ".$
45     cod_escritura;
46     $query1 = $mysqli->query($sqlEscritura1);
47     $row1 = $query1->fetch_assoc();
```

Figura 84. Vista del Archivo de detalles de Escrituras

Reportes Diarios de Ingreso a la Base de Datos

Opciones: Exportar la Lista a Excel | Menu Principal

Trabajador: Todos

Día: (Marque la Fecha en el Calendario)

Empieza: 27/12/2018

Finaliza: dd/mm/aaaa

Buscar

Total de Filas: 442
Trabajador: Todos los Usuarios

Hra_ing	Usuario	Cod_sct	Num_Sct
2010-12-13 06:01:02	1101	39376	149
2010-12-13 06:04:07	1103	39379	196
2010-12-13 06:05:40	1101	39380	149
2010-12-13 06:07:33	1103	39381	197
2010-12-13 06:07:40	1106	39382	4363
2010-12-13 06:10:09	1106	39383	4364
2010-12-13 06:10:18	1101	39384	150
2010-12-13 06:13:14	1103	39385	198
2010-12-13 06:13:44	1106	39386	4365
2010-12-13 06:16:09	1103	39387	199
2010-12-13 06:16:41	1106	39388	4366
2010-12-13 06:17:35	1101	39389	152

Figura 85. Vista de reportes e ingreso a Base de Datos por fechas.

CAPÍTULO IV

RESULTADOS Y DISCUSIÓN

Consideramos que uno de los logros más importantes que hemos logrado es colocar al ARP (Archivo Regional de Puno) en un lugar preponderante a nivel de todo el país, debido a que en ningún otro archivo regional existe o cuentan con un sistema de esta categoría. Los demás archivos tienen unos sistemas más pequeños, algunas funcionalidades que le permiten desarrollar mejor su trabajo, sin embargo, la cantidad de datos que poseen son limitadas.

A nivel de desarrollo de software también ha resultado importante, gracias a la cantidad de trabajo invertido en el software, y la cantidad de correcciones que han hecho que este sistema siga funcionando de manera ininterrumpida.

Hasta la fecha tenemos más de 360 mil registros de escrituras, y si el proyecto continúa, el número total a alcanzar es de 8 millones de registros. En base a un estudio realizado hace un par de años atrás, se había evaluado el avance que asciende a un 25% del avance total, aunque este número no quedará estático, debido principalmente, a que la cantidad de libros del fondo notarial, con el tiempo se sigue incrementando, ósea, cuando un notario fallece, toda la documentación de este notario, pasará a manos del ARP y se quedará en su custodia perpetua.

Ahora presentaremos algunos reportes extraídos principalmente del mismo sistema, el cual utilizamos para medir como estamos avanzando, y los reportes incluyen reportes personales, de grupo, tomados de manera mensual y anual.

Para esto, ingresaremos al módulo de administración, el cual se encarga de 3 cosas puntuales, 1) Muestra estadísticas anuales, Mensuales o de manera individual 2) Nos

permite visualizar los reportes diarios o entre fechas que uno desee, y 3) Realiza correcciones a los datos (control de calidad de la información), es que puede corregir los datos si hubiese algún error. Puede visualizar la Figura 53 para apreciar el menú y sus respectivas funciones.

Ingresaremos al menú de gráficos > Gráficos anuales.



Figura 86. Menú principal del módulo de Administración

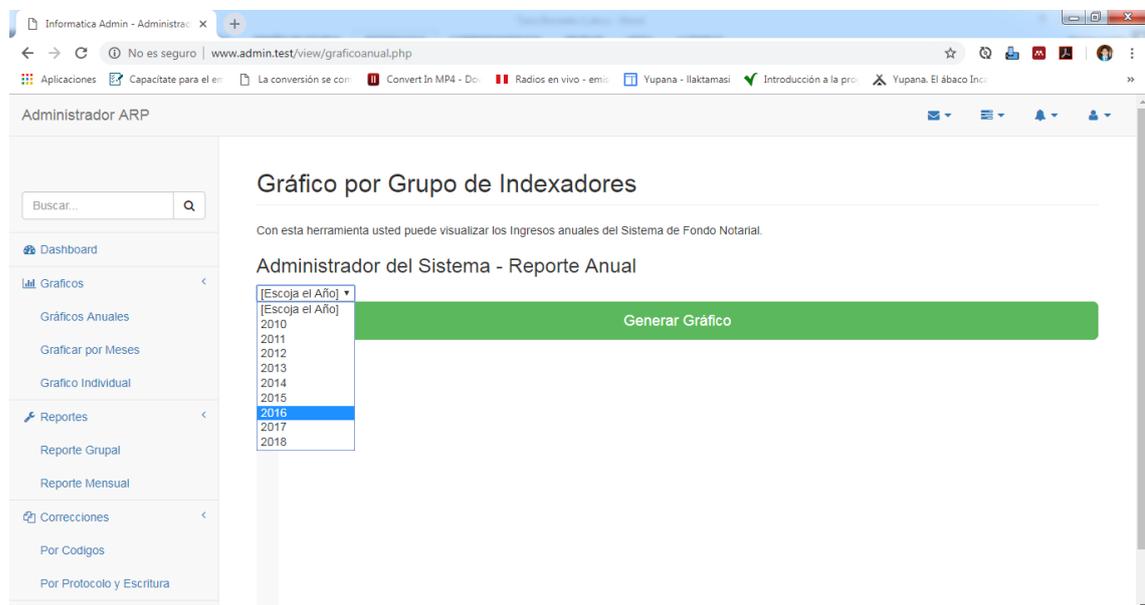


Figura 87. Ventana de reportes y gráficos estadísticos de la aplicación por años

Elegimos el año del reporte que deseamos, la información inicia desde el mes de Julio del 2010 hasta la fecha. Efectuamos una consulta para el año 2013, y obtenemos la siguiente información.



Figura 88. Figura del reporte de ingresos correspondientes al año 2013

Como se puede apreciar en la Figura, la cantidad mas alta de registros ingresados en este año fue de 18,843 escrituras.

Tabla 29

Ingreso de Escrituras correspondiente al año 2010

Trabajador	Ingresos Base Datos
ADMINISTRADOR ARP	71
Edgar Apaza Choque	3,996
Chester Miguel Zamudio Briceño	8,064
Grimanesa Ccori Valdivia	10,219
Victor Alejo Begazo	5,974
Roxana Julia Villa Ruiz	3,815
Florencio Ireneo Quispe Condemayta	10,190
REPORTE INGRESO	41
TOTAL	42,370

Fuente: Sistema Administrator

Tabla 30

Ingreso de Escrituras correspondientes al año 2011

Año 2011

Trabajador	Ingresos Base Datos
ADMINISTRADOR ARP	81
Edgar Apaza Choque	708
Chester Miguel Zamudio Briceño	489
Grimanesa Ccori Valdivia	205
Victor Alejo Begazo	2,693
Roxana Julia Villa Ruiz	5,602
