

Sistema web de correspondencia para la gestión de documentación electrónica aplicando firma digital en los trámites del Gobierno Autónomo Municipal de Entre Ríos, Bolivia

Correspondence web system for the management of electronic documentation applying digital signature in the procedures of the Municipal Autonomous Government of Entre Ríos, Bolivia

David Flores

davichof31@gmail.com
Código ORCID: 0000-0001-9776-2754

Víctor Hugo Pérez Rojas

victorvirtual@gmail.com
Código ORCID: 0000-0002-0809-7471

**Gobierno Autónomo del Departamento Cochabamba
Cochabamba - Bolivia**

Artículo aceptado en septiembre 2020 Arbitrado en octubre 2020 Publicado en enero 2021

Resumen

El Gobierno Autónomo Municipal de Entre Ríos realiza la gestión de correspondencia, aplicando procedimientos manuales y utiliza la firma hológrafa en documentos, ocasionando pérdidas de tiempo e incertidumbre sobre el estado de la documentación. Para solucionar este problema, se propone un sistema web de correspondencia para la gestión de documentación electrónica aplicando firma digital. Teóricamente, se investigó sobre firma digital, algoritmos, estándares tecnológicos y normativas legales. Metodológicamente, se aplicó la Programación Extrema para construir un sistema que satisfaga los requerimientos establecidos por el cliente. El desarrollo del software, del lado del servidor, se centró en el uso de lenguajes de software libre y el gestor de base de datos. Por el lado del cliente, el Framework utilizado fue Angular 8. Para lograr la conectividad entre cliente y servidor se empleó la arquitectura cliente servidor. Finalmente, se adquirieron los Tokens, certificados digitales para realizar la firma y verificación, estableciendo una conexión con los servicios de la Agencia para el Desarrollo de la Sociedad de Información en Bolivia. En conclusión, se implementó el uso de la aplicación web y la firma digital como la vía más apropiada para garantizar la autoría e integridad de los documentos digitales.

Palabras clave: Firma digital, entidad certificadora, certificado digital.

Abstract

The Autonomous Municipal Government of Entre Ríos manages correspondence, applying manual procedures and uses the holographic signature on documents, causing loss of time and uncertainty about the status of the documentation. To solve this problem, a web correspondence system is proposed for the management of electronic documentation applying digital signature. Theoretically, the digital signature, algorithms, technological standards and legal regulations were investigated. Methodologically, Extreme Programming was applied to build a system that satisfies the requirements established by the client. The development of the software, on the server side, focused on the use of free software languages and the database manager. On the client side, the Framework used was Angular 8. To achieve connectivity between client and server, the client server architecture was used. Finally, the Tokens, digital certificates to perform the signature and verification, were acquired, establishing a connection with the services of the Agency for the Development of the Information Society in Bolivia. In conclusion, the use of the web application and the digital signature was implemented as the most appropriate way to guarantee the authorship and integrity of digital documents.

Keywords: Digital signature, certifying entity, digital certificate.

INTRODUCCIÓN

La gestión de la correspondencia en el Gobierno Autónomo Municipal de Entre Ríos (GAMER) tiene gran relevancia, ya que todo asunto o solicitud y trámite que sea externo o interno tiene que ser tratado bajo documentos físicos plasmados en papel, es decir que la población, los servidores públicos e instituciones tienen que realizar sus peticiones o solicitudes a través de correspondencia.

En las dependencias del GAMER se cursa diariamente un elevado número de correspondencia interna y externa de toda índole, que es de mucha importancia y requiere una atención inmediata, satisfactoria para el progreso y desarrollo del municipio. Estos procesos involucran determinados niveles de autorización mediante firma hológrafa y un significativo uso de papel para el resguardo de antecedentes.

Los trámites que se realizan en el GAMER son: Asuntos administrativos como aprobación de planos de urbanización, aprobación de planos de regularización de lote, aprobación de planos de lote, aprobación de planos de construcciones, bienes inmuebles, actualización de datos técnicos, certificaciones catastrales, certificación de uso del suelo, licencias de funcionamiento, así como también, la autorización eventual del uso de espacios públicos para actividades cívicas, religiosas, tradicionales y culturales. Además, se gestionan solicitudes de alumbrado público,

mejoramiento de calles urbanas y rurales, saneamiento básico y agua potable, entre otros.

La correspondencia interna es generada por los funcionarios o servidores públicos del Gobierno Autónomo Municipal. La administración pública ha utilizado desde siempre al papel como medio de soporte para las tramitaciones, fundamentalmente por el hecho de que, en las intervenciones de servidores públicos actuantes, estos dejan de manifiesto su identidad a través del uso de la firma hológrafa. En cambio, la correspondencia externa es generada por instituciones públicas relacionadas a gobiernos municipales, por ejemplo: hospitales, policía, unidades educativas, Federación de Juntas Vecinales (FEJUVE) y otras.

Los procedimientos manuales aplicados en la gestión de documentación y correspondencia al utilizar la firma hológrafa en los trámites internos o externos del Gobierno Autónomo Municipal de Entre Ríos, provocan pérdida de tiempo del solicitante, incertidumbre sobre el estado de la documentación, demora en la respuesta de trámites, impiden el flujo normal de procesos, provocando un servicio manual, produciendo desorden, desorganización de la correspondencia y quejas recurrentes entre los funcionarios.

Por otra parte, es importante acotar que en la actualidad, con la implementación de la tecnología firma digital, se pueden agilizar y simplificar los

trámites (Fernández et al., 2020), los procesos administrativos existentes reduciendo el uso del papel al no requerir la impresión de documentos, minimizar el tiempo en mensajerías, brindando mayor seguridad en el envío y recepción de documentos electrónicos firmados digitalmente, transacciones electrónicas, garantizando así, la autenticación de la identidad del firmante, la integridad de la información, el no repudio de la misma.

Con el uso de aplicaciones Web, se pueden automatizar los procesos manuales, el tiempo que se requiere para la tramitación, consultar el estado en el que la correspondencia se encuentra, el monitoreo y procesamiento resulta mucho más fácil y rápido. Además, se tendría información actualizada y fiable de la documentación electrónica.

Particularmente, Bolivia ingresa a la era de la firma digital en base a la Ley N° 164, Ley General de Telecomunicaciones, Tecnologías de Información y Comunicación, de 08 de agosto de 2011 (Ley N° 164, 2011), brinda validez jurídica a la firma digital, la norma estipula que la Autoridad de Regulación y Fiscalización de Telecomunicaciones y Transportes - ATT, se constituye como la "Entidad Certificadora Raíz" y le otorga las atribuciones para autorizar, regular, fiscalizar, supervisar y controlar a las Entidades Certificadoras Públicas y Privadas (ATT, 2016). Es decir, se establecen los puntos que se deben

reglamentar en el uso de documentos digitales y la implementación de la firma digital, y se determina que la Agencia para el Desarrollo de la Sociedad de la Información en Bolivia (ADSIB) se convierte en la Entidad Certificadora Pública, con el fin de marcar un cambio al automatizar, modernizar, transparentar la gestión pública y hacer más eficientes los procesos de las actividades.

En concordancia con lo expuesto, en este estudio se pretende desarrollar un sistema web de correspondencia para la gestión de documentación electrónica aplicando firma digital en los trámites del Gobierno Autónomo Municipal de Entre Ríos, Bolivia.

BASES METODOLÓGICAS

Técnicas de recopilación de información

Entre las técnicas para la recopilación de información seleccionadas se encuentran: la entrevista, porque permite obtener la información de forma rápida como los requerimientos, solicitudes, peticiones y a la vez tener un contacto de frente con el usuario; y la observación, porque puede ser utilizada por el analista para desarrollar los sistemas de información y ayudará a encontrar los principales problemas y necesidades existentes.

Modelado de negocio

El modelado de negocios es de gran ayuda en la etapa de análisis de desarrollo de software, ya que tener un

buen modelo permite lograr comprender el ámbito de la información, además de identificar las actividades y procesos que se realizan dentro de la organización para lograr una correcta operación y así lograr una buena comprensión del negocio para automatizar procesos al crear sistemas computacionales que se ajusten a la medida de una organización (León y Asato, 2009).

Ingeniería de software

La ingeniería de software no sólo se interesa por los procesos técnicos del desarrollo de software, sino que también incluye actividades como la administración del proyecto de software y el desarrollo de herramientas, así como métodos y teorías para apoyar la producción de software (Sommerville, 2011).

Procesos de desarrollo de software

Un proceso de desarrollo de software posee reglas preestablecidas, y deben ser aplicados en la creación del software de mediano y gran porte,

siempre se debe aplicar un modelo de ciclo de vida del desarrollo de software y es una estructura aplicada al desarrollo de software.

Para efectos de esta investigación, se seleccionó la programación extrema (XP), la cual es una metodología ágil de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Usa un enfoque orientado a objetos como paradigma de desarrollo preferido.

La programación extrema (XP) es quizás el método ágil mejor conocido y más ampliamente usado. El nombre lo acuñó Beck (2000) debido a que el enfoque se desarrolló llevando a niveles "extremos" las prácticas reconocidas, como el desarrollo iterativo. Por ejemplo, en la XP muchas versiones actuales de un sistema pueden desarrollarse mediante diferentes programadores, integrarse y ponerse a prueba en un solo día (Sommerville, 2011). En la figura 1 se esquematiza el proceso XP para producir un incremento del sistema por desarrollar.

