



UNIVERSIDAD
CATÓLICA
DE CUENCA

UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

**UNIDAD ACADÉMICA DE INFORMÁTICA,
CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN
TECNOLÓGICA**

**CARRERA DE INGENIERÍA EN SISTEMAS DE
INFORMACIÓN**

**PROPUESTA Y DESARROLLO DE UN PROTOTIPO PARA EL
REGISTRO DE PAGOS EN LA JUNTA DE AGUA MONSEÑOR
“LEÓNIDAS PROAÑO” DE LA COMUNIDAD DE GUARTANCHÚN
UTILIZANDO ALMACENAMIENTO EN LA NUBE**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS DE INFORMACIÓN**

AUTOR: KEVIN RAÚL GUAMÁN ARENILLAS

DIRECTOR: ING. ANDRES SEBASTIÁN QUEVEDO SACOTO

AZOGUES-ECUADOR

2023

DIOS, PATRIA, CULTURA Y DESARROLLO



UNIVERSIDAD CATÓLICA DE CUENCA

Comunidad Educativa al Servicio del Pueblo

UNIDAD ACADÉMICA DE INFORMÁTICA, CIENCIAS DE LA COMPUTACIÓN E INNOVACIÓN
TECNOLÓGICA

CARRERA DE INGENIERÍA EN SISTEMAS DE INFORMACIÓN

**PROPUESTA Y DESARROLLO DE UN PROTOTIPO PARA EL REGISTRO DE PAGOS EN
LA JUNTA DE AGUA MONSEÑOR “LEÓNIDAS PROAÑO” DE LA COMUNIDAD DE
GUARTANCHÚN UTILIZANDO ALMACENAMIENTO EN LA NUBE**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE INGENIERO EN SISTEMAS DE INFORMACIÓN**

AUTOR: KEVIN RAÚL GUAMÁN ARENILLAS

DIRECTOR: ING. ANDRES SEBASTIÁN QUEVEDO SACOTO

AZOGUES-ECUADOR

2023

DIOS, PATRIA, CULTURA Y DESARROLLO



Declaratoria de Autoría y Responsabilidad

Kevin Raúl Guamán Arenillas portador de la cédula de ciudadanía N° 0302366281. Declaro ser el autor de la obra: **"Propuesta y desarrollo de un prototipo para el registro de pagos en la Junta de Agua Monseñor "Leónidas Proaño"**, sobre la cual me hago responsable sobre las opiniones, versiones e ideas expresadas. Declaro que la misma ha sido elaborada respetando los derechos de propiedad intelectual de terceros y eximo a la Universidad Católica de Cuenca sobre cualquier reclamación que pudiera existir al respecto. Declaro finalmente que mi obra ha sido realizada cumpliendo con todos los requisitos legales, éticos y bioéticos de investigación, que la misma no incumple con la normativa nacional e internacional en el área específica de investigación, sobre la que también me responsabilizo y eximo a la Universidad Católica de Cuenca de toda reclamación al respecto.

Azogues, **21 de marzo de 2023**

F: 

Kevin Raúl Guamán Arenillas

C.I. 0302366281

UNIDAD ACADÉMICA DE TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIÓN

CERTIFICACIÓN DEL DIRECTOR DE TESIS

Ing. Paúl Chérrez Vintimilla. Mgs.

DIRECTOR DE LA CARRERA DE INGENIERIA EN SISTEMAS DE INFORMACIÓN

De mi consideración:

Certifico que el presente trabajo de titulación denominado: "PROPUESTA Y DESARROLLO DE UN PROTOTIPO PARA EL REGISTRO DE PAGOS EN LA JUNTA DE AGUA MONSEÑOR "LEÓNIDAS PROAÑO" DE LA COMUNIDAD DE GUARTANCHÚN UTILIZANDO ALMACENAMIENTO EN LA NUBE", realizado por: Guamán Arenillas Kevin Raúl, con cédula de identidad: 0302366281, previo a la obtención del título de INGENIERO EN SISTEMAS DE INFORMACIÓN ha sido asesorado, orientado, revisado y supervisado durante su ejecución, bajo la tutoría del Ing. Sebastián Quevedo Sacoto en todo el proceso. Ante la ausencia del tutor en el país por su estancia doctoral y por su informe de aprobación, me permito certificar que el presente documento, fue desarrollado siguiendo los parámetros del método científico, se sujeta a las normas éticas de investigación que exige la Universidad Católica de Cuenca, por lo que está expedito para su presentación y sustentación ante el respectivo tribunal.

Azogues, 20 de marzo de 2023



Ing. Paúl Chérrez Vintimilla. Mgs
DIRECTOR DE CARRERA

DEDICATORIA

Dedico este trabajo a mis padres y a mis hermanos, quienes me han apoyado incondicionalmente durante todo mi camino académico. Gracias por ser mi fuente de motivación constante y por creer en mí incluso cuando yo mismo dudaba. Este logro no habría sido posible sin su amor y apoyo inquebrantable.

También quiero dedicar este trabajo a mis amigos y seres queridos, quienes me han brindado su apoyo emocional y me han acompañado en los momentos difíciles. Gracias por estar siempre ahí, por sus palabras de aliento y por creer en mí.

Por último, quiero dedicar este trabajo a todas las personas que han contribuido de alguna manera a mi formación académica, desde mis profesores y compañeros de clase hasta aquellos que me han inspirado en mi campo de estudio. Gracias por compartir su conocimiento y por ayudarme a crecer como persona y como profesional.

AGRADECIMIENTO

Quiero expresar mi más sincero agradecimiento a todas las personas que han contribuido de alguna manera a la realización de este trabajo.

En primer lugar, agradezco a Dios por darme la fuerza, la sabiduría y la confianza necesaria para la realización de este proyecto.

Seguidamente, quiero agradecer a mi tutor y a mi cotutora de tesis, por su orientación, apoyo y paciencia durante todo el proceso de investigación. Sus consejos y sugerencias han sido invaluable y me han ayudado a mejorar mi trabajo de manera significativa.

También quiero agradecer a la Universidad Católica de Cuenca, por brindarme la oportunidad de realizar este estudio y por proveerme de los recursos necesarios para llevarlo a cabo. Agradezco a la administración de la Junta de Agua Monseñor “Leónidas Proaño” por su constante apoyo y por permitirme hacer uso de su información.

Agradezco a mi familia y amigos por su amor, apoyo y comprensión durante todo este tiempo. Sus palabras de aliento y su presencia han sido fundamentales para mantenerme motivado y enfocado en mi trabajo.

Finalmente, quiero agradecer a todas las personas que participaron en este estudio, especialmente a los participantes que dedicaron su tiempo y energía para hacer posible este trabajo. Su colaboración y disposición fueron esenciales para la recolección de datos y para el éxito de esta investigación.

Gracias a todos por ser parte de este logro.

Resumen

El siguiente trabajo se desarrolla en la Junta de Agua Potable de comunidad de Guartanchun, donde se busca analizar y resolver el problema de facturación y registro de datos; debido a que este proceso se ha vuelto cada vez más complejo en los últimos años, porque implica una mayor administración.

En base a esa problemática, se desarrolló un prototipo para los trabajos de cobro y facturación, de esa manera, agilizar los diferentes procesos que maneja la Junta de Agua utilizando recursos de la nube.

Para la creación del prototipo se utilizó metodología scrum y se establecieron 4 Sprints con diferentes tareas a cumplir, estas tareas se establecieron con la administración de la Junta de Agua junto con una fecha límite para cada sprint, que se realizarían hasta lograr que el prototipo esté en capacidad de operar con los requerimientos convenidos.

Como resultado el prototipo es capaz de gestionar el proceso de recaudación y facturación del servicio permitiendo registrar la información de los socios, medidores, alcantarillados, entre otros. Estas funcionalidades fueron sometidas a la opinión de los usuarios con el objeto de realizar ajustes esenciales; sin embargo, el despliegue de este producto de software requerirá cumplir con otros aspectos de seguridad, proceso de facturación electrónica entre otros.

Palabras clave: prototipo, sistema, facturación, Cloud Computing

Abstract

This research was conducted in The Division of Drinking Water in the Guartanchun community to analyze and solve the billing and data registration problems; this process has become increasingly complex in recent years because it involves improving administration.

Based on this problem, a prototype was developed for charging and billing operations, thus, streamlining the different processes managed by the Water Council using cloud resources.

Scrum methodology was used for the prototype creation, and 4 Sprints were established with different tasks to accomplish; these tasks were set with the Water Council administration, including a deadline for each sprint, which would be performed until the prototype could operate with the agreed requirements.

As a result, the prototype can manage the service, collection, and billing processes, permitting the registration of members' information, meters, and sewers, among others. These functionalities were submitted to user feedback to make essential adjustments; however, deploying this software product will require compliance with other aspects of the security electronic billing process.

Keywords: prototype, system, billing, Cloud Computing

TABLA DE CONTENIDO

CAPITULO 1. Introducción.....	1
1.1 Planteamiento Del Problema	1
1.2 Objetivos	2
1.2.1 Objetivo General.....	2
1.2.2 Objetivos Específicos	2
1.3 Justificación.....	2
1.4 Alcance.....	3
1.5 Conceptos Relacionados.....	4
1.6 Trabajos Relacionados.....	5
1.7 Metodología	6
CAPITULO 2. Fundamentación Teórica.....	8
2.1 Marco Teórico	8
2.2 Metodologías para el desarrollo de Software	8
2.3 Metodología SCRUM.....	9
2.3.1 Elementos de la Metodología SCRUM.....	9
2.3.2 ¿Por qué implementar SCRUM?.....	9
2.4 HTML	10
2.4.1 Etiquetas HTML	10
2.4.2 HTML 5.....	10
2.5 JAVA	11
2.6 Framework	11
2.6.1 Definición general	11
2.6.2 Framework Web	11
2.7 Desarrollo de Aplicaciones Web	12
2.8 Editores de Código	13
2.9 Visual Studio Code.....	13
2.10 Frontend	13
2.11 Angular.....	15
2.11.1 Arquitectura de Angular	16
2.12 Backend.....	17
2.13 Spring Boot	17
2.14 Spring Framework.....	18
2.14.1 Módulos de Spring Framework	18
2.14.2 ¿Por qué utilizar Spring Framework?	19

2.15	Intellij Idea	19
2.16	Servicios REST	20
2.17	Lombok	20
2.17.1	Instalación	21
2.17.2	Anotaciones Lombok.....	21
2.18	Maven.....	22
2.19	JAVA EE.....	22
2.20	Cloud Computing	22
2.20.1	Modelos de Cloud Computing.....	24
2.20.2	Tipos de Nube.....	25
2.21	Amazon Web Services (AWS)	27
2.22	PostgreSQL	28
2.23	StarUML	29
CAPITULO 3. Propuesta		30
3.1	Análisis de requerimientos	30
3.2	Implementación de la Metodología Scrum	31
3.3	Definición de los Sprints	31
3.4	Planificación de los Sprints	33
3.5	Sprint 1	35
3.5.1	Eventos del Sprint 1.....	35
3.5.2	Roles del Sprint 1.	35
3.5.3	Artefactos del Sprint 1.	35
3.5.4	Sprint Backlog 1	35
3.5.5	BurnDown Chart Sprint 1	37
3.5.6	Justificación del sprint 1	37
3.5.7	Entregable Sprint 1	40
3.6	Sprint 2.....	42
3.6.1	Eventos del Sprint 2.....	42
3.6.2	Roles del Sprint 2.	42
3.6.3	Artefactos del Sprint 2.	42
3.6.4	Sprint Backlog 2.....	43
3.6.5	BurnDown Chart Sprint 2.....	44
3.6.6	Justificación del sprint 2	45
3.6.7	Entregable Sprint 2	47
3.7	Sprint 3.....	49