Florencio Ireneo Quispe Condemayta	3,709
REPORTE INGRESO	104
German Calsin Yana	886
Alejandro Mamani Rodriguez	2,133
Rosa Churata Lezama	1,729
Tatiana Velez Gallegos	294
TOTAL	18,633

Fuente: Sistema Administrator

Tabla 31

Ingreso de Escrituras correspondientes al año 2012

Año 2012

Trabajador	Ingresos Base Datos
ADMINISTRADOR ARP	196
Edgar Apaza Choque	2,908
Roxana Julia Villa Ruiz	4,843
REPORTE INGRESO	30
Rosa Churata Lezama	76
Gryssset Roque Espinoza	3,090
Yesica Mamani Ramos	4,921
Nelly Chambilla Asqui	24
Esmeri Greis Mamani Quispe	18,390
Sonia Gutierrez Turpo	798
Gimbley Sandro Alata Tito	12,688
Sabina Colque Medina	14,402
Iveth Mariela Mamani Fernandez	1,801
Graciela Roque Mamani	3,524
gimbley alata tito	1,943
Rosmery Ccallata Gutierrez	2,417
Luz Marina Luque Jallo	3,725
Vilma Isabel Huaricallo Ramos	1,653
Yosseph Luna Lerma	68
TOTAL	77,497

Fuente: Sistema Administrator

Tabla 32

Ingreso de Escrituras correspondientes al año 2013

Trabajador	Ingresos Base Datos
Esmeri Greis Mamani Quispe	15,508
Iveth Mariela Mamani Fernandez	13,556
Graciela Roque Mamani	18,842
Luz Marina Luque Jallo	13,627
Zenaida Blanca Mamani Phocco	14,720
Rosa Amelia Churata Lezama	1,932
Maria Elena Laquise Quispe	2,343
TOTAL	80,528

Fuente: Sistema Administrator

Tabla 33

Ingreso de Escrituras correspondientes al año 2014

Año 2014

Trabajador	Ingresos Base Datos
Edgar Apaza Choque	39
Roxana Julia Villa Ruiz	1,862
Esmeri Greis Mamani Quispe	845
Iveth Mariela Mamani Fernandez	14,128
Graciela Roque Mamani	17,062
Gimble Alata Tito	1
Luz Marina Luque Jallo	2,736
Zenaida Blanca Mamani Phocco	18,210
Rosa Amelia Churata Lezama	22,328
Maria Elena Laquise Quispe	14,573
Corina Vilma Rodriguez Choque	2,346
Sofia Mamani Ccopa	184
Emilio Ramos Huaraya	528
Sandro Alata Tito	594
Daniel Zevallos Vega	24
TOTAL	95,460

Fuente: Sistema Administrator

Tabla 34

Ingreso de Escrituras correspondientes al año 2015

Año 2015

Trabajador	Ingresos Base Datos
Edgar Apaza Choque	4
Rosa Amelia Churata Lezama	22,035
Emilio Ramos Huaraya	3
Fredy Ordoñez Huaraya	5,248
Julio Cesar Condorena Mamani	6,327
Amanda Janeth Mendoza Flores	9,854
TOTAL	43,471

Fuente: Sistema Administrator

Tabla 35

Ingreso de Escrituras correspondiente al año 2016

Año 2016

Trabajador	Ingresos Base Datos
Edgar Apaza Choque	1
Grimanesa Ccori Valdivia	758
Julio Cesar Condorena Mamani	174
Amanda Janeth Mendoza Flores	59
Edith Marleny Huanca Condemayta	3,376
Jorge Vargas Gonzales	519
Dilman Quispe Luna	31
Braddy Gabriel Abarca Aranibar	119
Johnny Daniel Ojeda Gomez	1,598
Dante Elvis Cachi Alanoca	226
Franz Waldir Jalire Calderon	463
Gregorio Rosa Chambilla	1,513
Gonzalo Brolyn Flores Bermejo	618
David Huanca Chino	822
Maria Isabel Sanjinez Copari	1,252
Diego Hugo Yucra Mamani	62
TOTAL	11,591

Fuente: Sistema Administrator

Tabla 36

Ingreso de Escrituras correspondientes al año 2017

Año 2017

Trabajador	Ingresos Base Datos
Edith Marleny Huanca Condemayta	54
Lucy Lelia Villa Perez	88
Livia Ada Ccuno Condori	32
TOTAL	174

Fuente: Sistema Administrator

La siguiente Tabla muestra el número consolidado de escrituras ingresadas correspondientes al año 2010 al 2017, y es como sigue:

Tabla 37

Número total de escrituras ingresadas por años y Total general

AÑO	Num. Escrituras
2010	42,370
2011	18,633
2012	77,497
2013	80,528
2014	95,460
2015	43,471
2016	11,591
2017	174
TOTAL	369,724

Fuente: Sistema Administrator



Figura 89. Ingreso de Escrituras por años 2010- 2017

De la Tabla 37 podemos observar que gracias a este Software, se han ingresado 369,724 escrituras a los largo de los años, lo que permite a los trabajadores del ARP poder acceder de manera rápida a la localización de estos proyectos.

Otro de los aspectos importantes de este proyecto, es la gran cantidad de personal que ha participado en el desarrollo del proyecto, específicamente 66 usuarios registrados en Nuestra base de datos, desde personal nombrado, personal contratado y practicantes.

Tabla 38

Listado del personal involucrado en el Proyecto Fondo Notarial

Código de Usuario	Nombre	Paterno	Materno	Nivel Usuario
1101	Edgar	Apaza	Choque	2
1102	Chester Miguel	Zamudio	Briceño	2
1103	Grimanesa	Ccori	Valdivia	2