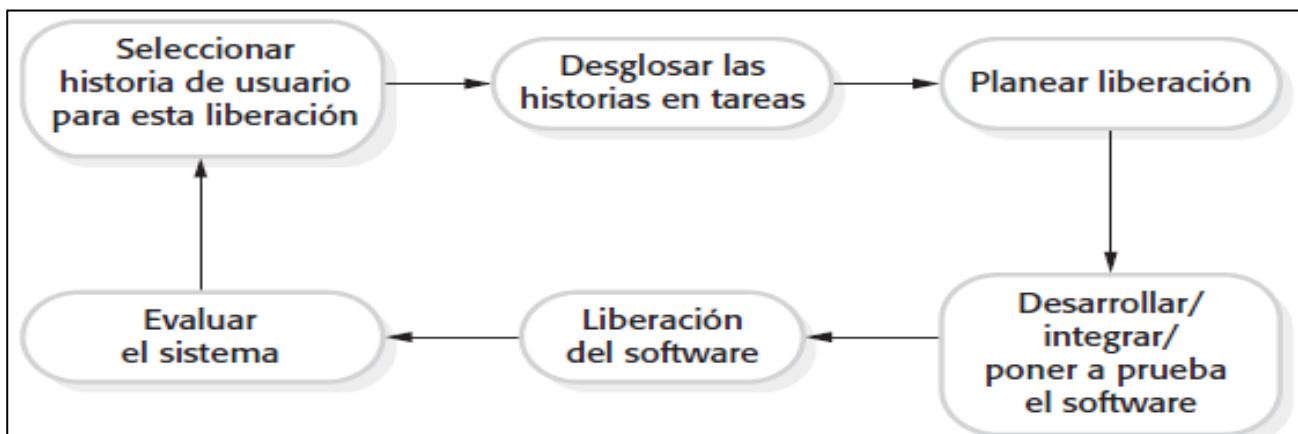


Figura 1. El Ciclo de liberación de la programación extrema. Fuente: Sommerville (2011)

En la programación extrema, los requerimientos se expresan como escenarios (llamados historias de usuario), que se implementan directamente como una serie de tareas. Los programadores trabajan en pares y antes de escribir el código desarrollan pruebas para cada tarea. Todas las pruebas deben ejecutarse con éxito una vez que el nuevo código se integre en el sistema. Entre las liberaciones del sistema existe un breve lapso (Sommerville, 2011).

Según Pressman (2010), cualquier proceso del software ágil se caracteriza por la forma en la que aborda cierto número de suposiciones clave [Fow02] acerca de la mayoría de proyectos de software, de modo que, la programación extrema usa un enfoque orientado a objetos como paradigma preferido de desarrollo, y engloba un conjunto de reglas y prácticas que

ocurren en el contexto de cuatro actividades estructurales: planeación, diseño, codificación y pruebas. En la figura 2 se representa el proceso XP y se resaltan algunas de las ideas y tareas clave que se asocian con cada actividad estructural.

Lenguaje unificado de modelado (UML).

El Lenguaje unificado de modelado (UML) del inglés Unified Modeling Language, desempeña un papel importante en el desarrollo de software, pero también en otros sistemas de muchos sectores de la industria, ya que es un medio de mostrar visualmente el comportamiento y la estructura de un sistema o un proceso. UML ayuda a identificar posibles errores en las estructuras de la aplicación, el comportamiento del sistema u otros procesos empresariales.

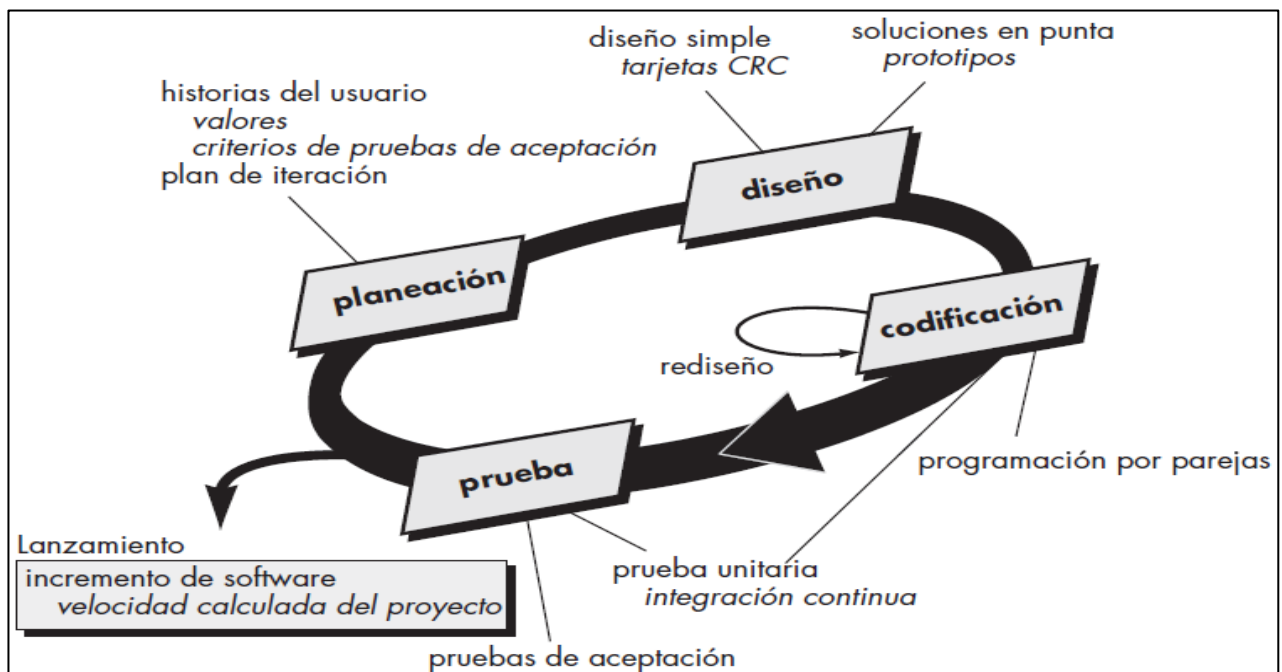


Figura 2. El proceso de la programación extrema. Fuente: Pressman (2010)

Diagrama de Casos de Uso

El diagrama de uso de caso UML ayuda a determinar la funcionalidad y características del software desde la perspectiva del usuario. Los diagramas de casos de uso son el equivalente del arte rupestre moderno. Los símbolos principales de un caso de uso son el actor y el óvalo del caso de uso. Es decir que son responsables principalmente de documentar los macro requisitos del sistema.

Diagrama de clases

Los diagramas de clases describen la estructura estática de un sistema. Una clase es una categoría o grupo de cosas que tienen atributos (propiedades) y acciones similares. Un rectángulo es el símbolo que representa a la clase, y se divide en tres áreas. Un diagrama de clases está formado por varios rectángulos de este tipo conectados por líneas que representan las asociaciones o maneras en que las clases se relacionan entre sí.

Diagrama de colaboración

Un diagrama de colaboración o diagrama de comunicación transmite la misma información que un diagrama de secuencia, en donde el ordenamiento en el tiempo es implícito en la disposición lineal de un diagrama de secuencia, se indica explícitamente el orden en el tiempo numerando los mensajes en los diagramas de colaboración geoméricamente organizados. Los símbolos clave en los diagramas de colaboración son el rectángulo, llamado papel clasificador,

y una línea que indica el mensaje, una vez más llamada conector. El papel clasificador representa los objetos (Kimmel, 2008).

Diagrama de despliegue

Los diagramas de despliegue no son difíciles de crear, en general no contienen un gran número de elementos y sólo se necesitan para aplicaciones de complejidad mediana a grande. Estos diagramas son buenos para la visualización del panorama de su despliegue, para sistemas con múltiples elementos. Desde luego, se es libre para crear un diagrama de despliegue para todo modelo, pero ésta es un área en donde podría economizar (Kimmel, 2008).

Herramientas para el desarrollo de aplicaciones Web

Las aplicaciones web son herramientas que pueden ser utilizadas por los usuarios accediendo mediante un servidor web a través de un navegador que ejecutará la tarea. La programación en la Web ha generado como consecuencia la creación de varias herramientas de desarrollo, por lo que es importante identificar cuáles ofrecen un mejor rendimiento y bajo qué Sistema Operativo.

Lenguajes de programación para el desarrollo del sistema Web

Para el desarrollo de sistemas Web, existen una multitud de lenguajes de programación para el desarrollo de sistemas Web. Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por

máquinas como las computadoras. Además, es un lenguaje diseñado para describir el conjunto de acciones consecutivas que una computadora debe ejecutar. Esto implica que, es un modo práctico para que los seres humanos puedan dar instrucciones a una computadora y está formado por un conjunto de símbolos y reglas sintácticas que definen su estructura. Los lenguajes de programación Web que serán analizados son: PHP vs ASP.NET, bajo los sistemas operativos Linux y Windows utilizando criterios comunes.

Arquitectura

La arquitectura de software son patrones o lineamientos que ayudan a la construcción de un programa. Estos patrones permiten tener una guía para los desarrolladores, analistas y todos los cargos relacionados para lograr cumplir con los requerimientos de la aplicación. La arquitectura del software por tanto define la estructura que debe de tener un software, las piezas que se debe construir y el modo en el que se deben de juntar y trabajar entre ellas. Se define a alto nivel mediante una serie de patrones y abstracciones que seguir para el desarrollo del software y para la interacción entre sus diversas piezas (Baquero, 2020).