3.7.1	Eventos del Sprint 3.....	49
3.7.2	Roles del Sprint 3.	49
3.7.3	Artefactos del Sprint 3.	49
3.7.4	Sprint backlog 3.....	49
3.7.5	BurnDown Chart Sprint 3.....	50
3.7.6	Justificación del sprint 3.....	51
3.7.7	Entregable Sprint 3.....	52
3.8	Sprint 4.....	55
3.8.1	Eventos del Sprint 4.....	55
3.8.2	Roles del Sprint 4.	55
3.8.3	Artefactos del Sprint 4.....	55
3.8.4	Sprint Backlog 4.....	55
3.8.5	BurnDown Chart Sprint 4.....	56
3.8.6	Justificación del sprint 4.....	57
3.8.7	Entregable Sprint 4.....	61
3.9	Pruebas del prototipo.....	62
CAPITULO 4. RESULTADOS OBTENIDOS.....		64
4.1	Resultados.....	64
4.2	Conclusiones.....	65
4.3	Recomendaciones.....	67
CAPITULO 5. Anexos.....		72

Índice de Tablas

Tabla I Comparativa de metodologías de desarrollo de Software.....	8
Tabla II Comparativa de Editores de Código	13
Tabla III Comparativas de Framework para el desarrollo de la interfaz	14
Tabla IV Comparativas de Framework para el desarrollo del Backend.....	17
Tabla V Uso de Métodos CRUD en Colecciones	20
Tabla VI Uso de Métodos CRUD Entidades	20
Tabla VII Comparativa de Servicios de la Nube	26
Tabla VIII Product Backlog/ Pila del producto Inicial	31
Tabla IX Historias de Usuario 1 y 2	32
Tabla X Historias de Usuario 3 y 4	32
Tabla XI Historias de Usuario 5 y 6	32
Tabla XII Historias de Usuario 7 y 8.....	32
Tabla XIII Sprint 1	33
Tabla XIV Sprint 2.....	33
Tabla XV Sprint 3	34
Tabla XVI Sprint 4.....	34
Tabla XVII Planificación del Sprint 1	35
Tabla XVIII Lista de tareas del Sprint 1	36
Tabla XIX Lista de Tareas completadas del Sprint 1	36
Tabla XX Planificación del Sprint 2.....	43
Tabla XXI Lista de tareas del Sprint 2	43
Tabla XXII Lista de Tareas completadas del Sprint 2	43
Tabla XXIII Planificación del Sprint 3.....	49
Tabla XXIV Lista de tareas del Sprint 3.....	50
Tabla XXV Lista de Tareas completadas del Sprint 3	50
Tabla XXVI Planificación del Sprint 4	56
Tabla XXVII Lista de tareas del Sprint 4	56
Tabla XXVIII Lista de Tareas completadas del Sprint 4	56

Índice de Figuras

Fig. 1 Arquitectura Angular	16
Fig. 2 Módulos de Spring Framework.....	19
Fig. 3 Dependencia Lombok	21
Fig. 4 Cloud Computing.....	23
Fig. 5 Productos de Amazon Web Service	27
Fig. 6 Burn down chart Sprint 1	37
Fig. 7 Diseños de la base de datos	38
Fig. 8 Generación de la clase Socio.....	38
Fig. 9 Generación de la clase Comunidad	39
Fig. 10 Generación de la clase Alcantarillado	39
Fig. 11 Generación de la clase medidores	40
Fig. 12 Demostración de la interfaz principal.....	40
Fig. 13 Demostración de la ficha de Socio	41
Fig. 14 Demostración de la ficha de Socio Comunidad.....	41
Fig. 15 Demostración de la ficha de Alcantarillado.....	41
Fig. 16 Demostración de la ficha de Medidor.....	42
Fig. 17 Burn down chart Sprint 2	44
Fig. 18 Generación de la clase Tipo Pendiente.....	45
Fig. 19 Generación de la clase Pendiente	46
Fig. 20 Generación de la clase Lectura.....	46
Fig. 21 Generación de la clase ConsumoAgua	47
Fig. 22 Demostración de la ficha de TipoPendiente	47
Fig. 23 Demostración de la ficha de Pendiente.....	48
Fig. 24 Demostración de la ficha Lectura.....	48
Fig. 25 Burn down chart Sprint 3	51
Fig. 26 Generación de la clase Factura Cabecera	52
Fig. 27 Generación de la clase Factura Detalle.....	52
Fig. 28 Demostración del ingreso de un socio	53
Fig. 29 Demostración de ingreso de datos en la ficha de Lectura	53
Fig. 30 Demostración de resultados de pendientes de un Usuario	53
Fig. 31 Demostración de la ficha de Factura de un Usuario	54
Fig. 32 Demostración de la ficha de Reportes	55

Fig. 33 Burn down chart Sprint 4	57
Fig. 34 Creación de la cuneta en aws	58
Fig. 35 Creación de una Instancia en AWS	58
Fig. 36 Configuración de la Instancia	59
Fig. 37 Compilación del Frontend	59
Fig. 38 Compilación del Backend	60
Fig. 39 Carga de los archivos compilados	60
Fig. 40 Pruebas de los archivos compilados	61
Fig. 41 Levantamiento del Servicio en la Nube	61
Fig. 42 Resultados del servicio Levantado	62

CAPITULO 1. Introducción

1.1 Planteamiento Del Problema

El presente proyecto fue desarrollado en la comunidad de Guartanchun, perteneciente a la parroquia Guapán que se ubica al norte, aproximadamente a 4Km de la ciudad de Azogues. Al igual que otras comunidades, Guartanchun tiene su propia junta de agua potable y alcantarillado, la cual se encarga de brindar el servicio de agua potable y alcantarillado a sus habitantes. Actualmente, la Junta de Agua de Guartanchun cuenta con 129 usuarios o socios a los que se prestan los servicios.

En los sectores urbanos la prestación del servicio de agua potable está a cargo de empresas públicas municipales, específicamente en Azogues, esto le corresponde a la Empresa Municipal de Agua Potable y Alcantarillado (EMAPAL). Para prestar sus servicios poseen tecnología que agiliza los procesos y mejoran la prestación del servicio a la comunidad.

El uso de la tecnología en los últimos años se ha convertido en la piedra angular de la EMAPAL, debido a que permite tener una mejor comunicación con los usuarios, planificar y ejecutar los proyectos de manera más ágil, administrar recursos de forma eficiente y reducir errores. Sin embargo, en las zonas rurales (Guartanchun) ocurre lo contrario; a pesar de que el uso de la tecnología fue creciendo durante los últimos años, varias de sus actividades siguen siendo ineficientes, la infraestructura es insuficiente y obsoleta. En esta Junta de Agua Potable, quienes se encargan de receptor los pagos de servicios por parte de los usuarios y de realizar la facturación; deben buscar los datos de cada usuario y el consumo del servicio en una libreta de facturas mensuales, receptor el pago y registrarlo en una hoja electrónica; lo que ha llevado a cometer imprecisiones, dificultad en localizar la información, ralentización del e incluso el riesgo de pérdidas económicas.

En base a esta situación se ha optado por realizar un prototipo de un sistema de facturación para realizar los procesos de cobros y generación de informes de la Junta de Agua Monseñor “Leónidas Proaño” de la comunidad Guartanchun.

1.2 Objetivos

1.2.1 Objetivo General

Desarrollar un prototipo de un sistema de recaudación de agua potable y alcantarillado para la Junta de Agua Potable Monseñor “Leónidas Proaño” de la comunidad Guartanchun, a través de recursos de la nube.

1.2.2 Objetivos Específicos

- Realizar la fundamentación teórica
- Desarrollar el prototipo del software mediante la metodología Scrum.
- Probar el prototipo del software mediante la técnica de pruebas de aceptación

1.3 Justificación

De acuerdo con un estudio realizado por el Instituto Nacional De Estadísticas y Censos (INEC), en la actualidad Ecuador cuenta con una población de alrededor de 18 millones de habitantes, donde el 40% aproximadamente vive en áreas rurales o a los alrededores de áreas urbanas; en estos sectores es difícil el acceso a los servicios básicos controlados por entidades como municipios o gobiernos, siendo responsabilidad de la propia comunidad, llevar el control sobre el uso y consumo de los servicios. [26]

Comúnmente, en los sectores rurales se tiene desconocimiento de los beneficios que la tecnología puede brindar para la solución de problemas, a ello se suma que la falta de recursos económicos y financiamiento impiden que se opte por soluciones apoyadas por la tecnología.

Con el pasar de los años la población de las áreas rurales va en aumento y seguir utilizando formas tradicionales para guardar la información puede tener consecuencias como la pérdida de información, difícil accesibilidad y análisis de casos.

Las instituciones parroquiales al tener la responsabilidad de administrar los servicios que otorgan a sus usuarios se ven obligadas a buscar formas de realizar los cobros; y al no contar con los recursos tecnológicos los realizan de forma manual, lo cual ocasiona tiempos de espera inadecuados e inconformidad para sus socios.

El presente trabajo tiene por objeto ofrecer un prototipo de gestión de cobros a la Junta de Agua Potable de la comunidad de Guartanchun para beneficia a su administración y a los usuarios reduciendo el tiempo de espera, evitando errores humanos en el cálculo de valores a cobrar, incrementado la eficiencia, reduciendo las cargas de trabajo, entre otros. La implementación del sistema podría además ser replicado a otras organizaciones formadas con fines similares para contribuir al desarrollo y crecimiento local.

1.4 Alcance

Con el desarrollo del presente trabajo de investigación, se pretende reducir la carga de trabajo realizado por el personal encargado de controlar y cobrar por los servicios que la junta de agua potable y alcantarillado Monseñor “Leónidas Proaño” de la comunidad de Guartanchún de la parroquia Guapán.

Administrar la organización y cobrar por el uso de los servicios a los usuarios, no es un proceso complejo, pero si tedioso cuando se realiza de forma manual, el cual inicia con la toma de lecturas en cada medidor y finaliza con la elaboración de comprobantes de pago para cada usuario por el consumo y uso de los servicios.

Para cumplir con este trabajo, se desarrollará un aplicativo basado en Cloud Computing de Amazon Web Services para garantizar la disponibilidad, escalabilidad, accesibilidad y flexibilidad a bajo costo en el alojamiento de los datos y la aplicación y a la vez que permitir su despliegue a través de internet.

Por ser un prototipo, es una versión preliminar de la solución lo que implica que se creará con la finalidad de probar y validar la funcionalidad y rendimiento antes de su despliegue final. Por ello esta versión conlleva de ciertas limitaciones como: no contar con un login o inicio de sesión debido a que el prototipo se enfocó solamente en las funciones de facturación, adicionalmente, este prototipo no utiliza facturación electrónica porque el decreto ejecutivo para obligar a que esta fuese ejecutada fue posterior a la definición de los requerimientos iniciales convenidos, por lo que esta funcionalidad está fuera del alcance de este trabajo.

1.5 Conceptos Relacionados

- **TIC.** – las Tecnologías de la Información y Comunicación se refiere a las redes de comunicación e informática que llegan a clasificarse en diferentes tipos de tecnologías, desde dispositivos de radio y televisión hasta computadoras [2].