1104	Victor Alejo	Begazo	Begazo	2
1105	Roxana Julia	Villa	Ruiz	2
1106	Florencio Ireneo	Quispe	Condemayta	2
1100	Administrador	ARP		1
1110	Reporte	Ingreso		3
1111	Búsqueda	Archivo		4
1112	German	Calsin	Yana	2
1115	Alejandro	Mamani	Rodriguez	2
1116	Rosa	Churata	Lezama	2
1117	Tatiana	Velez	Gallegos	2
1120	Gryset	Roque	Espinoza	2
1121	Yesica	Mamani	Ramos	2
1122	Nelly	Chambilla	Asqui	2
1123	Esmeri Greis	Mamani	Quispe	2
1124	Sonia	Gutierrez	Turpo	2
1125	Gimble Sandro	Alata	Tito	2
1126	Sabina	Colque	Medina	2
1127	Iveth Mariela	Mamani	Fernandez	2
1134	Luz Marina	Luque	Jallo	2
1135	Vilma Isabel	Huaricallo	Ramos	2
1136	Yosseph	Luna	Lerma	2
1131	Graciela	Roque	Mamani	2



Código de Usuario	Nombre	Paterno	Materno	Nivel Usuario
1132	Gimbley	alata	tito	2
1133	Rosmery	Ccallata	Gutierrez	2
1137	Juan Jose	Pari	Mamani	2
1138	Rosa	Quispe	Mamani	2
1139	Milagros	Alvarez	Avalos	2
1140	Zenaida Blanca	Mamani	Phocco	2
1141	Rosa Amelia	Churata	Lezama	2
1142	Maria Elena	Laquise	Quispe	2
1144	Corina Vilma	Rodriguez	Choque	2
1145	Sofia	Mamani	Ccopa	2
1146	Emilio	Ramos	Huaraya	2
1147	Sandro	alata	tito	2
1148	Anthony	Herrera	Machicao	2
1149	Daniel	Zevallos	Vega	2
1150	ARCHIVO	REGIONAL	PUNO	1
1151	Fredy	Ordoñez	Huaraya	2
1152	Julio Cesar	Condorena	Mamani	2
1153	Amanda Janeth	Mendoza	Flores	2
1154	Emilio	Ramos	Huaraya	1
1155	anthony	herrera	machicao	1
1156	Edwin	flores	choque	1
1157	nancy melania	taype	turpo	3
1158	William	quinteros	oblitas	1
1159	Liliana	Condori	Carcasi	1
1160	Henry Alcides	BALCONA	COILA	1
1161	JHOEL	ARIZACA	CARI	1
1162	Edith Marleny	Huanca	Condemayta	2
1163	Edith Marleny	Huanca	Condemayta	1

Código de Usuario	Nombre	Paterno	Materno	Nivel Usuario
1164	Jorge	Vargas	Gonzales	2
1165	Dilman	Quispe	Luna	2
1166	Braddy Gabriel	Abarca	Aranibar	2
1167	Johnny Daniel	Ojeda	Gomez	2
1168	Dante Elvis	Cachi	Alanoca	2
1169	Franz Waldir	Jalire	Calderon	2
1170	Gregorio	Rosa	Chambilla	2
1171	Gonzalo Brolyn	Flores	Bermejo	2
1172	David	Huanca	Chino	2
1173	Maria Isabel	Sanjinez	Copari	2
1174	Diego Hugo	Yucra	Mamani	2
1175	Lucy Lelia	Villa	Perez	2
1176	Livia Ada	Ccuno	Condori	2

Fuente: Base de Datos “dbarp” Tabla usuarios.

La Tabla N° 38 muestra la cantidad de trabajadores registrados en el sistema y sus respectivos accesos, el cual se puede visualizar en la última columna (nivel usuario), para lo cual presentamos la Tabla de las asignaciones según el perfil del personal.

Tabla 39

Tabla para la asignación de perfiles de usuarios y accesos al sistema

Perfil de asignación del personal	Código
Administrador	1
Personal Ingreso de Escrituras	2
Personal solo Búsquedas	3

Otro punto importante es la utilización de tecnología web, que simplifica los procesos, por lo menos en una parte, la utilización de clases, así como también de AJAX, en el cual no se ha presentado ahora, sin embargo, en otras mejoras se han utilizado estas herramientas para mejorar la experiencia del usuario.

Tabla 40

Datos estadísticos recogidos de la Base de Datos del sistema Fondo notarial

DATOS ESTADÍSTICOS DE BASE DE DATOS

CATEGORÍA	CANTIDAD
Nombres de personas como Otorgantes	449,128
Nombres de personas como Favorecidos	441,965
Nombres de Instituciones o nombres Jurídicos	17,721
Total de Notarios	117
Número de Proyectos completados	421

Fuente: Sistema Fondo Notarial, Base de datos

Tabla 41

Número de solicitudes atendidas por años en el ARP 2011 al 2016

Año	Numero de Solicitudes	Escrituras Publicas	% Escrituras	Partidas	% Partidas	Otros	% Otros
2011	5,469	3,949	72%	488	9%	1032	19%
2012	6,922	5,703	82%	830	12%	389	6%
2013	2,343	2,020	86%	209	9%	114	5%
2014	7,890	6,738	85%	725	9%	427	5%
2015	7,646	6,643	87%	609	8%	394	5%
2016	7,692	6,899	90%	495	6%	298	4%
TOTAL	37,962	31,952	84%	3356	9%	2654	7%

Fuente: Sistema Recepción ARP

Como se puede apreciar en la Tabla N°40 se puede observar que más del 82% de las solicitudes que atiende al ARP son solicitudes de escrituras públicas. Para el año 2011 de las 5469 solicitudes de búsquedas, el 72.21% (3,949 solicitudes) fueron solicitudes de escrituras públicas, un 8.92% (488 solicitudes) fueron de partidas (Nacimiento, Matrimonio o defunción), y un 18.87% (1032 solicitudes) fueron de otro tipo de documentos (Expedientes civiles, penales, etc.). Para el año 2012 de los 6922 solicitudes el 82.39% de estos corresponden a solicitudes de búsqueda, 11.99% para búsqueda de