Arquitectura Cliente – Servidor

En esta arquitectura la computadora de cada uno de los usuarios, llamada cliente, produce una demanda de información a cualquiera de las computadoras que proporcionan información, conocidas como servidores estos últimos responden a la demanda

del cliente que la produjo. Los clientes y los servidores pueden estar conectados a una red local o una red amplia, como la que se puede implementar en una empresa o a una red mundial como lo es la Internet.

Bajo este modelo cada usuario tiene la libertad de obtener la información que requiera en un momento dado proveniente de una o varias fuentes locales o distantes y de procesarla como según le convenga. Los distintos servidores también pueden intercambiar información dentro de esta arquitectura (EcuRED, 2020).

La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes. Ayuda a comprender las bases sobre las que están contruidos los algoritmos distribuidos. La figura 3 ilustra la arquitectura cliente servidor y resalta algunas de las ideas y tareas clave que se asocian con cada actividad estructural.

Selección de la arquitectura de software

En el presente proyecto se eligió la Arquitectura Cliente servidor. Por qué esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras. La interacción cliente-servidor es el soporte de la mayor parte de la comunicación por redes. Ayuda a comprender las bases sobre las

que están contruidos los algoritmos distribuidos.

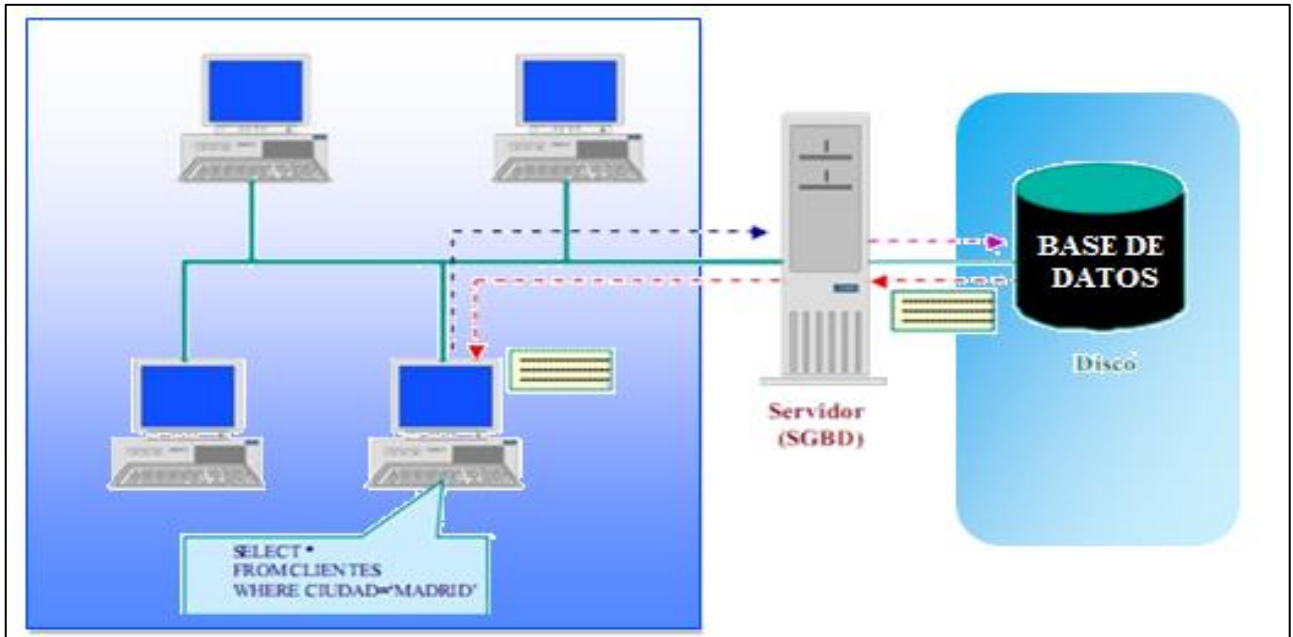


Figura 3. Arquitectura Cliente / Servidor. Fuente: EcuRED (2020)

Servicios Web

Un servicio web es un componente software que se comunica con otras aplicaciones codificando los mensajes en XML y enviándolos a través de protocolos estándares de Internet, tales como Hipertexto Transfer Protocol (HTTP). Básicamente, los servicios web son elementos que les permiten a las aplicaciones comunicarse entre sí, sin importar el lenguaje o plataforma en que se desarrollen. Pueden ser diseñados para soportar la interoperabilidad máquina a máquina a través de una red. El servicio web tiene una interfaz descrita en un formato procesable por máquina. Otros sistemas interactúan con el servicio Web de la manera prescrita por la descripción utilizando mensajes SOAP, normalmente transmitidos mediante HTTP con una serialización XML junto con otros estándares relacionados (Sayago, 2019).

Tecnologías para el desarrollo web

Los lenguajes de programación que existen dirigidos al desarrollo web son de tecnologías frontend y tecnologías backend.

Framework para desarrollo de sistema Web backend

Son las tecnologías que hacen vida en la capa de acceso de datos, están del lado del servidor por lo cual, los usuarios no tienen acceso a ella. Un servidor web es un ordenador donde se alojan aplicaciones/páginas en un espacio de memoria con la finalidad

de dar respuestas a las peticiones de tipo HTTP que hacen los usuarios. Los servidores son los que dan vida al internet.

Las tecnologías back-end con las que se desarrollan sistemas son: Lenguajes de programación web interpretados de lado del servidor, Frameworks (marcos de trabajo) elaborados en base a un lenguaje de programación, Bases de datos y Servidores HTTP.

Framework para desarrollo del sistema Web Frontend

Es la capa visual que permite a los usuarios/clientes la interacción con el sistema, por lo tanto, es la capa que define las experiencias de los usuarios en el sistema y la única a la que tienen acceso. El frontend son todas esas tecnologías que viven en el navegador y con las cuales se crea el diseño del sistema. Cuando se habla del diseño de un sistema se trata de la creación de las interfaces, las propiedades que se pretende que tenga y la interacción de las mismas.

Angular 8

Angular es un marco de diseño de aplicaciones y una plataforma de desarrollo para crear aplicaciones eficientes y sofisticadas de una sola página. Estos documentos de Angular ayudan a aprender y usar el marco Angular y la plataforma de desarrollo, desde su primera aplicación hasta la optimización de aplicaciones complejas de una sola página para empresas. Los tutoriales y guías incluyen ejemplos

descargables para acelerar sus proyectos.

Gestor de base de datos

Un Sistema Gestor de Base de Datos (SGBD), en inglés DataBase Management System (DBMS) es un sistema de software que permite la definición de bases de datos; así como la elección de las estructuras de datos necesarios para el almacenamiento y búsqueda de los datos, ya sea de forma interactiva o a través de un lenguaje de programación. Un SGBD relacional es un modelo de datos que facilita a los usuarios describir los datos que serán almacenados en la base de datos junto con un grupo de operaciones para manejar los datos. Los SGBD relacionales son una herramienta efectiva que permite a varios usuarios acceder a los datos al mismo tiempo. Brindan facilidades eficientes y un grupo de funciones con el objetivo de garantizar la confidencialidad, la calidad, la seguridad y la integridad de los datos que contienen, así como un acceso fácil y eficiente a los mismos.

Entre los SGBD se encuentra el lenguaje de consulta estructurado (MySQL 8.0), el cual es un SGBD multihilo y multiusuario utilizado en la gran parte de las páginas web actuales. Además, es el más usado en aplicaciones creadas como software libre. Se ofrece bajo la GNU GPL, aunque también es posible adquirir una licencia para empresas que quieran incorporarlo en productos privativos. Desde la compra por parte de Oracle se está orientando a este ámbito empresarial. También el

PostgreSQL es un potente sistema de base de datos relacional de objetos de código abierto que usa y amplía el lenguaje SQL, combinado con muchas características que almacenan y escalan de manera segura las cargas de trabajo de datos más complicadas. Los orígenes de PostgreSQL se remontan a 1986 como parte del proyecto POSTGRES en la Universidad de California en Berkeley y tiene más de 30 años de desarrollo activo en la plataforma central. Otro sistema es el SQL Server, cuyas principales características son: Soporte exclusivo por parte de Microsoft; Escalabilidad, estabilidad y seguridad; Posibilidad de cancelar consultas; Potente entorno gráfico de administración que permite utilizar comandos DDL y DML; Aunque es nativo para Windows puede utilizarse desde hace ya un tiempo en otras plataformas como Linux o Docker (The PostgreSQL, 2020).

Selección del gestor de base de datos

Para el presente proyecto se seleccionó un SGBD MySQL 8.0 multihilo y multiusuario utilizado en la gran parte de las páginas web actuales. Además, es el más usado en aplicaciones creadas como software libre. Por sus principales ventajas de este SGB de datos que son: Facilidad de uso y gran rendimiento, Facilidad para instalar y configurar, Soporte multiplataforma y Soporte SSL.

Pruebas y seguridad del software

Las pruebas de software, son aquellas que se diseñan y ejecutan para la obtención de nueva información a fin de establecer si el producto es de calidad (Cem. Kaner, 2008). Las pruebas de software son la investigación empírica y técnica, cuyo propósito se da para facilitar a los interesados información objetiva sobre la calidad del producto o servicio bajo pruebas.