Con los avances tecnológicos el impacto que la tecnología tiene socialmente ha crecido y se ha vuelto más indispensable tener dispositivos tecnológicos y servicios en línea para adaptarse a nuevos entornos sociales, laborales, educativos, etc.

- **Área Rural.** – a diferencia de un área urbana un área rural se le puede definir como un lugar o ubicación dedicado a tareas agropecuarias como agricultura, ganadería y cría de aves y animales menores. Normalmente son lugares abiertos que se encuentran en zonas periféricas de áreas urbanas cercanas y áreas lejanas a las ciudades [3].

- **Cloud Computing**. – este es un sistema computacional distribuido de forma que están conectados de forma dinámica, creando un solo recurso unificado basado en los servicios que se vayan a proveer al usuario [4].

- **Servicios en la nube**. – la nube ofrece diferentes servicios que a la vez está clasificada en 3 clases que se adaptan a diferentes necesidades de la siguiente manera [5]:

- **Infraestructura como servicio (IaaS)**. Son los servicios más simples que pueden ser ofrecidos como almacenamiento.
- **Plataforma como servicio (PaaS)**. Se puede ofrecer una plataforma de desarrollo como servicio donde se pueden llegar y ejecutar aplicaciones.
- **Software como servicio (SaaS)**. No se requiere montar una infraestructura ya que todo se encontrará en la red para su uso.

- **AWS**. – es conocido como uno de los proveedores de servicios en la nube más conocido, que cuenta con una gran cantidad de productos hechos para adaptarse a las diferentes necesidades que el usuario o la empresa requiere. Con ese modelo llegó a impulsar grandes negocios entre ellos Netflix, a la vez impulsando a organismos gubernamentales [6].

1.6 Trabajos Relacionados

- En el año 2012 se realizó la implementación del sistema contable para la Junta Administradora de Agua Potable Mobiloil & Mosquera, donde se buscaba agilizar los diferentes procesos de contabilidad [7].

- En el año 2013 se busca desarrollar una aplicación para dispositivos móviles, que permite hacer pagos de servicios básicos como electricidad, agua, TV cable, entro otros. De igual forma se

busca agilizar la manufacturación y dar una advertencia de corte de algún servicio, este proyecto da la capacidad a las personas de pagar los servicios básicos en cualquier lugar en el que se encuentre [8].

- En el año 2019 se realiza una aplicación web para agilizar los procesos de pagos y facturación de una empresa de acceso a vehículos como el registro y pago de alcuotas por lo que se buscó agilizar este proceso de mejor forma para dar atención a los usuarios [9].

1.7 Metodología

Identificación de Variables dependientes e Independientes

- Variables Dependientes: Satisfacción, Agilidad en los Procesos
- Variables Independientes: Facilidad de manejo del Programa, Conocimiento acerca del proceso de Facturación.

Para este trabajo se planea utilizar metodología Scrum por la integración de diferentes elementos, dando la posibilidad de realizar revisiones en un determinado tiempo.

A continuación, realizamos una descripción los pasos que se harán durante esta investigación:

1.- **Levantamiento de requerimientos.** Se busca indagar acerca de las funciones que deberá cumplir la aplicación y se establece las herramientas óptimas que se utilizarán para su desarrollo.

2.- **Diseño y desarrollo de la aplicación.** Se comienza con el diseño y desarrollo del backend y el frontend, y se analiza los componentes de los diferentes menús que tendrá el aplicativo y las conexiones con una base de datos.

3.- **Evaluar el aplicativo.** Una vez desarrollado el prototipo se realizan diferentes pruebas al sistema para buscar fallos y realizar arreglos.

4.– **Probar el aplicativo.** Una vez ya desarrollado el aplicativo se procede a realizar pruebas con las personas encargadas del área de facturación para determinar que el software es funcional y cumple con los objetivos establecidos.

CAPITULO 2. Fundamentación Teórica

2.1 Marco Teórico

En el capítulo anterior se describió los diferentes trabajos relacionados, opciones y herramientas para el desarrollo del prototipo. En base a los diferentes elementos que se analizaron en este capítulo se define de mejor manera varios de los elementos como la metodología que se va a utilizar y las herramientas que se utilizarán.

2.2 Metodologías para el desarrollo de Software

En la actualidad existen diversas metodologías para el desarrollo de software que pueden llegar a adaptarse a diversos tipos de proyectos de desarrollo de software, estas metodologías definen el proceso de desarrollo e implementación, pero cada metodología es diferente en sus procesos, al momento de entregar el software de forma definitiva y en el análisis de los requerimientos del software. Por ello, se planteó hacer una comparativa entre dos metodologías para en base a ese análisis elegir la mejor opción a tomar para el desarrollo del sistema (Tabla I).

Tabla I. Comparativa de metodologías de desarrollo de Software

	SCRUM	XP
Utiliza Roles	Si	Si
Etapas	Sprint	4
Concepto	Son un conjunto de varias prácticas de trabajo colaborativas para el desarrollo de software que se realiza de manera cíclica (Sprint).	Son prácticas y reglas para el desarrollo de un sistema de software.
Ventajas	Es una metodología que se adapta a los cambios a medida que se va desarrollando, cumpliendo con los diferentes estándares.	Se tiene una mejor comunicación con el equipo de trabajo evitando tener errores al momento de entregar el software.
Desventajas	El tiempo que se requiere suele ser bastante amplio debido al poco avance que puede haber con el equipo de trabajo	Se utiliza para trabajos muy pequeños y demandan altos costos.
Adaptabilidad	Está adaptada a equipos de varias o pocas personas durante el desarrollo	Se adapta a pequeños grupos y puede ser problemático al momento de hacer en grupos grandes

En base a la comparativa realizada, se estableció que la mejor alternativa para el avance del prototipo sería la metodología Scrum, ya que esta permitirá el adaptarse a los diferentes cambios

que se requieran y es la más recomendable al momento de realizar prototipos, también con la posibilidad de realizar revisiones durante un determinado periodo de tiempo para adaptarse a los cambios que se requieran.

2.3 Metodología SCRUM

La metodología de SCRUM se basa en un conjunto de buenas prácticas para la creación de software de forma colaborativa, se puede definir como una metodología ágil que se enfoca en obtener el mejor producto posible, basado en el apoyo mutuo entre los colaboradores para entregar el software de la mejor manera [9].

2.3.1 Elementos de la Metodología SCRUM

La metodología SCRUM suele estar conformada por 3 elementos principales [10]:

Product Backlog: es una lista o inventario donde se registra todos los requisitos y funciones que tendrá, se implementarán durante el desarrollo y que tendrá el producto final.

Sprint Backlog: Es una lista en la que se especifica que trabajo debe cumplir cada integrante del equipo de desarrollo y el tiempo en el que deberá cumplir con la parte que se le ha asignado.

Incremento: se muestran los objetivos ya completados y en base a ello se analiza si se debiera realizar algún cambio o dar algún añadido.

2.3.2 ¿Por qué implementar SCRUM?

SCRUM ayuda a resolver los problemas involucrando al cliente con el equipo de desarrollo, estos en conjunto se comprometen a realizar avances sobre el desarrollo y el sprint. A diferencia de otras metodologías tradicionales que se realizan en una estructura de desarrollo definida,

SCRUM se basa en hacer entregas de las partes de un producto al cliente, después, este dará su opinión y soporte para entregar el proyecto de la mejor forma posible [10].

2.4 HTML

HTML o (lenguaje de marcado de hipertexto) es un lenguaje para el desarrollo de sitios web, con el cual se puede incluir diversos elementos desde textos, gráficos, sonidos, videos, etc.

Desde los primeros momentos que una persona utiliza internet encontramos páginas con una gran cantidad de contenido visual y auditivo como imágenes, sonidos, efectos, etc. Cada Página web es diferente en contenido, pero comparten un elemento en común el cual es el lenguaje de programación utilizado para su creación, en este caso HTML [11].

2.4.1 Etiquetas HTML

Una parte fundamental dentro de HTML es el uso de las etiquetas ya que estas ayudarán a organizar la estructura de la página, la inserción de elementos audiovisuales y establecer la conexión con una base de datos, en resumen, la esencia de la programación en HTML son las etiquetas [11].

2.4.2 HTML 5

Esta es la última versión de HTML que fue desarrollada con la intención de resolver los problemas que tenía la 4ta versión, esta versión permite tener una mejor interacción con los datos locales y con servidores de manera más fácil y rápida, adicionalmente, HTML 5 ya no depende de APIs y extensiones de terceros que causaban problemas de incompatibilidad, de esa forma, ahora se reducen tiempos de carga y se obtienen una mejor compatibilidad con los servidores y con la interfaz [11]. Además, se agregaron nuevas funciones entre las cuales se destaca:

- Implementación de nuevos atributos
- Capacidad de eliminar elementos repetitivos
- Edición de páginas sin necesidad de estar conectado.
- Almacenamiento en bases de datos como MySQL.

2.5 JAVA

JAVA es creada en 1995 por Sun Microsystems como un lenguaje de programación, el cual, ha evolucionado a lo largo del tiempo desde sus inicios hasta la actualidad siendo un gran apoyo en el entorno digital para el crecimiento de diversos servicios y aplicaciones [12]. Hoy en día JAVA llega a ser parte fundamental de varias aplicaciones inclusive habiendo una gran variedad de páginas web que hacen uso de java para su ejecución.

2.6 Framework

2.6.1 Definición general

“Conjunto de conocimientos y técnicas cuya aplicación permite la utilización racional de los materiales y de los recursos naturales, mediante invenciones, construcciones u otras realizaciones provechosas para el hombre” [13].

En base a esa definición se puede entender a un framework como una estructura de un grupo de conocimientos y técnicas que se tienen al momento de hacer uso de las diferentes herramientas y recursos para obtener un beneficio.

2.6.2 Framework Web

“Un Framework para aplicaciones web se puede considerar como una aplicación genérica incompleta y configurable, con directrices arquitectónicas ofreciendo al desarrollador un conjunto de herramientas para agilizar el proceso de construir una aplicación web concreta, siempre

teniendo en cuenta que es necesario adaptarlo para cada una de las aplicaciones a desarrollarse” [14].

Un Framework web son diferentes librerías las cuales ayudan el desarrollo de aplicativos webs, ahorran los tiempos de desarrollo, reutilizando códigos ya preestablecidos para un determinado propósito. Una de las ventajas del uso de frameworks es que con estos se pueden hacer una separación entre sus componentes y realizar su implementación separada e independientemente.

2.7 Desarrollo de Aplicaciones Web

Durante un inicio, la interacción con la web se basaba en recolectar diferentes archivos como páginas estáticas como documentos, imágenes, entre otros. Todo este proceso funcionaba haciendo uso de peticiones a un servidor, creando el protocolo HTTP, para de esa manera obtener una respuesta de parte de un servidor. Con el tiempo este proceso ha recibido bastantes cambios que el día de hoy nos garantizan seguridad y nos dan mayor velocidad al momento de navegar por la red, los elementos que se requerían para tener una conexión a la red fueron [15]:

Dominio. - es un espacio concreto en la red donde se puede acceder con una dirección o enlace determinado.

Servidor DNS. – este descifra el nombre de una dirección web para convertirla a una dirección IP en donde se encuentra alojada el servidor.

Servidor Web. –se encarga de almacenar la información que reciba de parte de otros ordenadores. Su trabajo es estar operativo el mayor tiempo posible dando una disponibilidad continua.