partidas y 5.62% para otros documentos. Para el año 2013 el número de solicitudes recibidas fueron de 2343, y esto se debió a que el ARP cambio de local, y mucha gente no sabía dónde se habían trasladado, sin embargo el 86.21% de estas solicitudes (2020) son para búsqueda de escrituras públicas, el 8.92% corresponde a búsqueda de partidas, y un 4.87% de estas a otros documentos. Para el año 2014 de los 7890 solicitudes recibidas el 85.40% de estas corresponden a búsqueda de escrituras públicas, el 9.19% corresponde a partidas y el 5.41% a otros documentos. Para el año 2015 de las 7646 solicitudes de búsquedas el 86.88% corresponden a búsquedas de escrituras públicas el 7.96% a partidas y un 5.15% a solicitudes de búsqueda de otros documentos. Para el 2016 de los 7692 solicitudes el 89.69% (6899) corresponden a solicitudes de búsqueda de escrituras públicas, el 6.44% a partidas, y el 3.87% a otros documentos.

En total, en estos 6 años que se tienen registros, se han recogido 37,962 solicitudes de búsqueda, de las cuales el 84.17% (31,952 solicitudes) corresponden a búsqueda de escrituras públicas, el 8.84% corresponde a búsquedas de partidas y el 6.99% a solicitudes de búsquedas de otros documentos, lo que prueba que la mayor cantidad de solicitudes que recibe el ARP supera el 84%, y por esa razón es que el proyecto tubo un impacto a diferentes niveles del ARP, económicos, de prestigio y otras que acompañan este proyecto.



Figura 90. Número de solicitudes recibidas para búsqueda por años



El total de usuarios atendidos en este periodo del 2011 al 2016 asciende a 27,223 personas en su mayoría ciudadanos de diferentes partes de nuestra región, de otras ciudades, y países, en particular investigadores.

CONCLUSIONES

- En vista de la problemática que atraviesa el Archivo Regional de Puno, que no contaba con ningún sistema informático funcionando, realizando todos sus trámites y atención al público de manera manual, es que nos planteamos el objetivo de desarrollar un Sistema de Registros Notariales usando un sistema de virtualización Vagrant (docker) para acelerar la búsqueda y recuperación de información, con 369,724 escrituras ingresadas, 449,128 nombres de personas como otorgantes de escrituras públicas y 441,965 nombres de personas favorecidos, 17,721 instituciones públicas registradas (Municipalidades, colegios, empresas, etc.), 115 notarios de toda la región puno y trabajando en este proyecto más de 75 personas entre personal nombrado y contratado.

Desde de la perspectiva de desarrollo, se ha realizado un trabajo serio en el análisis y diseño de sistemas usando metodologías ágiles, para el desarrollo, haciendo uso de UML para el modelado del sistema, junto con el análisis de base de datos, haciendo uso de la normalización de base de datos, la cual nos ha dado el soporte para implementar una serie de mejoras con el tiempo, incluso se le ha implementado vistas, procedimientos almacenados y en algunos casos trigger en MySQL. A nivel de código, se comenzó con la versión 4 de PHP, y en el cambio de la versión 4 a 5 en PHP, solo se hicieron pocos cambios debido a que todo de ha trabajado con clases en PHP y de forma modular, haciendo uso de MVC que nos resultó sumamente útil. Muchos de estos cambios, se han registrado usando el sistema de control de versiones GIT y almacenados en GitHub, de lo cual tenemos los cambios que se han realizado y las últimas mejoras al código que se han hecho, actualmente trabajando con la versión de PHP la 7.2, con la base de datos en MySQL corriendo en un servidor instalado con CentOS y Debían.

- El control de versiones, se realizó mediante GIT, que se ha adoptado después de haber comenzado el proyecto, llegando a utilizarlo de manera completa en los meses y años posteriores.