Tipos de pruebas de Software

Las pruebas de software pueden ser funcionales y no funcionales. La automatización de pruebas, tanto funcionales como no funcionales, optimiza los procesos de calidad y reduce significativamente el esfuerzo invertido en la actividad de Testing, gracias a ello, no sólo se cuenta con resultados oportunos y precisos del comportamiento de un producto software con base en requisitos funcionales y no funcionales, si no también es posible construir un framework de pruebas automáticas, que permitan reutilizar en un futuro los componentes generados en los proyectos (Echeverri, Aristizábal y González, 2013).

Pruebas unitarias

Sólo se prueban componentes individuales (clases, funciones, módulos, etc.). Cada componente se prueba de forma independiente. Descubre errores producidos por defectos internos. Los casos de prueba podrán ser obtenidos

a partir de: Código fuente, modelo de datos, diseño de software.

Este tipo de pruebas pueden ser de aceptación, de integración y de regresión. A efectos de esta investigación se seleccionaron las pruebas de aceptación, las cuales comparan el comportamiento del sistema con los requisitos, sin importar como estén especificados. El cliente realiza o especifica tareas típicas para comprobar que se satisfacen sus requisitos o para comprobar que la organización los ha identificado para el mercado al que se destina el software. Esta actividad puede incluir o no a los desarrolladores del sistema. Las pruebas de aceptación son iterativas, donde se pueden aplicar un conjunto de metodologías como la programación extrema (XP).

Selección de prueba del software

Se eligió la prueba de aceptación, por las siguientes razones: son pruebas para aceptar formalmente el software, son las pruebas de sistema del cliente/usuario; es conveniente haber definido los criterios de aceptación verificables de manera previa y consensuada; están enfocadas a demostrar que no se cumplen los requisitos, criterios de aceptación o el contrato; el usuario selecciona casos de prueba concretos para sus pruebas de aceptación, según las prioridades de su negocio; las pruebas se realizarán en el entorno del cliente (real) y se utilizan técnicas de caja negra.

ANÁLISIS FÍSICO Y LÓGICO DE LA APLICACIÓN

En este apartado se enfatiza la descripción de los procesos y las funcionalidades del sistema.

DISEÑO DEL MODELADO PARA EL PROCESO CORRESPONDENCIA

Modelado de negocio actual

En la figura 4 se presenta un fragmento del modelado de negocio actual.

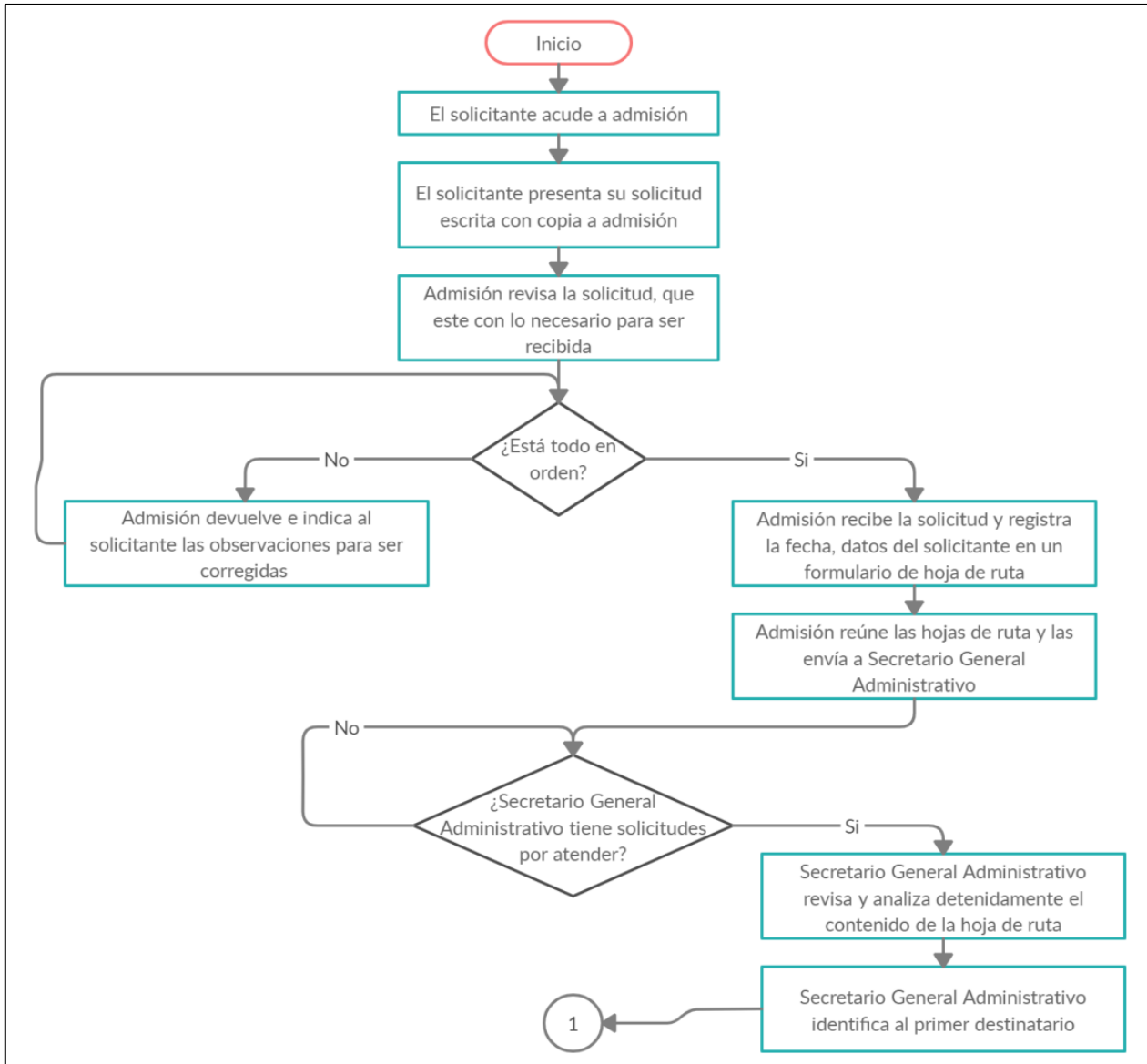


Figura 4. Modelado de negocio actual. Fuente: Elaboración propia

Modelado de negocio alternativo

En la figura 5 se especifican algunos de los aspectos inherentes al modelado de negocio alternativo.

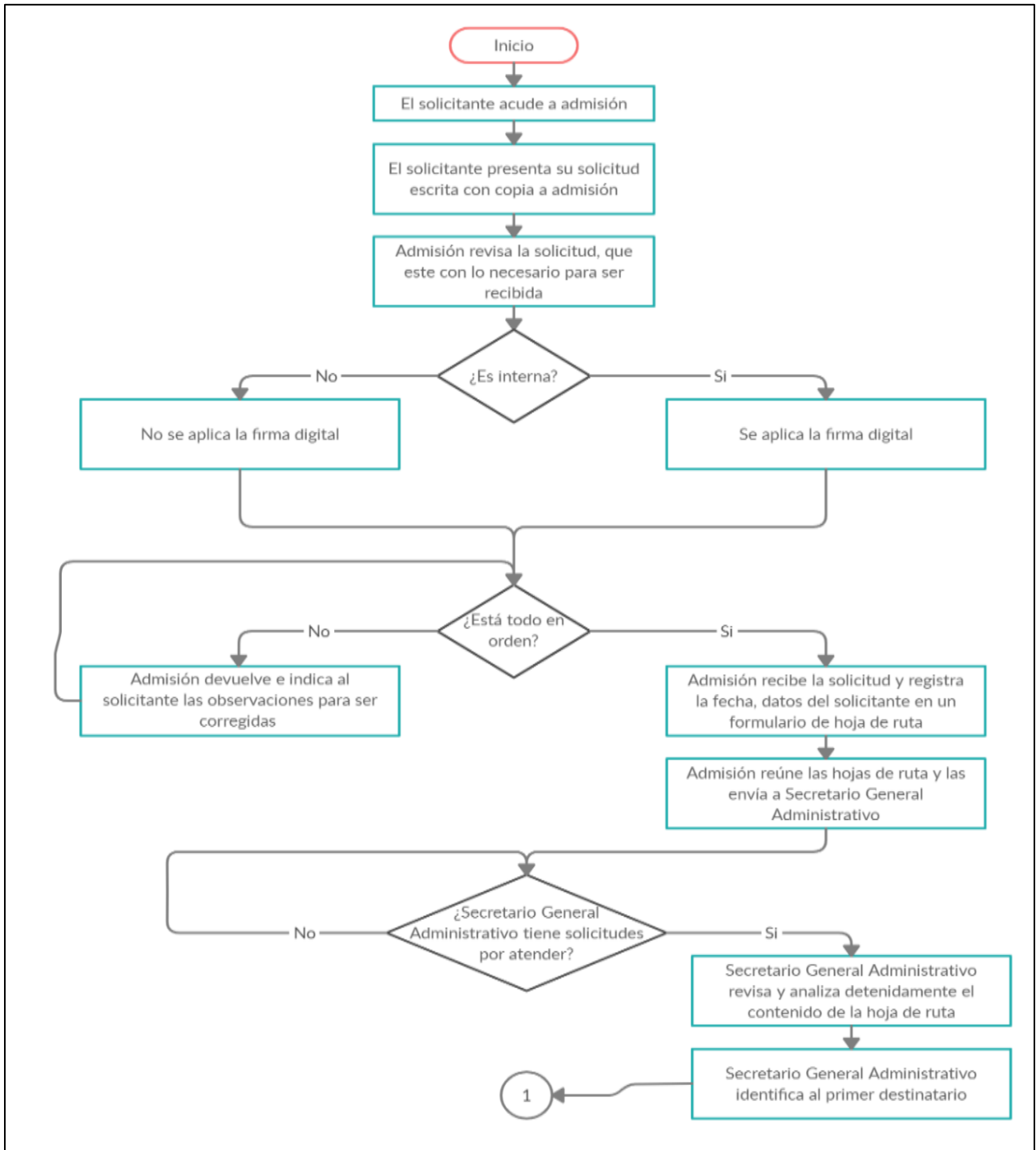


Figura 5. Modelado de negocio alternativo. Fuente: Elaboración propia

Planificación del proyecto

La planificación del proyecto se efectuó mediante la metodología XP.