2.8 Editores de Código

Existe una gran variedad de editores de código en el mercado, desde gratuitos hasta editores de código de pago, de escritorio y en línea. Las cuales tienen el objetivo de facilitar el trabajo de desarrollo y edición de software, estos editores nos ayudan al momento de realizar cambios en el código, buscar errores y entender mejor el lenguaje de programación que estemos utilizando. A continuación, una comparativa de diferentes editores de código:

Tabla II Comparativa de Editores de Código

	VSC (Visual Studio Code)	ATOM	WebStorm
Multiplataforma	Si	Si	Si
Open Source	Si	Si	No
Documentación	Si	Si	Si
Terminal Integrado	Si	No	Si
Lenguajes de extensión	Si	Si	Si
Compatibilidad con otros lenguajes	Si	Si	Si

Fuente: <https://dSPACE-UCACUE-EDU-EC.VPN.UCACUE.EDU.EC/HANDLE/UCACUE/10422>

En base a esta comparativa se ha optado por utilizar Visual Studio Code para el desarrollo del prototipo del aplicativo.

2.9 Visual Studio Code

VSC es conocido como un ligero, pero poderoso editor de código multiplataforma para escritorio, que soporta diversos lenguajes de programación como JavaScript, TypeScript y Node.js. Este también posee un ecosistema adaptable y una gran variedad de extensiones para poder hacer uso de otros lenguajes de programación como C#, Python, entre otros [9].

2.10 Frontend

El Frontend o GUI (Graphical User Interface) o en español Interfaz Gráfica del usuario, se define como la sección de la creación de software que está encargada de definir el diseño y el aspecto de la página con la cual el usuario deberá interactuar y donde se le será presentada la información de una manera fácil de comprender [16].

Durante el desarrollo del Frontend es importante el tener un buen conocimiento sobre los procesos y técnicas que manejan los usuarios a los cuales les será entregado el software, a la vez se deberá diseñar una interfaz que pueda ser cómoda para el usuario y pueda tener acceso de la manera sencilla a las funciones básicas del software, para de esta forma poder ofrecer una mejor experiencia entre el usuario y el programa [16].

Para desarrollar el frontend existen diferentes herramientas de programación que se especializan en este apartado y que utilizan diferentes lenguajes, pero una de los más destacados y más utilizados es HTML, esto junto a la utilización de un Framework ayudan a agilizar los procesos de desarrollo de software utilizando modelos precargados, herramientas y uso de bibliotecas para la mejora del proyecto. A continuación, se realiza una comparativa de diferentes Frameworks:

Tabla III Comparativas de Framework para el desarrollo de la interfaz

	Angular	Vue	React
Lenguaje de Programación	TypeScript	JavaScript	JavaScript
Desarrollador	Google	Comunitario	Facebook
Open Source	Si	Si	Si
Modelo de desarrollo	MVC	DOM	DOM
Curva de Aprendizaje	Empinado	Medio	Moderado
Enlace de datos	Bidireccional	Bidireccional	Unidireccional

Fuente: <http://sedici.unlp.edu.ar/handle/10915/120476>

Basado en esa comparativa contemplamos las diferentes características que ofrece cada framework para el desarrollo del frontend, Angular a diferencia de Vue y React utiliza el lenguaje de programación TypeScript además de utilizar un modelo de desarrollo de tipo MVC (Modelo Vista Controlador) el cual es el más adecuado para el avance de la propuesta.

Basado en ello se ha optado por utilizar Angular Framework ya que este ofrece las herramientas necesarias, además, brinda un entorno al cual uno ya está familiarizado.

2.11 Angular

Angular es un framework OpenSource desarrollado por Google, el cual es planeado como un rediseño de su predecesor, AngularJS. Esta fue la primera versión de Angular que fue lanzada en septiembre de 2016 con el nombre de Angular 2. Tiempo después, su nombre se cambiaría a solo llamarse Angular. En un mismo año el equipo de desarrollo de angular realizaría varios cambios significativos para este framework. Futuros lanzamientos llegarían con nuevas características, mejor rendimiento y pequeños paquetes de códigos ya generados, pero a pesar de todo está seguía manteniendo su arquitectura [17].

Angular Framework es una plataforma que incluye todo lo que se necesita para el desarrollo de aplicativos webs. Es un framework que cumple con varias funciones y ofrece diversas características:

- Generar un SPA (Aplicación de una sola Página) en un par de segundos utilizando Angular CLI.
- Insertar y remplazar servicios de una manera sencilla.
- Establecer una clara separación entre la interfaz y la parte lógica.
- Crear Interfaces más modernas haciendo uso de las diferentes librerías de Angular.
- Se puede hacer uso de diferentes lenguajes de programación como HTML, CSS y JavaScript.

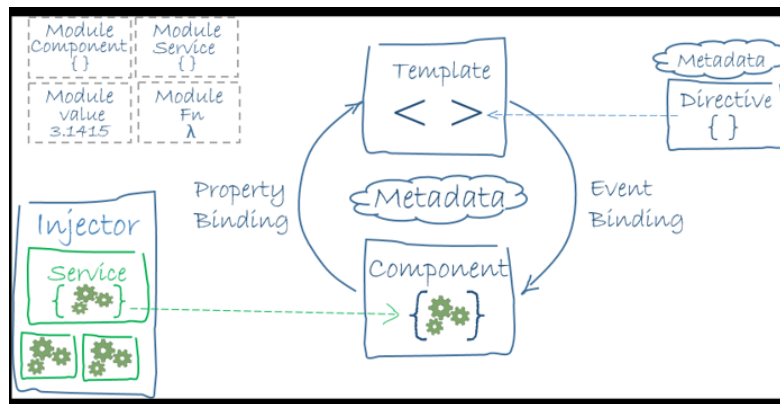


Fig. 1 Arquitectura Angular

Fuente: <https://angular.io/generated/images/guide/architecture/overview2.png>

2.11.1 Arquitectura de Angular

La arquitectura de Angular se basa en el uso de plantillas y componentes [9]:

Módulos: en el entorno de Angular son denominadas NgModules los cuales complementan módulos de JavaScript, pueden utilizarse los diferentes recursos que nos ofrecen para implementarlos dentro de nuestro código.

Componentes: los componentes son bloques de códigos ya construidos que formarán parte del aplicativo. Este está dividido en la parte visual y la parte lógica asociándose con una plantilla HTML que será la vista del sistema.

Servicios: angular ofrece código reutilizable con el objetivo de dar más funcionalidad a sus componentes.

Inyección de dependencias: Mantiene los componentes de las clases de tal forma que pueda ejecutar sus funciones.

2.12 Backend

El Backend es la parte en donde se centra en el manejo de datos y la parte lógica del funcionamiento interno del software como: entablar una relación entre la base de datos y el aplicativo de software, seguridad del aplicativo y uso de recursos [16].

En el Backend se busca desarrollar y entablar una relación con el servidor, ya sea que el aplicativo sea en la web, de escritorio o móvil. Por ello es necesario tener conocimiento sobre diferentes lenguajes de programación como C#, PHP, JAVA, entre otros. Además de poder contar con las herramientas adecuadas para esta parte del proyecto, a la vez se requiere el poder entender las interacciones con diferentes bases de datos como pueden ser: MySQL, PostgreSQL, MongoDB, entre otras [18].

Tabla IV Comparativas de Framework para el desarrollo del Backend

	Spring Boot	Node.js
Desarrollador	SpringSource	Node.js Developers
Open Source	Si	Si
Soporta Bases de Datos SQL	MySql, Postgres, Oracle, DB2, MariaDB.	MongDB, Cassandra, CouchDB, Redis.
Seguridad	Alta	Media
Rest API	Si	Si

Fuente:<https://www.proquest.com/docview/2195127847/fulltextPDF/73706B8F87734DBCPQ/1?accountid=61870>

Se ha optado por utilizar Spring Boot ya que esta ofrece las herramientas necesarias para el desarrollo del prototipo de una manera más sencilla.

2.13 Spring Boot

Spring Boot es una herramienta que funciona para la creación de aplicativos web y microservicios de forma más eficiente, utilizando Spring Framework se puede configurar las diferentes variables al momento de crear un nuevo proyecto. El objetivo principal de Spring Boot era el de resolver los diferentes problemas que tenía Java EE, pero las versiones iniciales eran muy

incompletas y el desarrollo del aplicativo no era nada sencillo por ello el objetivo de los desarrolladores era el de crear un producto accesible para todo el mundo que pudiese facilitar el desarrollo de software [19].

2.14 Spring Framework

Spring Framework es una plataforma desarrollada por SpringSource que tiene el objetivo de proveer librerías y componentes para la creación de aplicaciones en JAVA. Este se ocupa de manejar el apartado de infraestructura para tener un mejor enfoque en el desarrollo del aplicativo [17].

Spring Framework nos ofrece una gran diversidad de características para la creación del prototipo como:

- Creación de aplicaciones independientes
- Configuración automática de bibliotecas nativas y de terceros
- Hace fácil el uso e implementación de dependencias
- Se pueden realizar testeos de una aplicación de forma sencilla
- Tiene librerías que facilitan y evitan las líneas de condigo

2.14.1 Módulos de Spring Framework

Spring Framework ofrece una gran librería de diferentes tipos de módulos y herramientas para determinadas funciones como integración de información, integración web, pruebas o testeo, entre otras funciones que están clasificadas en diferentes categorías:

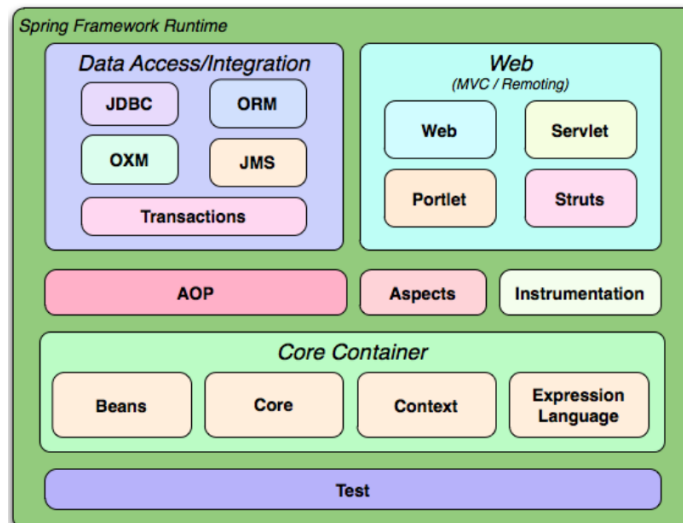


Fig. 2 Módulos de Spring Framework

2.14.2 ¿Por qué utilizar Spring Framework?

Porque este nos permite desarrollar un aplicativo de manera más rápida y sencilla y tener una aplicación completamente funcional, nos ofrece ayuda para trabajar con diferentes bases de datos de tipo SQL como Postgres o MySQL, nos ofrece un excelente nivel de seguridad y nos ofrece una gran cantidad de librerías para diferentes apartados.

2.15 IntelliJ Idea

IntelliJ Idea es un editor de código basado en Java desarrollado para la creación de software con lenguajes de programación de tipo JVM. Desarrollado por la empresa JetBrains, IntelliJ Idea se destaca por su compatibilidad al momento de utilizar JavaScript junto con varios marcos de trabajo como Spring Boot, Jakarta, Helidion, entre otros.

Para el desarrollo de este prototipo se utilizará IntelliJ Idea para el desarrollo del Backend del aplicativo, haciendo uso de las librerías de Spring Boot y Lombok para facilitar la creación de los diferentes componentes del sistema [20].