- Una de las cosas que nos ha quedado claro, después de terminar el proceso, es la robustez que el sistema presenta, debido a que en estos últimos años, no ha requerido ningún mayor significativo, salvo las mejoras en la parte de las interfaces del usuario y rediseño de las ventanas, sin embargo, la funcionalidad continua trabajando desde su diseño original. Esto se logró gracias a la utilización de las metodologías ágiles como XP principalmente. Se ha comenzado a analizar desde los objetivos principales y requisitos que deben cumplir, los módulos que eran necesarios crear y definir, las prioridades y otros artefactos, que según la metodología XP ha servido, así como la programación en parejas que ha resultado ser muy productiva. Se han identificado las variables y otros artefactos que intervinieron en el proyecto, cumpliendo de esa manera con nuestro primer objetivo específico.
- Al inicio del proyecto de los 3 paquetes que este conforma en su totalidad, solo se implementó el primero (Modulo de Ingreso de escrituras), llegando a ingresar más de 42 mil escrituras en solo 6 meses, sin embargo, aún no se contaba con un sistema de búsqueda y de corrección de errores y control de calidad, esta ultima se realizaba de manera directa en la base de datos. El diseño e implementación de los 2 módulos faltantes, también se hicieron siguiendo la metodología ágil y los resultados son los que hemos presentado, colaborando en el proyecto alumnos de institutos tecnológicos superiores y de universidades de carreras afines, quienes también se han beneficiado de los desafíos que representaba el desarrollo de este sistema y la influencia que tiene en la actualidad en el trabajo diario del personal del archivo regional.
- Nuestro segundo objetivo fue aplicar las técnicas y metodologías ágiles para el desarrollo de software haciendo uso de la virtualización, comenzando inicialmente el proyecto usando VirtualBox versión 4 (<https://virtualbox.org>), instalando la versión de GNU/Linux CentOS en la versión 5, siendo una versión muy ligera, bastante completa, ya que pertenece a una rama de RedHat, (esta última de pago). Después de que haber probado este sistema y validar su funcionamiento, se adquirió un servidor IBM x3400 exclusivamente para este software. El funcionamiento de este sistema en la red, resulto mucho mas rápido

en comparación de una virtualización, y la transferencia de los datos fue mucho más rápida, trabajando entre 7 personas en el ingreso y con unos 6 para búsquedas.

- Todo este proceso, ha sido testeado desde sus inicios, sin embargo, debemos indicar que el uso de adopción de GIT como sistema de control de versiones, ha hecho que este proceso de control fuera mucho mas sencillo y mas manejable debido al uso de las ramas que permite crear y desarrollar, implementar mejoras sin tocar el código fuente que corre sin problemas, y esto ha facilitado y permitido que se puedan probar cambios sin que este afecte al desarrollo de la aplicación, debido a que trabajamos en un rama paralela del mismo proyecto, y si “rompemos” algo, o deseamos regresar hacia atrás en el tiempo, GIT nos proporciona todas estas herramientas que han sido muy útiles en este proceso. De esta manera se han identificado y corregido los fallos, hasta lograr un producto que se ha mantenido muy firme en el tiempo.
- Esperamos que las aplicaciones y técnicas utilizadas en este software, sirva de ayuda e inspiración para aquellos que les interesa este mundo de la digitalización y la archivística.

RECOMENDACIONES

- En este proyecto solo se ha tocado sobre la implementación y puesta en funcionamiento de un sistema informático, que ha contemplado desde la parte técnica, el uso de los conocimientos de la ingeniera de software y redes, así como las capacitaciones tanto al equipo de desarrollo y al personal del Archivo Regional y otras acciones; sin embargo, es necesario plantear algunas recomendaciones que consideramos importantes.
- En el presente trabajo, no se ha tocado sobre el tema de seguridad tanto para el acceso de la información, acceso al servidor, SSH, el uso del protocolo HTTPS, entre otras que no se han tocado en estos documentos, sobre la seguridad a nivel de redes con el uso de firewall, Proxy, encriptación usando AES entre otras, que se han utilizado para este proyecto, sin embargo, no se ha profundizado en estos temas. Sugerimos encarecidamente, que no se deje de lado la seguridad del sitio, y hacer uso de métodos de acceso de doble pacto, mediante envíos de códigos SMS al celular o correo electrónico, entre otras, debido a que nosotros consideramos que todas las aplicaciones deben estar protegidas de accesos de fuera, protegida de ataques de terceros, troyanos, etc.