Identificación de actores

Entre estos se encuentran el administrador o personal de sistemas a cargo de dar asistencia a los demás usuarios; la admisión o personal encargado de recibir las solicitudes y derivarlas a su correspondiente oficina;

el funcionario, es el personal que ocupa un cargo en la institución y atiende las solicitudes presentadas a su oficina; y el solicitante, es decir, la persona que realiza una solicitud esperando que sea atendida.

Historias de usuario

Las historias de usuario se desglosan en la tabla 1.

Tabla 1. Historias de usuario

ID	COMO	NECESITO	PARA
1	Usuario	Iniciar sesión en el sistema, con usuario y contraseña.	Interactuar con el sistema.
2	Administrador	Administrar los usuarios.	Permitir el acceso al sistema al personal involucrado.
3	Administrador	Administrar los cargos.	Registrar los cargos de las unidades que prestan los servicios de atender las solicitudes.
4	Administrador	Administrar las oficinas.	Registrar las oficinas que atenderán las diferentes solicitudes.
5	Administrador	Administrar los tipos de correspondencia.	Registrar los diferentes tipos que hay en el sistema.
6	Administrador	Administrar las instrucciones de correspondencia.	Registrar instrucciones breves y precisas sobre la manera de proceder con la correspondencia en curso.
7	Admisión	Registrar las hojas de ruta en el sistema.	Obtener un código generado por el sistema.
8	Admisión	Realizar el envío de copias de la hoja de ruta.	Permitir que puedan tener conocimiento de la correspondencia.
9	Admisión	Adjuntar archivos en la hoja de ruta.	Adicionar el detalle necesario sobre el proceso.
10	Admisión	Imprimir hoja de ruta.	Respalda la información del sistema.
11	Funcionario	Derivar hoja de ruta.	Enviar el proceso a quien corresponde según el procedimiento.
12	Funcionario	Adjuntar información adicional.	Facilitar el trabajo de la parte correspondiente.

Continuación Tabla 1. Historias de usuario

ID	COMO	NECESITO	PARA
13	Funcionario	Imprimir la hoja de ruta.	Tener una copia de respaldo sobre las autorizaciones realizadas.
14	Funcionario	Ver hojas de ruta por cargo.	Buscar una hoja de ruta de mayor prioridad.
15	Solicitante	Ver estado de la solicitud.	Realizar un seguimiento de la solicitud presentada.
16	Funcionario	Subir archivos adjuntos de las correspondencias.	Respalidar el proceso.
17	Funcionario	Filtrar información de los archivos adjuntos.	Buscar documentos de importancia.
18	Funcionario	Anular los documentos de una hoja de ruta.	Mostrar solo los que tienen importancia.
19	Funcionario	Crear documentos PDF con plantillas.	Adjuntar información en la hoja de ruta.
20	Funcionario	Conexión con API de la ADSIB para configuración de la firma digital.	Acceder a las funcionalidades que ofrece.
21	Funcionario	Firmar documentos PDF al adjuntar a la correspondencia.	Adicionar una firma en un documento de la hoja de ruta.
22	Funcionario	Verificar documento con firma digital.	Ver las firmas en el documento.
23	Funcionario	Ver el buzón con las hojas de ruta derivadas.	Tener la lista de solicitudes que precisan de mi autorización.
24	Funcionario	Ver el detalle de la hoja de ruta.	Tener el conocimiento de lo que se necesita realizar.
25	Funcionario	Concluir hoja de ruta.	Autorizar la solicitud presentada en la hoja de ruta.

Fuente: Elaboración propia.

Plan de entregas

En la tabla 2 se discriminan los elementos constitutivos del plan de entregas del proyecto.

Tabla 2. Estimaciones del plan de entregas

ENTREGABLES	HISTORIAS DE USUARIO		ESTIMACIÓN		
	ID	HISTORIA	ESFUERZO	PRIORIDAD	SEMANAS
Entregable 1: Gestión de correspondencias internas y externas.	1	Como usuario necesito iniciar sesión en el sistema, con usuario, y contraseña para interactuar con el sistema.	3	Alta	
	2	Como administrador necesito administrar los usuarios para permitir el acceso al sistema al personal involucrado.	2	Alta	
	3	Como administrador necesito administrar los cargos para registrar los cargos de las unidades que prestan los servicios de atender las solicitudes.	1	Media	
	4	Como administrador necesito administrar las oficinas para registrar las oficinas que atenderán las diferentes solicitudes.	1	Bajo	
	5	Como administrador necesito administrar los tipos de correspondencia para registrar los diferentes tipos que hay en el sistema.	1	Bajo	
	6	Como administrador necesito administrar las instrucciones de correspondencia para registrar instrucciones breves y precisas sobre la manera de proceder con la correspondencia en curso.	1	Bajo	1
	7	Como Admisión necesito registrar las hojas de ruta en el sistema para obtener un código generado por el sistema.	2	Alto	
	8	Como Admisión necesito realizar el envío de copias de la hoja de ruta para permitir que puedan tener conocimiento de la correspondencia.	2	Medio	
	9	Como Admisión necesito adjuntar archivos en la hoja de ruta para adicionar el detalle necesario sobre el proceso.	2	Medio	
	10	Como Admisión necesito imprimir la hoja de ruta para respaldar la información del sistema.	2	Medio	
	11	Como funcionario necesito derivar la hoja de ruta para enviar el proceso a quien corresponde según el procedimiento.	2	Bajo	

Fuente: Elaboración propia.

Continuación Tabla 2. Estimaciones del plan de entregas

ENTREGABLES	HISTORIAS DE USUARIO		ESTIMACIÓN		
	ID	HISTORIA	ESFUERZO	PRIORIDAD	SEMANAS
Entregable 1: Gestión de correspondencias internas y externas.	12	Como funcionario necesito adjuntar información adicional a la hoja de ruta para adicionar el detalle necesario sobre el proceso.	2	Medio	
	13	Como funcionario necesito imprimir la hoja de ruta para tener una copia de respaldo sobre las autorizaciones realizadas.	1	Medio	1
	14	Como funcionario necesito ver las hojas de ruta por cargo para buscar una hoja de ruta de mayor prioridad.	1	Bajo	
Entregable 2: Seguimiento de correspondencia a través de hojas de ruta	15	Como solicitante necesito ver el estado de la solicitud para realizar un seguimiento de la solicitud presentada.	2	Alto	1
Entregable 3: Administración de documentos y archivos electrónicos	16	Como funcionario necesito subir archivos adjuntos de las correspondencias para respaldar el proceso.	2	Medio	
	17	Como funcionario necesito filtrar información de los documentos y archivos adjuntos para buscar documentos de importancia.	3	Medio	1
	18	Como funcionario necesito anular los documentos de una hoja de ruta para mostrar solo los que tienen importancia.	2	Medio	
Entregable 4: Firma y verificación de documentos digitales.	19	Como funcionario necesito crear documentos PDF para adjuntar información en la hoja de ruta.	1	Bajo	
	20	Como funcionario conectar con la API de la ADSIB para acceder a las funcionalidades que ofrece.	2	Medio	
	21	Como funcionario necesito firmar documentos PDF para adicionar una firma en un documento de la hoja de ruta.	2	Bajo	
	22	Como funcionario necesito verificar documento con firma digital para ver las firmas en el documento.	2	Bajo	1
	23	Como funcionario necesito ver el buzón con las hojas de ruta derivadas para tener la lista de solicitudes que precisan de mi autorización.	2	Medio	
	24	Como funcionario necesito ver el detalle de la hoja de ruta para tener el conocimiento de lo que se necesita realizar.	2	Medio	
	25	Como funcionario necesito concluir una hoja de ruta para autorizar la solicitud presentada en la hoja de ruta.	1	Bajo	

Fuente: Elaboración propia.

Plan de las iteraciones

El plan de entregas del proyecto se detalla en la tabla 3.

Tabla 3. Estimación de tiempo del plan de entregas

ITERACIONES N°	ENTREGABLE N°	DESCRIPCIÓN	DURACIÓN EN SEMANAS	FECHA DE ENTREGA
1	1	Gestión de correspondencias internas y externas.	1	11/05/2020
2	2	Seguimiento de correspondencia a través de hojas de ruta.	1	18/05/2020
3	3	Administración de documentos y archivos electrónicos.	1	01/06/2020
4	4	Firma y verificación de documentos digitales.	1	15/06/2020

Fuente: Elaboración propia.

Relación entre objetivos específicos e iteraciones

La correspondencia entre objetivos específicos iteraciones se muestra en la tabla 4.