2.16 Servicios REST

Un servicio web de tipo API REST, es utilizado para exponer o receptor información mediante tramas que comúnmente son Json o texto plano; utilizando los métodos propios considerados en la tabla 2.5 se puede realizar un proceso de CRUD que consiste en (Create, Read, Update, Delete, este tipo de servicios se utilizan en los diferentes procesos y operaciones de mantenimiento a la información almacenada en la base de datos. Estos servicios se pueden realizar en entidades o colecciones.

Las colecciones son conjuntos o listas de entidades, haciendo uso de CRUD sobre estas listas el mapeo se haría de la siguiente manera:

Tabla V Uso de Métodos CRUD en Colecciones

Método HTTP	Ejecución
Get	Lee todas las entidades dentro de la colección
Put	Actualización una colección existente
Delete	Elimina todas colección con sus entidades
Post	Crear una nueva entidad

Las Entidades se definen como instancias en concreto, estos se ordenan dentro de una colección.

De la misma manera se hace uso de CRUD:

Tabla VI Uso de Métodos CRUD Entidades

Método HTTP	Ejecución
Get	Lee los datos de una entidad
Put	Actualiza una entidad existente
Delete	Borra toda una entidad
Post	Añade información a una entidad ya existente

2.17 Lombok

Lombok es una librería de Java que facilita la creación de código evitando repeticiones de código reemplazando estas repeticiones con una sola anotación al principio del código. Al momento de hacer la integración de las herramientas de Lombok al proyecto se realizan inyecciones de código [21].

2.17.1 Instalación

Para utilizar los recursos de lombok al momento de crear el proyecto y hacer uso de Maven en el archivo pom.xml agregamos la dependencia de lombok y actualizamos el pom.xml con Maven para hacer uso de las anotaciones de lombok [21].

```
1. | <dependencies>
2. |   <dependency>
3. |     <groupId>org.projectlombok</groupId>
4. |     <artifactId>lombok</artifactId>
5. |     <version>0.9.2</version>
6. |   </dependency>
7. | </dependencies>
8. | <repositories>
9. |   <repository>
10. |     <id>projectlombok.org</id>
11. |     <url>http://projectlombok.org/mavenrepo</url>
12. |   </repository>
13. | </repositories>
```

Fig. 3 Dependencia Lombok

2.17.2 Anotaciones Lombok

Lombok ofrece un gran paquete de anotaciones para facilitar la creación de clases, entre algunas anotaciones utilizadas tenemos [21]:

@AllArgsConstructor: funciona para la generación de constructores.

@Getter: genera automáticamente los métodos getter de cada campo.

@Setter: genera automáticamente los métodos setter de cada campo.

@ToString: genera la implementación de un método toString a cada campo.

2.18 Maven

Maven es una herramienta de construcción para proyectos de JAVA, diseñado para realizar los procesos de construcción a través del uso de declaratorias en donde se describe la estructura del proyecto y sus componentes, a diferencia de hacer uso de complementos de la manera tradicional, Maven aprovecha los estándares y reduce el tiempo que se requiere para la construcción de los Scripts [22].

2.19 JAVA EE

JAVA EE (Enterprise Edition), es una plataforma que fue creada para facilitar la creación de aplicaciones con JAVA de forma más fácil y rápida. El principal propósito de la plataforma de JAVA EE es el de dar a los desarrolladores mejores y más poderosas herramientas o APIs, de esa manera para poder reducir el tiempo de desarrollo del prototipo, reducción de la complejidad de la aplicación e improvisar el rendimiento del aplicativo [23].

JAVA EE es una plataforma creada principalmente por Java Community Process (JCP), los cuales son los responsables de varias de las tecnologías de parte de JAVA. Está conformada por grupos de expertos compuestos por grupos de interés que son los creadores de JSRs (JAVA Specification Requests) para definir las tecnologías de JAVA EE. La comunidad JAVA que se encuentra a cargo de JCP ayuda a mantener los estándares de estabilidad y compatibilidad multiplataforma de las tecnologías JAVA [23].

2.20 Cloud Computing

Existe un gran número de definiciones, pero a esta se le podría definir como un modelo que da acceso a una gran variedad de opciones dependiendo de nuestras necesidades y bajo una cierta demanda, este es realizado a través de la red ofreciendo diferentes opciones de recursos que pueden

ser configurables dependiendo del proveedor, algunos de los servicios pueden ofrecer configuración de las servidores, redes, almacenamiento, servicios y aplicaciones. Funciones que pueden ajustarse de manera sencilla de parte del proveedor. Las características que estos servicios suelen ofrecer son:

- Utiliza un modelo de pago, dependiendo de los recursos que un requiera estos pueden contar con varios recursos dependiendo de la demanda.
- Utilizan una infraestructura que le da la facilidad al usuario de manejar los diferentes procesos dinámicos y remotos a través de la red.
- Varios usuarios pueden hacer uso de los mismos recursos de manera simultánea desde diferentes lugares, esto sin provocar ningún daño.
- El tener servicios que serán controlados, monitorizados y cuantificados para de esa manera cumplir con las políticas de información y privacidad.

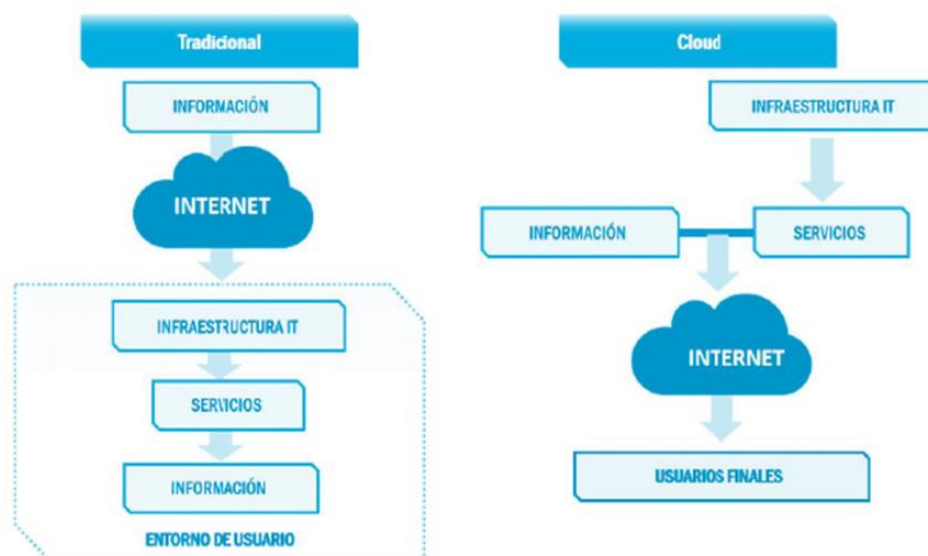


Fig. 4 Cloud Computing

Fuente: https://www.incibe.es/sites/default/files/contenidos/guias/doc/guia-cloud-computing_0.pdf

2.20.1 Modelos de Cloud Computing

Dentro del Cloud Computing existen 3 tipos de modelos los cuales son definidos a continuación:

Infraestructura como un Servicio (IaaS): este le ofrece al cliente un ambiente o una infraestructura donde él usuario pueda trabajar, de esta manera entregándole el software y el hardware sobre el que cual se montara el servicio y se maneja los diversos recursos y el almacenamiento. El acceso a estos recursos suele hacerse bajo demanda y puede ser utilizado desde cualquier lugar.

Plataforma como Servicio (PaaS): esta da al consumidor un sistema o plataforma en donde el usuario podrá instalar diferentes aplicaciones, a diferencia de la informática tradicional que requería de diferentes partes para el desarrollo e implementación, pero con este servicio todo ya viene listo y preparado para que pueda utilizar internet, ofreciendo un entorno completo para el desarrollo de aplicaciones para uso administrativo. Este también se utiliza bajo demanda.

Software como servicio (SaaS): se basa en entregar un aplicativo como servicio y que solo necesita tener una conexión a internet para hacer uso del servicio y sus recursos, con la informática tradicional el usuario debía pagar para poder obtener una licencia que le permitía configurar la infraestructura y la plataforma para la ejecución de la aplicación, con este servicio todo ese proceso ya no es necesario y se logra facilitar el uso del aplicativo y hacer más económico su uso. Los clientes ya no necesitan pagar por una licencia para su uso y este se pagará según el uso que este tenga.

2.20.2 Tipos de Nube

Existen diferentes maneras de implementar servicios de la nube, pero su implementación también se determina en base a cuáles sean las necesidades de los usuarios que utilicen ese servicio. Por ello se llegaron a establecer 4 tipos de nube los cuales se definen a continuación:

Nube Pública: Esta también es conocida como nube externa, es abierta a consumidores de cualquier tipo, es decir, que cualquiera puede hacer uso de ella con tal de pagar una suscripción. Ya sea que el consumidor sea una sola persona, empresa u organización. La ubicación física de los servidores esta fuera de las instalaciones y se puede acceder a ellos de forma remota. Los usuarios que hagan uso de este tipo de nube no tendrán muchos problemas en el ámbito administrativo e infraestructural. Algunos de los servicios que existen son: AWS (Amazon Web Service), Google Cloud, Microsoft Azure, entre otros.

Nube Privada: También conocida como nube interna, esta no proporciona su uso a todo público, está enfocado principalmente en el uso organizacional y de manera restringida al público en general. Estas son utilizadas con propósitos internos dentro de las organizaciones. Tiene un mayor enfoque en cuestiones analíticas y de seguridad de la información. Normalmente los servidores de este tipo de nube se suelen encontrar dentro de las propias organizaciones, de forma que son manejadas por la propia organización o sus consumidores. Algunos de los ejemplos mencionados anteriormente ofrecen también servicios privados de la nube.

Nube Comunitaria: Este servicio puede ser utilizado por organizaciones o consumidores los cuales pertenecen a una determinada comunidad para cumplir con un objetivo específico o un propósito en común de esa comunidad. Su implementación puede ser de manera local o externa, en resumen, el objetivo de este tipo de nube está en ofrecer a una comunidad los beneficios de una

nube pública y a la vez un gran nivel de seguridad y privacidad de una nube privada para ayudar a cumplir una meta en común.

Nube Híbrida: Esa es una combinación que suele darse de mayor forma entre un servicio de una nube privada con una nube comunitaria, este modelo ayuda a las empresas a poseer los beneficios de una nube privada y comunitaria para el almacenamiento y seguridad de datos. De la misma manera se obtiene una gran rentabilidad de mantener los datos y aplicaciones compartidas en una nube pública.

Tabla VII Comparativa de Servicios de la Nube

	AWS	Google Cloud	Microsoft Azure
Lenguajes	JavaScript (NodeJS), Python, binary	JavaScript (NodeJS)	C#, Node.js, Python, PHP, batch.
Sistema Operativo	Linux x64	Linux x64	Windows 32-bit
Licencia	Closed Source	Closed Source	Open Source
Máximo de funciones en un proyecto	Ilimitado	Ilimitado	20
Recursos Reservados	Memoria reservada por función	Ninguno	Memoria Reservada por servicio
Costes	\$ 40 un servidor pequeño y \$ 135 un servidor mediano	\$35 un servidor pequeño y \$ 143 un servidor mediano	\$ 45 un servidor pequeño y \$ 180 un servidor mediano

En esta comparación se examinan las diferentes características empezando por el lenguaje de programación que soportan; AWS soporta lenguajes básicos esenciales como JAVA y Python, seguido de Google Cloud el cual solo soporta JAVA y finalmente Microsoft Azure también da soporte a lenguajes esenciales como C#, Node.js y Python. En el apartado del sistema operativo AWS y Google Cloud utilizan Linux y Azure usa Microsoft. Con las licencias AWS y Google Cloud llegan a ser cerrados en sus licencias a diferencia de Azure que ofrece una licencia open source. Al hablar sobre disponibilidad de recursos Google Cloud ofrece una mayor flexibilidad y disponibilidad. Y finalmente al estimar los costes Google y Amazon llegan a ser las mejores opciones siendo las más económicas para ofrecer servicios pequeños y medianos.