- El proyecto es básicamente fotografiar los documentos, y almacenarlos en dos formatos, uno de alta calidad o resolución y la otra de alta calidad, pero en JPG, lo que nos permite ahorrar una cantidad enorme de datos en el disco duro, sin embargo, nuestra sugerencia es que cada imagen debe ser firmada digitalmente, haciendo uso de la firma digital, que la proporciona la RENIEC, que es solicitada por el gobierno regional, el cual nunca se hizo, pero que estaba considerado dentro del proyecto original, convirtiéndose a un trabajador en un “fedatario digital” para que este atienda las peticiones vía web o por otros medios.

- Este es un proyecto muy ambicioso, ya que conlleva muchas mejoras, como la indexación y unión de las minutas a las escrituras públicas, la digitalización y atención a investigadores mediante pantallas táctiles, y la facilidad de transferir las imágenes mediante la red, usando protocolos como sockets.
- Este sistema debe estar corriendo en un mejor servidor, con mejores características de hardware y con una cantidad de memoria RAM más alta, y que la estructura del cableado de red sea la mejor, debido a que la infraestructura de la red del Archivo Regional, es empírica, y no tiene comprado ni asignado ninguna de las especificaciones de red requeridos para el sistema, así como el aseguramiento de los datos a nivel de capa 3 en Switch administrables, y otros componentes que son necesarios para que este proyecto pueda funcionar mejor. Utilizar nuevas tecnologías y equipos, los cuales ofrecen una mayor cantidad de herramientas, y formas y métodos de trabajo; nosotros nos vimos limitados a lo que existía en su momento, sin embargo, pudimos realizar este proyecto sin mayores contratiempos.
- Considero de manera personal, que un sistema operativo Linux es mucho más que eficiente, seguro y fácil de utilizar que uno en Windows –en parte por las licencias– la que no permiten un mejor desarrollo de estas, aunque en la actualidad, Microsoft ha cambiado su forma de trabajo, y liberado varios programas de manera gratuita, así como la creación de máquinas virtuales usando AZURE de Microsoft.
- El uso de máquinas virtuales como AZURE de Microsoft, AWS de Amazon o Google Cloud, etc., podrían formar parte de una nueva solución informática, migrar todo a AWS incrementaría la velocidad de transferencia de los datos, aseguraría una copia de respaldo de nuestra base de datos, así como que el sistema estaría siempre disponible.
- Otro aspecto que me gustaría implementar en el futuro, es que la aplicación pueda trabajar mediante una VPN (Virtual Private Network) y que los accesos desde otros lugares sean de manera segura, usando métodos de encriptación de datos lo que aseguraría total transparencia a la hora de utilizar estos recursos, claro está que toda la infraestructura debe cambiar, y que la forma de trabajo también es necesario que cambie.

BIBLIOGRAFÍA

- Arce, J. L. (2016). *Desarrollo de un sistema web de gestión de documentos y portafolio virtual para la institución Educativa Básica “Maria de Nazareth” de la ciudad de Guayaquil*. Universidad de Guayaquil.
- Archivo General de la Nación. *Norma para la organización de documentos archivísticos en la entidad pública.*, Pub. L. No. Directiva Nro 10-2019-AGN/DDPA, 1 (2019).
- Archivo Regional de Puno, O. (2015). *Plan Operativo Institucional 2015 - Archivo Regional de Puno* (p. 18). p. 18. Puno: Oficina Técnica Administrativa.
- Asato E., J. A., Ramirez G., E., Fierro M., J. G., Gutiérrez V., F., y Galván M, P. (2017). Aplicación de marcos de trabajo ágil para el desarrollo de software en ambientes académicos. *Pistas Educativas 127 (CITEC 2017)*, 39(04), 6–25.
Retrieved from
<http://www.itcelaya.edu.mx/ojs/index.php/pistas/article/view/1083/874>
- Bahit, E. (2011). POO y MVC en PHP. *Argentina: Openlibra.*, 66.