Tabla 4. Relación entre objetivos específicos e iteraciones

OBJETIVOS ESPECÍFICOS	ITERACIONES
Desarrollar módulo de correspondencia para la recepción, derivación o cierre de la correspondencia.	Iteración 1. Gestión de correspondencias internas y externas.
Desarrollar módulo seguimiento del estado de un trámite a todas las entidades involucradas en su recepción, procesamiento o devolución.	Iteración 2. Seguimiento de correspondencia a través de hojas de ruta.
Desarrollar módulo de gestión de documentación electrónica.	Iteración 3. Administración de documentos y archivos electrónicos.
Implementar la firma digital en los documentos electrónicos registrados como comunicaciones oficiales, instructivos, solicitudes, realizados por los servidores públicos para agilizar, simplificar los trámites y permitir verificar la integridad, autenticidad del documento.	Iteración 4. Firma y verificación de documentos digitales.

Fuente: Elaboración propia.

DESARROLLO DE MÓDULO DE CORRESPONDENCIA PARA LA RECEPCIÓN, DERIVACIÓN O CIERRE DE LA CORRESPONDENCIA. PRIMERA ITERACIÓN.

La primera iteración comprende la realización del registro de nuevas cuentas de usuario, donde se le permitirá al sujeto autenticarse en el sistema por medio de un usuario y contraseña. Los elementos constitutivos de esta iteración se describen en la tabla 5.

Tabla 5. Primera iteración

ID	HISTORIA	SEMANA	CRITERIOS DE ACEPTACIÓN
1	Como usuario necesito iniciar sesión en el sistema con usuario y contraseña para interactuar con el sistema.		<ul style="list-style-type: none"> Validación de usuario y contraseña. Si es correcto, iniciar sesión, de lo contrario mostrar mensaje. Validación de la entrada de datos de autenticación, usuario y contraseña. Validación de tamaño máximo de caracteres para el nombre de usuario. Alerta, mensaje para usuarios inhabilitados. Listado de los usuarios registrados. Validación de formulario de registro de usuarios, datos requeridos. Validación de nombres de usuario únicos en el sistema. Validación de datos email, ci únicos en el sistema.
2	Como administrador necesito administrar los usuarios para permitir el acceso al sistema al personal involucrado.	1	<ul style="list-style-type: none"> Encriptación de la contraseña en la base de datos. Listado de opciones sobre cada uno de los registros de la lista. Paginado de la lista de usuarios. Datos del usuario editable. Opción para eliminar datos del usuario, solo si fuera necesario. Función para habilitar o deshabilitar las cuentas de usuario.
3	Como administrador necesito administrar los cargos para registrar los cargos de las unidades que prestan los servicios de atender las solicitudes.		<ul style="list-style-type: none"> Listado de los cargos registrados. Validación de formulario de registro de cargos, valores nulos en el nombre. Listado de opciones sobre cada uno de los registros de la lista Registro del cargo editable. Opción para eliminar registro del cargo.

Continuación Tabla 5. Primera iteración

ID	HISTORIA	SEMANA	CRITERIOS DE ACEPTACIÓN
5	Como administrador necesito administrar los tipos de correspondencia para registrar los diferentes tipos que hay en el sistema.		<ul style="list-style-type: none"> Listado de los tipos de correspondencia registrados. Validación de formulario de registro de tipos de correspondencia, valores nulos en el nombre.
6	Como administrador necesito administrar las instrucciones de correspondencia para registrar instrucciones breves y precisas sobre la manera de proceder con la correspondencia en curso.		<ul style="list-style-type: none"> Listado de las instrucciones registrados. Validación de formulario de registro de las instrucciones, valores nulos en el nombre. Listado de opciones sobre cada uno de los registros de la lista. Registro de las instrucciones editable. Opción para eliminar registro de las instrucciones.
7	Como Admisión necesito registrar las hojas de ruta en el sistema para obtener un código generado por el sistema.		<ul style="list-style-type: none"> Validación de formulario de registro, datos requeridos. Listado de las hojas de ruta registradas.
8	Como Admisión necesito realizar el envío de copias de la hoja de ruta para permitir que puedan tener conocimiento de la correspondencia.	1	<ul style="list-style-type: none"> Listado de oficinas registradas. Listado de los cargos de cada oficina seleccionada. Envío de copias de la hoja de ruta destinado a otro cargo.
9	Como Admisión necesito adjuntar archivos en la hoja de ruta para adicionar el detalle necesario sobre el proceso.		<ul style="list-style-type: none"> Directorio para guardar los archivos. Adjuntado de archivos a la hoja de ruta.
10	Como Admisión necesito imprimir la hoja de ruta para respaldar la información del sistema.		<ul style="list-style-type: none"> Impresión del detalle de las hojas de ruta.
11	Como funcionario necesito derivar la hoja de ruta para enviar el proceso a quien corresponde según el procedimiento.		<ul style="list-style-type: none"> Listado de las hojas de ruta en mi buzón. Información del estado de la hoja de ruta. Detalles de la hoja de ruta, interfaz en el sistema. Opción para derivar la hoja de ruta.

Continuación Tabla 5. Primera iteración

ID	HISTORIA	SEMANA	CRITERIOS DE ACEPTACIÓN
12	Como funcionario necesito adjuntar información adicional a la hoja de ruta para adicionar el detalle necesario sobre el proceso.	1	<ul style="list-style-type: none"> Opción para subir archivo e indicar el nombre del archivo. Listado de archivos adjuntos de la hoja de ruta.
13	Como funcionario necesito imprimir la hoja de ruta para tener una copia de respaldo sobre las autorizaciones realizadas.		<ul style="list-style-type: none"> Opción para una imprimir una hoja de ruta. Información del recorrido de la hoja de ruta.
14	Como funcionario necesito ver las hojas de ruta por cargo para buscar una hoja de ruta de mayor prioridad.		<ul style="list-style-type: none"> Reporte de hojas de ruta por cargo.

Fuente: Elaboración propia.

DESARROLLO DE MÓDULO SEGUIMIENTO DEL ESTADO DE UN TRÁMITE A TODAS LAS ENTIDADES INVOLUCRADAS EN SU RECEPCIÓN, PROCESAMIENTO O DEVOLUCIÓN. SEGUNDA ITERACIÓN.

Los aspectos asociados a la segunda iteración se presentan en la tabla 6.

Tabla 6. Segunda iteración

ID	HISTORIA	SEMANA	CRITERIOS DE ACEPTACIÓN
15	Como solicitante necesito ver el estado de la solicitud para realizar un seguimiento de la solicitud presentada.	1	<ul style="list-style-type: none"> Interfaz de consulta, antes de ingresar al sistema. Búsqueda de la hoja de ruta por el número de hoja de ruta asignado y el código. Alerta, mensaje en caso que no se encuentre la hoja de ruta. Alerta, mensaje para hoja de ruta encontrada. Visualización del detalle y estado de la hoja de ruta.

Fuente: Elaboración propia.

DESARROLLO DE MÓDULO DE GESTIÓN DE DOCUMENTACIÓN ELECTRÓNICA. TERCERA ITERACIÓN.

Los componentes inherentes a la tercera iteración se muestran en la tabla 7.

Tabla 7. Tercera iteración

ID	HISTORIA	SEMANA	CRITERIOS DE ACEPTACIÓN
16	Como funcionario necesito subir archivos adjuntos de las correspondencias para respaldar el proceso.		<ul style="list-style-type: none"> • Opción de subir archivos en el detalle de la hoja de ruta.
17	Como funcionario necesito filtrar información de los documentos y archivos adjuntos para buscar documentos de importancia.	1	<ul style="list-style-type: none"> • Filtros de búsqueda para los documentos y archivos. • Información de la fuente del archivo, la hoja de ruta a la que pertenece.
18	Como funcionario necesito anular los documentos de una hoja de ruta para mostrar solo los que tienen importancia.		<ul style="list-style-type: none"> • Opción de anulación de los documentos y archivos de una hoja de ruta.

Fuente: Elaboración propia.

Implementación de firma digital en los documentos electrónicos registrados como comunicaciones oficiales, instructivos, solicitudes por licencia de vacaciones, realizados por los servidores públicos para agilizar, simplificar los trámites y permitir verificar la integridad, autenticidad del documento. Cuarta iteración.

Para esta última iteración, los aspectos asociados a ella se observan en la tabla 8.

Tabla 8. Cuarta iteración

ID	HISTORIA	SEMANA	CRITERIOS DE ACEPTACIÓN
19	Como funcionario necesito crear documentos PDF para adjuntar información en la hoja de ruta.		<ul style="list-style-type: none"> • Conversión el archivo PDF a base 64.
20	Como funcionario conectar con la API de la ADSIB para acceder a las funcionalidades que ofrece.		<ul style="list-style-type: none"> • Instalación del programa para conectar con el servicio de la ADSIB.
21	Como funcionario necesito firmar documentos PDF para adicionar una firma en un documento de la hoja de ruta.		<ul style="list-style-type: none"> • Solicitud para firmar el archivo con los parámetros solicitados. • Opción para usar plantillas.
22	Como funcionario necesito verificar documento con firma digital para ver las firmas en el documento.	1	<ul style="list-style-type: none"> • Respuesta de la API de ADSIB sobre las firmas de un documento.
23	Como funcionario necesito ver el buzón con las hojas de ruta derivadas para tener la lista de solicitudes que precisan de mi autorización.		<ul style="list-style-type: none"> • Listado de hojas de ruta derivadas a mi cargo en el sistema. Opción de solo ver un resumen de las copias de la hoja de ruta.
24	Como funcionario necesito ver el detalle de la hoja de ruta para tener el conocimiento de lo que se necesita realizar.		<ul style="list-style-type: none"> • Listado de las hojas de ruta enviadas a mi cargo. • Opción de ver detalle en las hojas de ruta. • En el detalle, listado de oficinas por donde la hoja de ruta ha sido derivado.
25	Como funcionario necesito concluir una hoja de ruta para autorizar la solicitud presentada en la hoja de ruta.		<ul style="list-style-type: none"> • Opción para concluir la hoja de ruta. • Listado de hojas de ruta en sección concluidos.