Para el desarrollo de esta propuesta se ha optado por utilizar AWS como base de datos en la nube ya que sus funciones son las más adecuadas para almacenar la información de los usuarios, además de ello de contar con un periodo de pruebas para analizar el entorno del servidor.

2.21 Amazon Web Services (AWS)

AWS fue creado en 2006 con el objetivo de ofrecer servicios en la nube. Brindando a las empresas una infraestructura y que estas no tengan que preocuparse por la adquisición de servidores físicos, de esa manera este servicio busca adaptarse a las necesidades específicas de los usuarios o empresa [5].

Por ello, este servicio ofrece una gran cantidad de productos y herramientas de diferentes categorías:

Explorar las categorías de productos principales



Fig. 5 Productos de Amazon Web Service
Fuente: <https://aws.amazon.com/es/products>

Computación: es un servicio de cloud computing llamado Amazon Elastic Compute Cloud (EC2), el cual nos ofrece la capacidad de crear entornos virtuales en la nube.

Almacenamiento: esta sección ofrece un servicio de almacenamiento en la nube, este servicio se llama Amazon Simple Storage Service (S3)

Base de datos: ofrece diferentes tipos de bases de datos dependiendo de las diferentes necesidades de sus usuarios.

Redes y entrega de contenido: también llamado Amazon VPC (Virtual Private Cloud) da la capacidad a los usuarios de crear redes virtuales privadas a las cuales solo pueden acceder los miembros de una empresa o un determinado círculo, permitiendo el control y gestión de la información.

Análisis: ofrece herramientas de BI (Business Intelligence) las cuales nos proveen la posibilidad de obtener conclusiones sobre los datos obtenidos, a la vez permitiendo gestionar el almacenamiento de la información de una manera más fluida.

Machine Learning: Ofrece servicios utilizando Inteligencia artificial para su uso en diferentes entornos y ocuparse de diferentes trabajos comunes.

Seguridad, identidad y cumplimiento: da la capacidad de otorgar permisos de acceso a los usuarios.

2.22 PostgreSQL

Es una base de datos Open Source, que utiliza y extiende el lenguaje SQL combinándolo con diferentes funciones para el resguardo y escalabilidad de fuentes de datos más grandes y complicadas. PostgreSQL tiene una buena reputación debido a su arquitectura, rentabilidad, integridad de la información, su gran set de herramientas y dedicación con la comunidad Open Source a cargo del software, mejorando cada vez el software y dando soluciones innovadoras. [24]

PostgreSQL vienen con varias características para ayudar a los desarrolladores en la construcción y desarrollo de aplicaciones, administradores para mantener la integridad de la información y construir un ambiente a prueba de fallas, nos ayudara a manejar los datos sin importar lo pequeños o grandes que sean y finalmente los extensible que es con diferentes tipos de información, construcción de funciones y el utilizar diferentes lenguajes de programación.

2.23 StarUML

Este es un software multiplataforma, el cual fue creado como una herramienta para el desarrollo de diferentes tipos de esquemas y diagramas.

En este caso se hará uso de esta herramienta para el diseño de la base de datos del sistema con sus diferentes entidades, conexiones, llaves primarias y foráneas. Las cuales determinan el flujo de la información [25].

CAPITULO 3. Propuesta

Durante este capítulo se describirá el proceso de creación del prototipo de una aplicativo web para la facturación de una junta administradora de agua, y con ello agilizar los procesos de registro y análisis de los datos para mejorar la experiencia de los encargados del área de facturación y sus usuarios, en este caso los encargados del área de facturación y registro de cuotas, también se describirá todos los procesos previos al desarrollo y en base a la información obtenida de los encargados determinar los diferentes requerimientos que tendrá el software, cabe aclarar que se utilizará metodología SCRUM para el avance del proyecto, con ello, mantener el compromiso y la productividad con los involucrados.

3.1 Análisis de requerimientos

Se estableció una reunión junto con el presidente de la junta de agua y los encargados del área de facturación con el objetivo de analizar y determinar las funcionalidades del software, una vez recolectada la información se procede a enlistar las que serán las funcionalidades del software. (Ver Anexo A.)

- Almacenamiento de la información de los socios dentro de una base de datos.
- Capacidad de realizar reportes de forma diaria, mensual y anual para el análisis de datos.
- Capacidad de impresión de facturas.
- Almacenamiento de los datos de los medidores de los socios.
- Almacenamiento de las lecturas de los medidores de los socios.
- Almacenamiento de la información sobre el consumo de cada socio.
- Registro y guardado del RUC de las facturas.

3.2 Implementación de la Metodología Scrum

Se ha optado por hacer uso de la metodología SCRUM para el desarrollo del prototipo que consiste en determinar las diferentes etapas del desarrollo de software en los denominados Sprints, se empieza especificando el Product Backlog como historias de usuarios en base a los requerimientos obtenidos los cuales deberán contener:

- Código o Id de la Historia
- Nombre de la Historia
- Estimación de la Historia
- Importancia de la Historia
- Comentario sobre la Historia

En la tabla VIII mostrada a continuación se detallan los requerimientos del proyecto:

Tabla VIII Lista del Product Backlog del modelo inicial

Id	Nombre de Historia	Estimación	Importancia	Comentario
1	Administrar la información de los socios	4	5	La información de los socios podrá ser guardada y podrá ser editada cuando se requiera.
2	Mantenimiento de la información de los medidores y alcantarillados	2	5	Guardar la información de los medidores y alcantarillados, a quien pertenece y a donde pertenece.
3	Mantenimiento de la información de los consumos	2	5	Guardar las lecturas que se registran en cada medidor en un cierto periodo de tiempo.
4	Mantenimiento de los valores pendientes de cada socio	3	6	Registrar los diferentes valores como consumo, multas, alcantarillado. Además de que todo esto pueda ser visualizado.
5	Generación de Reportes	4	4	Generar reportes mensuales
6	Generación de facturas	4	10	Generar factura con los pendientes que tenga un socio
7	Almacenamiento	3	10	Guardar y asegurar información

3.3 Definición de los Sprints

Una vez que se encuentran definidos los Sprints previstos para el desarrollo del proyecto, se definen los Backlog que conforman cada una de las historias de usuario, donde agrupados entre si determinan el tiempo que se empleara para cumplir con cada sprint.

En cada grupo en base a su estimación e importancia se determina el número de horas en base al esfuerzo y el tiempo que requieren para determinar la duración de cada sprint. Estos se definen según:

- La Importancia de las Historias se determinan en rangos de 1 y 10.
- La Estimación de las Historias se determinan rangos de 1 y 4 Semanas (40 horas c/u)

En la tabla IX que se observa a continuación, se muestran las historias de usuario del primer Sprint.

Tabla IX Historias de Usuario 1 y 2

Id	Nombre de Historia	Estimación	Importancia	Comentario
1	Administración de la información de los socios	4	5	La información de los socios podrá ser guardada y ser editada cuando se requiera.
2	Mantenimiento de la información de los medidores y alcantarillados	2	5	Guardar la información de los medidores y alcantarillados, a quien pertenece y a donde pertenece.

En la tabla X que se observa a continuación, se muestran las historias de usuario del Segundo Sprint.

Tabla X Historias de Usuario 3 y 4

Id	Nombre de Historia	Estimación	Importancia	Comentario
3	Mantenimiento de la información de los consumos	2	5	Guardar las lecturas que se registran en cada medidor en un cierto periodo de tiempo.
4	Mantenimiento de los valores pendientes de cada socio	3	6	Registrar los diferentes valores como consumo, multas, alcantarillado, además de que todo esto pueda ser visualizado.

En la tabla XI que se observa a continuación, se muestran las historias de usuario del Tercer Sprint.

Tabla XI Historias de Usuario 5 y 6

Id	Nombre de Historia	Estimación	Importancia	Comentario
5	Generación de Reportes	4	4	Generar reportes mensuales
6	Generación de facturas	4	10	Generar factura con los pendientes que tenga un socio

En la tabla XII que se observa a continuación, se muestran las historias de usuario del Cuarto Sprint.

Tabla XII Historias de Usuario 7

Id	Nombre de Historia	Estimación	Importancia	Comentario
-----------	---------------------------	-------------------	--------------------	-------------------

7	Almacenamiento	3	10	Guardar la información de forma segura en una base de datos
---	----------------	---	----	---

Una vez ya agrupados los diferentes Sprints se procede a estimar el tiempo de realización de cada Sprint, esto basado en su estimación e importancia según lo requiera el proyecto.

3.4 Planificación de los Sprints

Los Sprints a más de abarcar las historias de usuario, cumplen con un rol de gran importancia donde su principal objetivo es el de controlar y realizar revisiones permanentes en cada una de las etapas que componen el ciclo de desarrollo del producto de software, donde al cumplir a cabalidad lo programado como resultado se obtiene una retroalimentación que nos permita realizar mejoras para las siguientes entregas del software.

El progreso que se realiza a lo largo del desarrollo de software se debe registrar en el TaskBoard, donde se enlistan todas las tareas a realizar y se marcará en un recuadro las actividades que se han realizado y las que se están por ser desarrolladas y las tareas finalizadas, con el TaskBoard también se debe de mostrar un Burndown de las actividades que tiene que mostrar la velocidad del desarrollo del proyecto.

A continuación, se muestran los Sprints definidos y una vez que un sprint sea finalizado se adjuntaran los resultados obtenidos de cada Sprint.

Tabla XIII Sprint 1

Sprint 1	
Autor: Kevin Guaman	
Fecha de Inicio	01/Septiembre/2022
Fecha de Fin	30/Septiembre/2022
Duración	80 horas
Tareas a desarrollar	Administración de la información de los socios Mantenimiento de la información de los medidores y alcantarillados

Tabla XIV Sprint 2

Sprint 2	
Autor: Kevin Guaman	
Fecha de Inicio	01/Octubre/2022
Fecha de Fin	31/Octubre/2022

Duración	80 horas
Tareas a desarrollar	Mantenimiento de la información de los consumos Mantenimiento de los valores pendientes de cada socio

Tabla XV Sprint 3

Sprint 3	
Autor: Kevin Guaman	
Fecha de Inicio	01/Noviembre/2022
Fecha de Fin	31/Diciembre/2022
Duración	80 horas
Tareas a desarrollar	Generación de Reportes Generación de facturas

Tabla XVI Sprint 4

Sprint 4	
Autor: Kevin Guaman	
Fecha de Inicio	01/Enero/2023
Fecha de Fin	31/Enero/2023
Duración	80 horas
Tareas a desarrollar	Almacenamiento

3.5 Sprint 1

3.5.1 Eventos del Sprint 1: Con los eventos de este Sprint se consiguió lograr los objetivos planificados puesto que se consideran a diario los avances del modelo Entidad-Relación que nos guiara para la creación del prototipo a lo largo del proyecto y se consideran los datos que se tienen para la generación del frontend y el backend. El progreso de trabajo diario se valida en el Burndown Chart donde se valora el progreso y finalización de las tareas diarias que al ser cumplidas cumplen al 100% con la historia de usuario.