- Carlos Ble, J., y Beas, M. (2010). Diseño Ágil con TDD. *Www.IExpertos.Com*, 309.
- Decreto Ley N° 1049, P. (2006). *Ley del Notariado*.
- Egas, L. M., y Jativa, J. X. (2014). *Implementación de una metodología de desarrollo ágil para el análisis, diseño e implementación del sistema software y aplicación de un caso de estudio práctico Latacunga, durante el período abril – diciembre del 2013*. Universidad de las Fuerzas Armadas.
- García-holgado, A. (2014). *la implantación de ecosistemas tecnológicos corporativos*

- Máster en Sistemas Inteligentes Trabajo Fin de Máster Análisis de integración de soluciones basadas en software como servicio para la implantación de ecosistemas tecnológicos corporativos Alicia Gar. *Artículo*, (July 2013).
- García-Holgado, A., y García-Peñalvo, F. (2018). *Metodologías de Ingeniería de Software*. Retrieved from <https://repositorio.grial.eu/bitstream/grial/1149/1/4.Metodologia.pdf>
- Godoy, D. alberto. (2017). *Diseño de un Simulador Dinámico de Proyectos de Desarrollo de Software que utilizan Metodología Scrum*. 91, 399–404.
- Hilario Fernandez, E. A. (2018). *Desarrollo de un sistema web para el control de biblioteca en la institución educativa Santa Rosa de Santo Domingo - Huarmey 2017*. Universidad Catoloca Los Angeles de Chimbote.
- Laiza Torres, D. J. (2015). *Uso y eficacia del Linux y Alfresco para implementar un software de documentos digitalizados en la red local de la empresa Supplies Corporation del Perú S.A* (Universidad Científica del Sur). Retrieved from http://repositorio.cientifica.edu.pe:8080/bitstream/handle/UCS/356/TL-Laiza_Torres.pdf?sequence=1&isAllowed=y
- LEY N° 27658, P. *Ley Marco de Modernización de la Gestión del Estado LEY N° 27658.*, (2002).
- Moreiro, J.-A., Sánchez-Cuadrado, S., Palacios, V., y Barra, E. (2011). Evaluación de Software Libre Para la Gestión de Archivos Administrativos. *El Profesional de La Información*, 20(2), 206–213. <https://doi.org/10.3145/epi.2011.mar.12>
- Muñoz Soro, J. F., y Nogueras-Iso, J. (2014). *La digitalización de documentos en la Administración de Justicia*. 8(1), 49–53.
- Normas Legales, P. *Normas legales 422679. DECRETO SUPREMO N° 010-201-JUS.*, (2010).
- Núñez Paula, I. A., y Núñez Govín, Y. (2005). ACIMED. In *ACIMED* (Vol. 13). Retrieved from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1024-94352005000200003
- Oppel, A. (2010). Fundamentos de Bases de Datos. In *campusMVP.es*. <https://doi.org/10.1017/CBO9781107415324.004>

- Oppel, A., y Sheldon, R. (2010). *Fundamentos de SQL*. Retrieved from http://pedrobeltrancanessa-biblioteca.weebly.com/uploads/1/2/4/0/12405072/fundamentos_de_sql_3edi_oppel.pdf
- Proyectos Agiles.org. (n.d.). Qué es SCRUM – Proyectos Ágiles. Retrieved November 29, 2018, from <https://proyectosagiles.org/que-es-scrum/>
- Proyectos Agiles.org. (2018). Métricas ágiles y cuadro de mandos integral para Scrum – Proyectos Ágiles. Retrieved June 1, 2018, from Métricas ágiles y cuadro de mandos integral para Scrum website: <https://proyectosagiles.org/2008/12/07/metricas-agiles-cuadro-mandos-balanceado-scrum/>
- Roger S. Pressman. (2005). *Ingenieria del Software -Roger Pressman 6th.Ed.pdf* (p. 980). p. 980.
- Romero, M. *Marco general de la funcion notarial en la formalizacion de instrumentos protocolares.* , (2014).
- Schwaber, K., y Sutherland, J. (2013). La Guía de Scrum. *Scrumguides.Org, 1*, 21. Retrieved from <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-ES.pdf>
- Takeuchi, H., Nonaka, I., Owner, P., Scrum, D., Meeting, S. R., Retrospective, S., ... Development, A. S. (2014). *Historia Características de Scrum Roles en Scrum*. 1–6.
- Villar, J. (2009). Introducción a la OOP y al C ++. *Grupo Eidos*, 1–49. Retrieved from <http://kataix.umag.cl/~ruribe/Utilidades/Introduccion a la Programacion Orientada a Objetos.pdf>
- Wikipedia. (2018). Git. Retrieved December 11, 2018, from 24 nov 2018 website: <https://git.kernel.org/?p=git/git.git;a=tree>
- Zambrana, E. (2003). Modelo Estándar. Universidad Mayor de San Andrés - UMSA.



ANEXOS

Anexo 1. Código fuente principales características

Clase Conexión a Base de Datos, dentro de la Carpeta coreapp.