Fuente: Elaboración propia.

Los módulos vinculados a las tres primeras iteraciones se desarrollaron en función de los siguientes componentes:

(a) Caso de uso de la iteración. Se representaron los casos de uso para el administrador, la admisión y el funcionario mediante diagramas similares al de la figura 6.

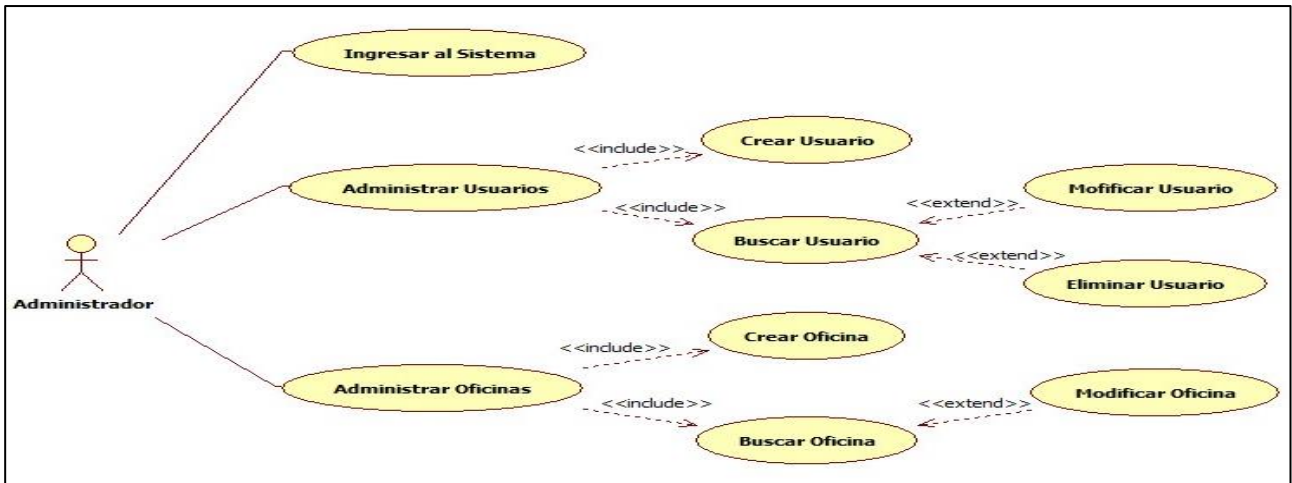


Figura 6. Casos de uso: Administrador Fuente: Elaboración propia.

(b) Diagrama de clases. Se muestra un modelo de este diagrama en la figura 7.

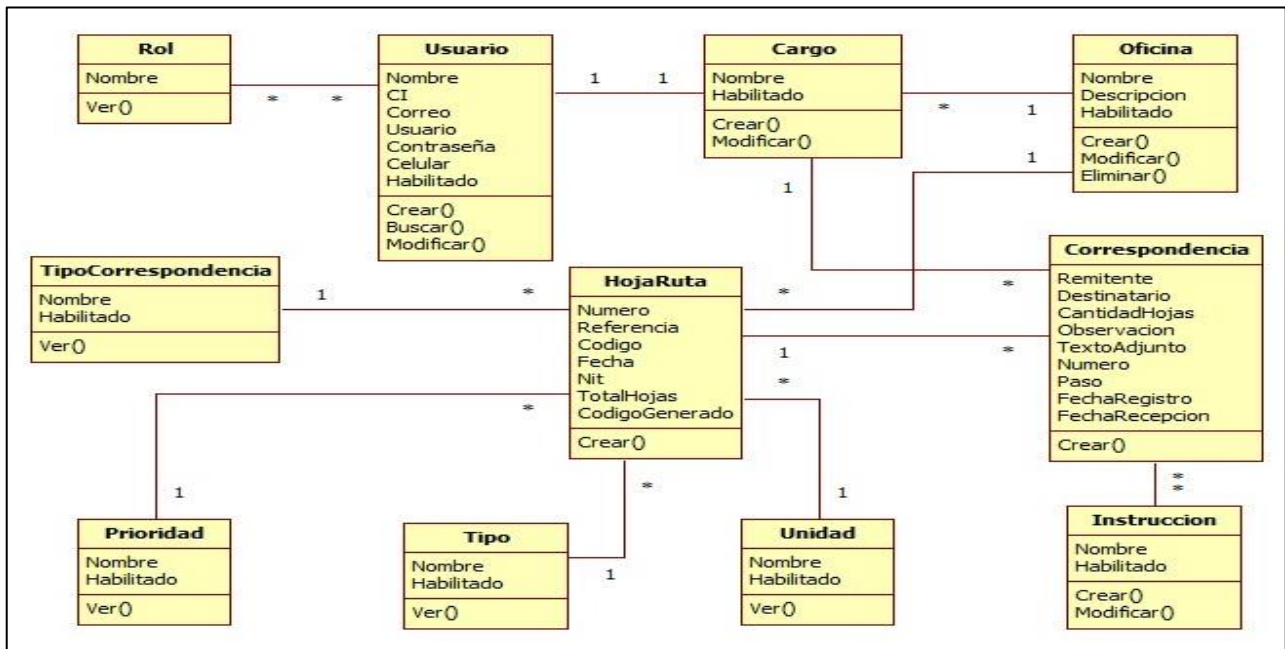


Figura 7. Diagrama de Clases primera iteración. Fuente: Elaboración propia.

(c) **Diagrama de base de datos.** En la figura 8 se puede visualizar uno de los diagramas de base de datos.

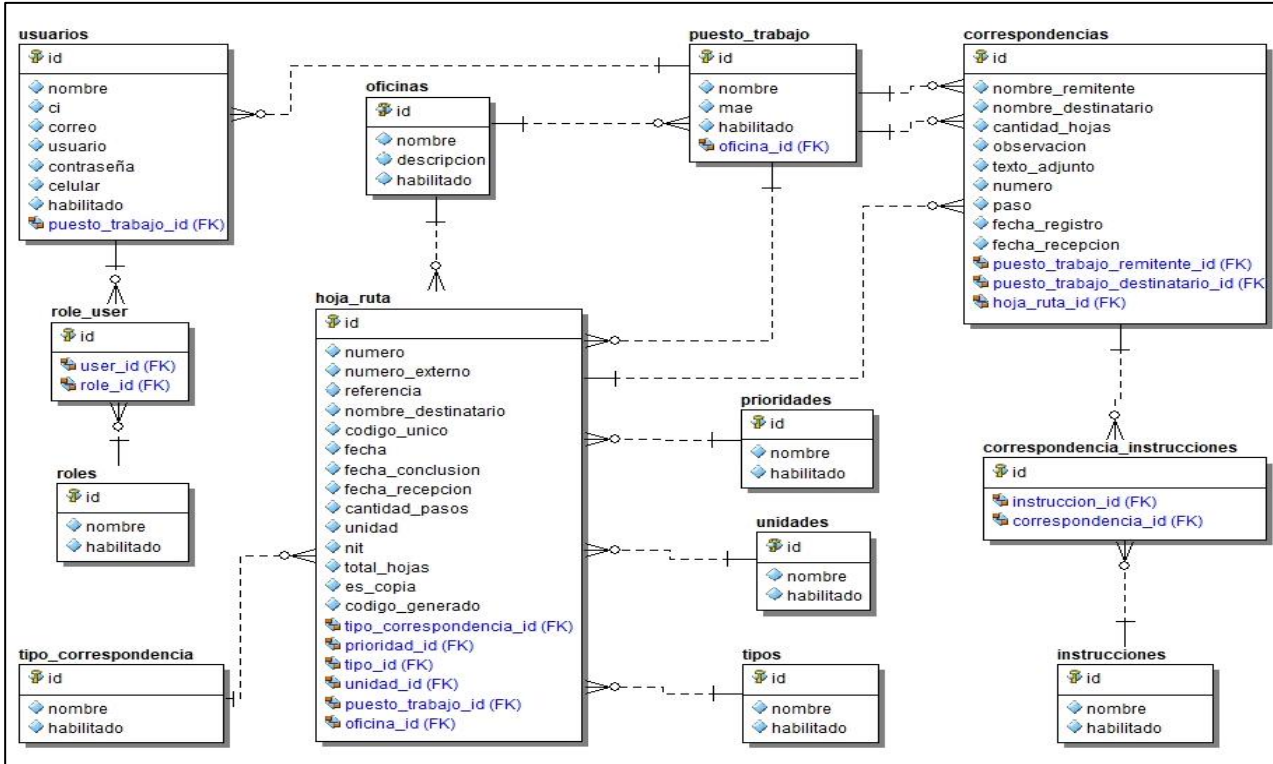


Figura 8. Diagrama de la base de datos primera iteración. Fuente: Elaboración propia.

(d) **Diagrama de arquitectura.** Un diagrama de esta naturaleza se puede observar en la figura 9.