3.5.2 Roles del Sprint 1: En este sprint intervienen el Product Owner y un desarrollador con el cual se establecieron los requerimientos para poder determinar las tareas a cumplir en este sprint.

3.5.3 Artefactos del Sprint 1: Una vez se cumpla lo establecido en el primer sprint obtendremos el modelo de entidad relación con el cual se definirá la estructura del sistema y de la base de datos junto con la creación de tablas base como son de los socios, medidores y alcantarillado.

3.5.4 Sprint Backlog 1

En la tabla XVII se listan las diferentes actividades a realizar junto con su fecha establecida de duración y el tiempo establecido para concluir las actividades relacionadas con las historias de usuario 1 y 2.

Tabla XVII Planificación del Sprint 1

Sprint 1	
Autor: Kevin Guaman	
Fecha de Inicio	01/Septiembre/2022
Fecha de Fin	30/Septiembre/2022
Duración	80 horas
Tareas a desarrollar	Administración de la información de los socios Mantenimiento de la información de los medidores

En la tabla XVIII se listan las tareas que se deben desarrollar para dar cumplimiento a las historias de usuario número 1 y 2, para lo cual se dispone de un lapso de tiempo destinado para el sprint 1 donde se llevará un minucioso registro de la fecha y el estado de avance de cada una de las tareas.

Tabla XVIII Lista de tareas del Sprint 1

Sprint 1			Fecha Inicio	Duración	Días	01-15 Sep	16 - 30 Sep
Delegado/Autor: Kevin Guaman			01-09-2022	4 semanas(80hrs)			
H. Usu	Tarea	Estado	H	40	40		
1	Diseños del esquema de la base de datos	Completo		X			
1	Creación de la clase Socio	Completo		X			
1	Creación de la clase Comunidad	Completo		X			
1	Diseño de las capas de persistencia (JPA)	Completo		X			
1	Creación de las metodologías CRUD	Completo		X			
1	Creación de los DTO y los controladores	Completo		X			
1	Creación de la interfaz gráfica de la clase Socios y Comunidad	Completo		X			
2	Creación de la clase Alcantarillado	Completo				X	
2	Creación de la clase Medidores	Completo				X	
2	Diseño de las capas de persistencia (JPA)	Completo				X	
2	Creación de las metodologías CRUD	Completo				X	
2	Creación de los DTO y los controladores	Completo				X	
2	Creación de la interfaz gráfica de la clase Alcantarillado y Comunidad	Completo				X	

Tabla XIX Lista de Tareas completadas del Sprint 1

Nombre del Proyecto: JuntaMLP Numero de tareas: 13 Sprint: 1			
Encargado de tareas: Kevin Guaman			
Id	Tarea	Historia de Usuario	Fecha
1	Diseños del esquema de la base de datos	1	10-septiembre-2022
2	Creación de la clase Socio	1	10-septiembre-2022
3	Creación de la clase Comunidad	1	12-septiembre-2022
4	Diseño de las capas de persistencia (JPA)	1	13-septiembre-2022
5	Creación de las metodologías CRUD	1	14-septiembre-2022
6	Creación de los DTO y los controladores	1	15-septiembre-2022
7	Creación de la interfaz gráfica de la clase Socios y Comunidad	1	15-septiembre-2022
8	Creación de la clase Alcantarillado	2	17-septiembre-2022
9	Creación de la clase Medidores	2	17-septiembre-2022
10	Diseño de las capas de persistencia (JPA)	2	18-septiembre-2022
11	Creación de las metodologías CRUD	2	18-septiembre-2022
12	Creación de los DTO y los controladores	2	25-septiembre-2022
13	Creación de la interfaz gráfica de la clase Alcantarillado y Comunidad	2	29-septiembre-2022

3.5.5 BurnDown Chart Sprint 1

En la figura 6 es donde se muestra el Burdow Chart de las actividades que indicará el progreso que se haga en cada sprint.

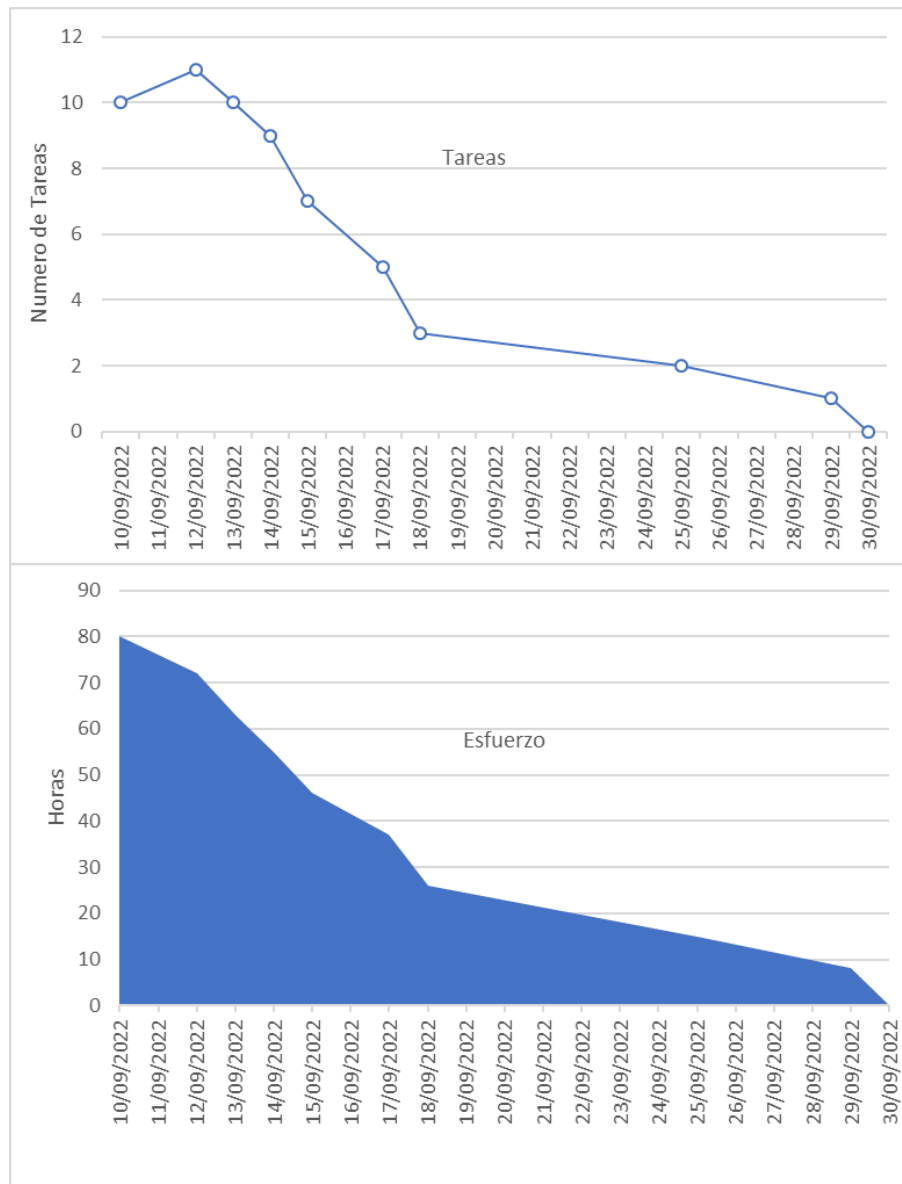


Fig. 6 Bum down chart Sprint 1

3.5.6 Justificación del sprint 1

A continuación, se pueden observar los resultados obtenidos como resultado de la ejecución del primer sprint, para una mejor comprensión se puede revisar la figura número 6.

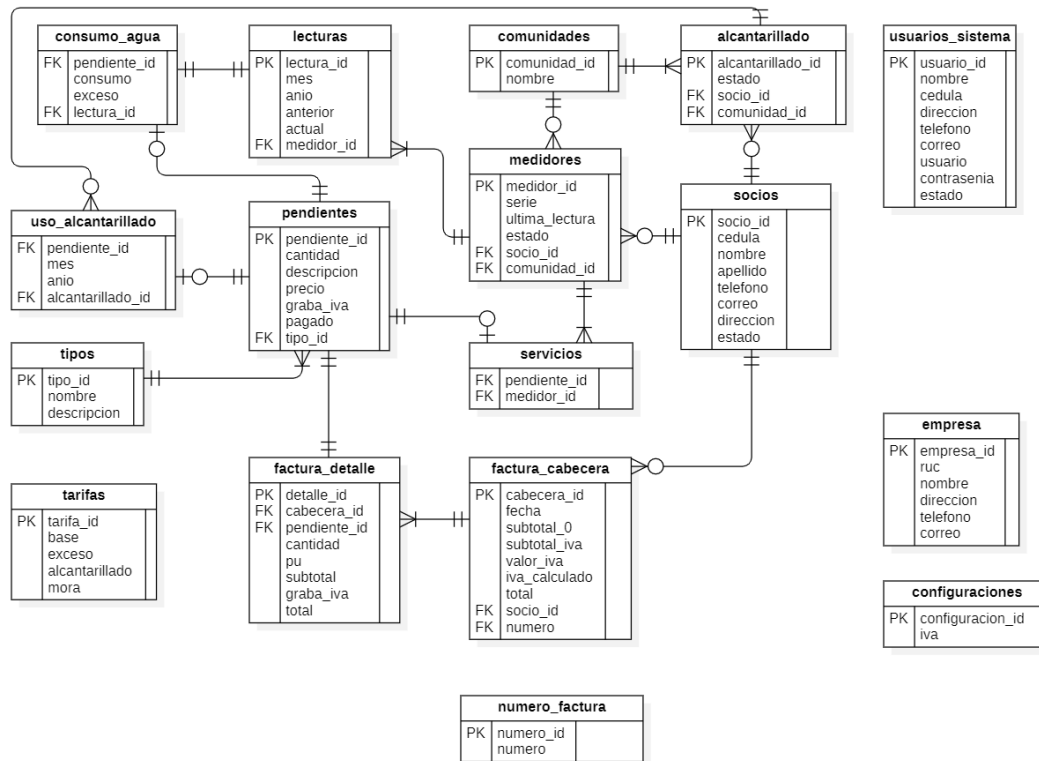


Fig. 7 Diseños de la base de datos

```

VCS Window Help  juntamp - Socio.java

Socio.java x
1  package com.juntamp.model;
2
3  import ...
9
18 usages
10 @NoArgsConstructor
11 @AllArgsConstructor
12 @Getter
13 @Setter
14 @ToString
15 @Entity
16 public class Socio {
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private int socioId;
20
21     private String cedula;
22
23     private String nombre;
24
25     private String apellido;
26
27     private String telefono;
28
29     private String correo;
30
31     private String direccion;
32
33     private String estado;

```

Fig. 8 Generación de la clase Socio

```

VCS Window Help juntamp - Comunidad.java
ad
FacturaCabeceraController.java x Comunidad.java x LecturaController.java x FacturaPDF.java x
1 package com.juntamp.model;
2
3 import ...
4
5
6
7
8
9
10 @NoArgsConstructor
11 @AllArgsConstructor
12 @Getter
13 @Setter
14 @ToString
15 @Entity
16 public class Comunidad {
17     @Id
18     @GeneratedValue(strategy = GenerationType.IDENTITY)
19     private int comunidadId;
20
21     private String nombre;
22 }
23