```
<?php
class Conexion {

    public function Conectar() {
        require_once ("config.php");
        $mysqli = new mysqli(DB_HOST, DB_USER, DB_PASSWORD, DB_NAME);
        $mysqli->set_charset("utf8");
        if ($mysqli->connect_errno) {
            echo "Error al conectar a MySQL:
                (".$mysqli->connect_errno.") ".$mysqli->connect_error;
            exit();
        }

        //echo $mysqli->host_info. " Test Conexion.php";
        return $mysqli;
    }
}
?>
```

Clase listado, Finalidad: La clase se conecta a la base de datos y proporciona un listado por otorgante, favorecido sean personas jurídicas o naturales.

```
<?php
class Listado {

    private $con;

    function __construct() {
        require_once "../coreapp/Conexion.php";
        $conexion = new Conexion();
        $this->con = $conexion->Conectar();
    }

    public function EscriturasOtorgantes($codigo) {

        //echo $this->con->host_info." Dentro de Notarios";
        $sql = "SELECT cod_inv_ju FROM escritor1 WHERE cod_sct=".$codigo;
        $datos = $this->con->query($sql);
        $fila = $datos->fetch_assoc();

        if ($fila["cod_inv_ju"] == 0) {
            $dat = self::ListadoOtorgantes($codigo);
        }

        if ($fila["cod_inv_ju"] != 0) {
            $dat = self::ListadoOtorgantesJuridicos($codigo);
        }

        return $dat;
    }

    public function ListadoOtorgantes($codigo) {
```

```
//echo "Natural";
$lista_nombres = array();

$sql = "SELECT cod_inv FROM escritor1 WHERE cod_sct = ".$codigo;
$datos = $this->con->query($sql);

//echo "Numero de Datos: ".$datos->num_rows."<br>";

while ($lista = $datos->fetch_assoc()) {
    $lista_nombres[] = self::CambiarNombres($lista["cod_inv"]);
}

return $lista_nombres;
}

public function ListadoOtorgantesJuridicos($codigo) {
    //echo "Juridicos";
    $lista_nombres = array();
    $sql = "SELECT cod_inv_ju FROM escritor1 WHERE cod_sct = ".$codigo;
    $datos = $this->con->query($sql);

    //echo "Numero de Datos: ".$datos->num_rows."<br>";

    while ($lista = $datos->fetch_assoc()) {
        $lista_nombres[] = self::CambiarNombresJuridicos($lista["cod_inv_ju"]);
    }

    return $lista_nombres;
}

public function CambiarNombres($codigo) {

    $sql = "SELECT CONCAT(nom_inv, ' ', pat_inv, ' ', mat_inv) AS persona FROM
involucrados1 WHERE cod_inv = ".$codigo;
    $datos = $this->con->query($sql);
    $lista = $datos->fetch_assoc();

    return $lista;
}

public function CambiarNombresJuridicos($codigo) {

    $sql = "SELECT Raz_inv FROM involjuridicas1 WHERE Cod_inv = ".$codigo;
    $datos = $this->con->query($sql);
    $lista = $datos->fetch_assoc();

    return $lista;
}
}
?>
```

Archivo usuarios.php en la carpeta Model, que verifica si un usuario existe o no

```
<?php
function verificar($usuario, $clave) {
```

```
require '../coreapp/Conexion.php';
$conection = new Conexion();
$con     = $conection->Conectar();

$rpta;
$sql = "SELECT cod_usu FROM usuarios WHERE log_usu='$usuario' AND psw_usu ='$clave'
LIMIT 0,1;";

$result = $con->query($sql);
$numero = $result->num_rows;

if ($numero > 0) {
    $rpta = TRUE;
} else {
    $rpta = FALSE;
}

return $rpta;
}
?>
```

Archivo: check_usuario.php de la carpeta Controller

```
<?php
session_start();
include "../model/usuarios.php";

$user = $_POST['username'];
$pas = $_POST['password'];

$rpta = verificar($user, $pas);

if ($rpta == 1) {
    header("Location: ../view/inicio.php");
    $_SESSION['busqueda'] = "busca";
} else {
    header("Location: ../index.html?error=error");
}
?>
```