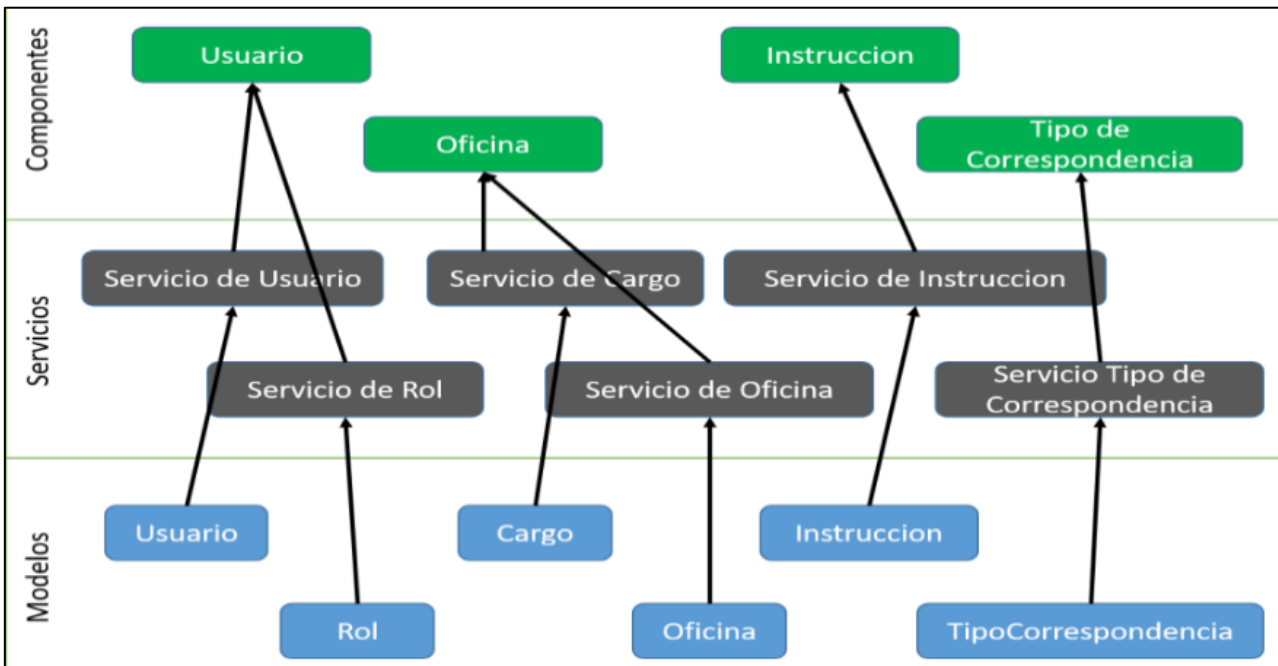


Figura 9. Diagrama de la arquitectura de la primera iteración. Fuente: Elaboración propia

(e) Codificación. En la figura 10 se muestra una de las imágenes referidas a la codificación. En este caso, es el código del servicio que conecta con la API, para realizar la autenticación de un usuario (nombre de usuario y contraseña), en la interfaz de inicio de sesión.

```
@Injectable()
export class Service {
  constructor(private http: HttpClient) {
  }

  login(username: string, password: string) {
    return this.http.post<any>(appConfig.apiUrl + '/token', {username, password})
      .pipe(map(user => {
        if (user && user.token) {
          localStorage.setItem('currentUser', JSON.stringify(user));
        }
        return user;
      }));
  }
}
```

Figura 10. Código para el inicio de sesión. Fuente: Elaboración propia.

(f) Interfaces. A continuación, se presenta una imagen de las diversas interfaces utilizadas (Figura 11).



Figura 11. Formulario de inicio de sesión. Fuente: Elaboración propia.

(g) Pruebas unitarias. En la figura 12 se puede detallar la imagen correspondiente a una de las pruebas unitarias realizadas.

```

1  public function test_verificar_existencia_de_usuario() {
2      $this->assertDatabaseHas('users',[
3          'username' => 'usertest'
4      ]);
5
6  }

```

Figura 12. Prueba unitaria verificando la existencia de un usuario. Fuente: Elaboración propia.

(h) Pruebas de aceptación. En la tabla 9 se describe lo relativo a las pruebas de aceptación de una de las iteraciones.

Tabla 9. Resultado de las pruebas de aceptación

ID	HISTORIA DE USUARIO	RESULTADOS ESPERADOS	RESULTADOS OBTENIDOS
19	Como funcionario necesito crear documentos PDF para adjuntar información en la hoja de ruta.	Mostrar opción de creación de documentos PDF.	Satisfactorio
20	Como funcionario conectar con la API de la ADSIB para acceder a las funcionalidades que ofrece.	Enviar el pin por Jacobitus. Conectar con las funciones que ofrece la ADSIB.	Satisfactorio Satisfactorio
21	Como funcionario necesito firmar documentos PDF para adicionar una firma en un documento de la hoja de ruta.	Firmar un documento PDF enviando los parámetros requeridos por la api. Convertir el documento PDF a base 64.	Satisfactorio Satisfactorio
22	Como funcionario necesito verificar documento con firma digital para ver las firmas en el documento.	Enviar los parámetros y el documento a verificar, y recibir como respuesta la información sobre la firma.	Satisfactorio
23	Como funcionario necesito ver el buzón con las hojas de ruta derivadas para tener la lista de solicitudes que precisan de mi autorización.	Mostrar las hojas de ruta derivadas a mi cargo.	Satisfactorio
24	Como funcionario necesito ver el detalle de la hoja de ruta para tener el conocimiento de lo que se necesita realizar.	Seleccionar una hoja de ruta y ver el detalle. Mostrar en el detalle los archivos adjuntos.	Satisfactorio Satisfactorio
25	Como funcionario necesito concluir una hoja de ruta para autorizar la solicitud presentada en la hoja de ruta.	Ver el estado de la hoja de ruta. Mostrar la opción para concluir la hoja de ruta. Revisar que el estado de la hoja de ruta ha cambiado tras su	Satisfactorio Satisfactorio

conclusión.

Fuente: Elaboración propia.

Con respecto a la implementación de firma digital en los documentos electrónicos registrados vinculada a la iteración 4, referida a la firma y verificación de documentos digitales, solo se contemplaron los elementos de: (a) Codificación, (b) Interfaces y (c) Pruebas de aceptación.

CONCLUSIONES

El sistema web de correspondencia para la gestión de documentación electrónica en los trámites del Gobierno Autónomo Municipal de Entre Ríos, en colaboración con la ADSIB, provee un programa para validar el uso de las firmas y con el registro apropiado. El sistema accede a las funciones para firmar y validar las firmas en un documento PDF. Para validar la firma, se acompaña el Certificado Digital del titular que contiene su respectiva clave pública y sus datos de identidad.

REFERENCIAS BIBLIOGRÁFICAS

- ATT. (2016). Bolivia Ingresa A La Era De La Firma Digital. La Paz – Bolivia. Disponible en: <https://www.att.gob.bo/content/bolivia-ingresa-la-era-de-la-firma-digital>
- Baquero, J. (2020). *Arquitectura del Software*. Disponible en: <https://www.arsys.es/blog/arquitectura-software/>
- Cem. Kaner. (2008). *Pruebas de Software Como Ciencia Social*. Disponible en: <http://www.kaner.com/pdfs/KanerSocialScienceSTEP.pdf>
- Echeverri, J., Aristizábal, M. y González, L. (2013). *Reflexiones sobre ingeniería de requisitos y pruebas de software*. Medellín, Colombia: Corporación Universitaria Remington. Disponible en: <https://elibro.net/es/ereader/biblioteca/cauab/68913?page=67>
- EcuRED (2020). *Arquitectura Cliente Servidor*. Disponible en: https://www.ecured.cu/Arquitectura_Cliente_Servidor
- Fernández, E. , Gutiérrez, J., ; Delgado, R.; Lopez, R. (2020). Aplicación web para la gestión de diplomas digitales en centros de capacitación mediante firma electrónica y blockchain *Revista Ibérica de Sistemas e Tecnologías de Información*; Lousada, E28: 498-509. Disponible en: <https://search.proquest.com/openview/e3b85a7260c758fd48b22ce5c52f0ced/1?pq-origsite=gscholar&cbl=1006393>
- Kimmel, P. (2008). *Manual de UML*. México. Disponible en https://www.academia.edu/13285619/Manual_de_UML
- León y Asato. (2009). **Modelado de negocios programa de desarrollo**. 4 edición, España.
- Ley N° 164. (2011). Ley General de Telecomunicaciones, Tecnologías de Información y Comunicación. Disponible en: https://www.minedu.gob.bo/files/documentos-normativos/leyes/ley_164__ley_general_de_telecomunicaciones_tecnolo

- gias_de_informacin_y_comunicacion.pdf
- Pressman, R. (2010). **Ingeniería del Software**. México.
- Sayago, P. (2019). *Análisis Comparativo entre los Estándares Orientados a Servicios Web SOAP, REST y GraphQL*. Disponible en: https://www.researchgate.net/publication/339660087_Analisis_Comparativ
- o_entre_los_Estandares_Orientados_a_Servicios_Web_SOAP_REST_y_GRAPHQL
- Sommerville, I. (2011). **Ingeniería de Software**. México.
- The PostgreSQL (2020). *Los gestores de Bases de Datos*. Disponible en: <https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/>