```

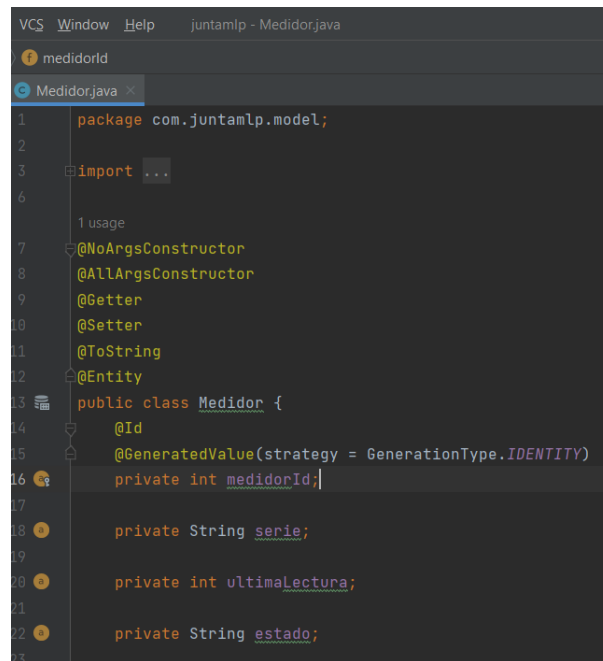
Fig. 9 Generación de la clase Comunidad

```

s VCS Window Help juntamp - Alcantarillado.java
rillado
FacturaCabeceraController.java x Comunidad.java x Alcantarillado.java x
3 import lombok.*;
4
5 import javax.persistence.*;
6
7
8
9
10
11
12
13 public class Alcantarillado {
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private int alcantarilladoId;
17
18     private String estado;
19
20
21     @ManyToOne
22     @JoinColumn(name = "socio_id", nullable = false)
23     private Socio socio;
24
25     @ManyToOne
26     @JoinColumn(name = "comunidad_id", nullable = false)
27     private Comunidad comunidad;
28 }

```

Fig. 10 Generación de la clase Alcantarillado



```
VCS Window Help juntamp - Medidor.java
medidorId
Medidor.java x
1 package com.juntamp.model;
2
3 import ...
4
5 1 usage
6
7 @NoArgsConstructor
8 @AllArgsConstructor
9 @Getter
10 @Setter
11 @ToString
12 @Entity
13 public class Medidor {
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private int medidorId;
17
18     private String serie;
19
20     private int ultimaLectura;
21
22     private String estado;
23 }
```

Fig. 11 Generación de la clase medidores

3.5.7 Entregable Sprint 1

En la siguiente sección se adjuntan las figuras correspondientes a los resultados del primer sprint donde se muestran las secciones de los socios y medidores.

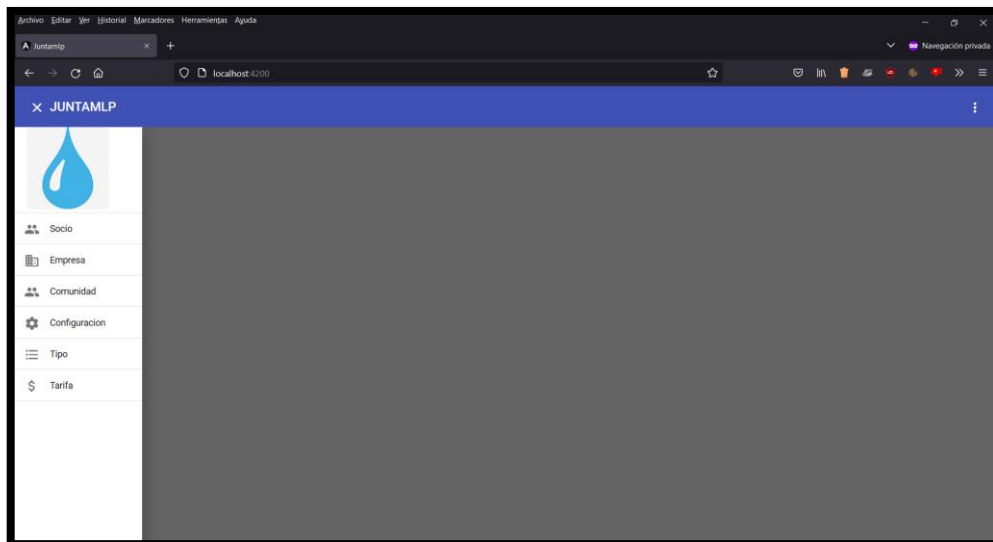


Fig. 12 Demostración de la interfaz principal

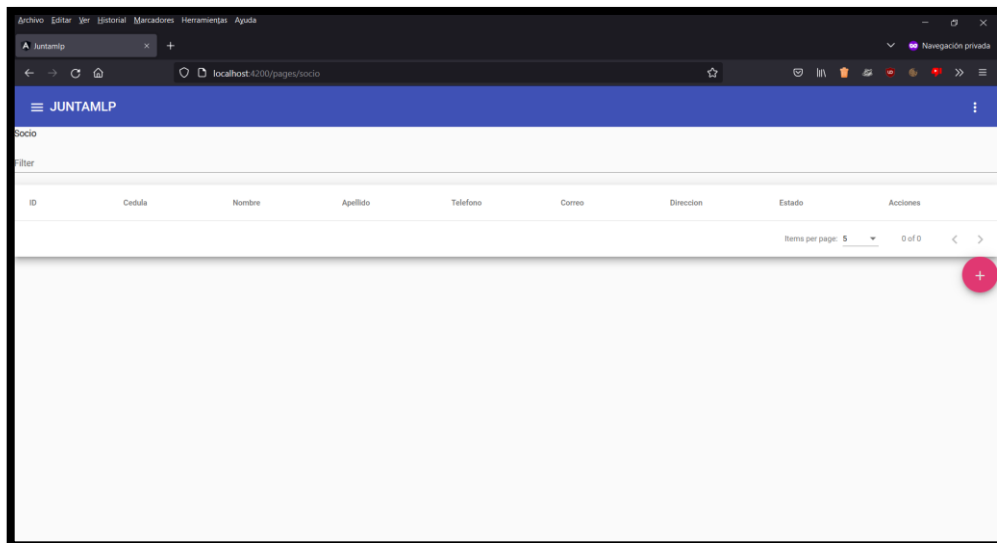


Fig. 13 Demostración de la ficha de Socio

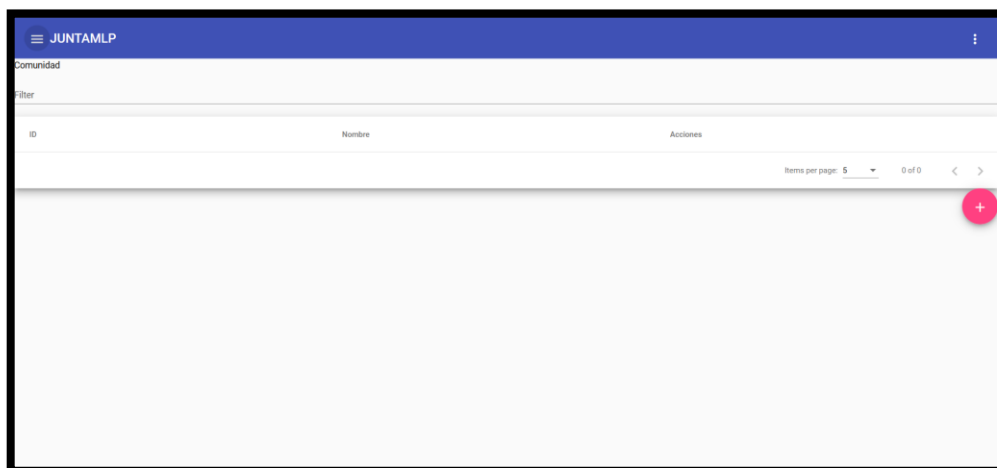


Fig. 14 Demostración de la ficha de Socio Comunidad

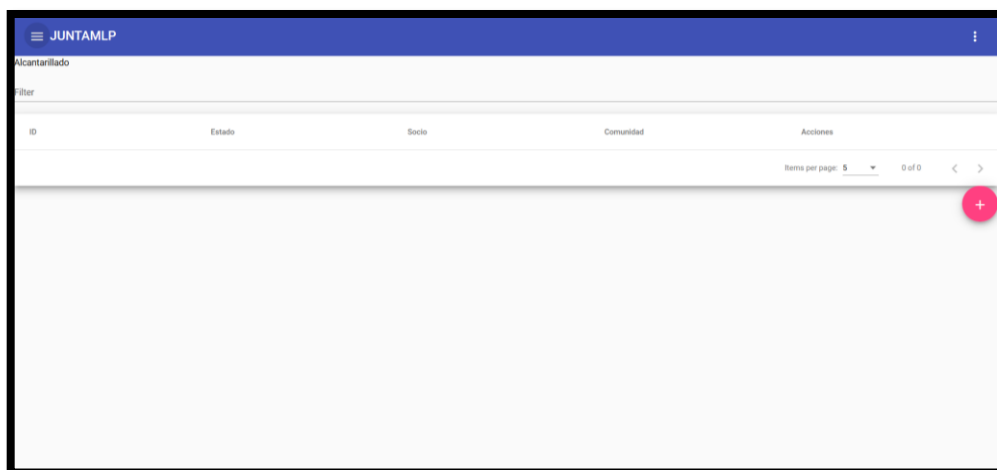


Fig. 15 Demostración de la ficha de Alcantarillado

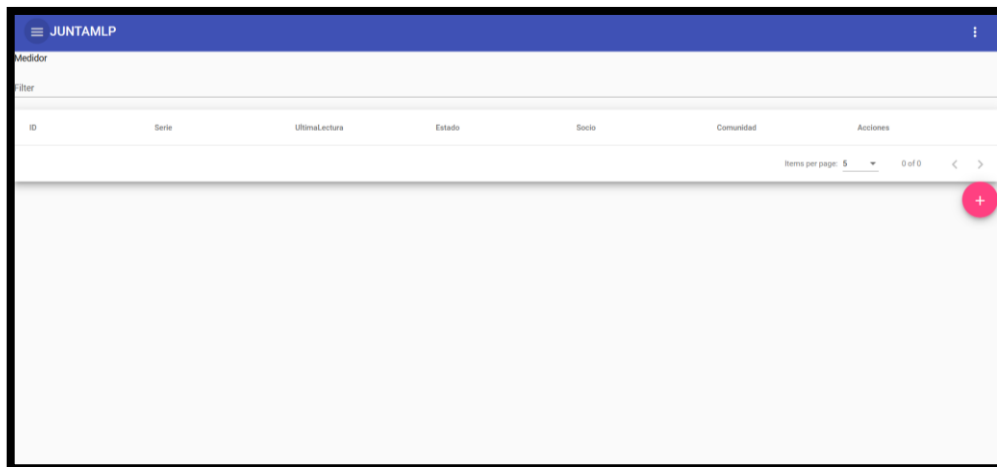


Fig. 16 Demostración de la ficha de Medidor

3.6 Sprint 2

3.6.1 Eventos del Sprint 2: Siguiendo con los eventos en el segundo sprint y en base a los resultados obtenidos en el primer sprint se continúa avanzando en el prototipo para cumplir con las historias de usuario establecidas para el segundo sprint que consisten en el almacenamiento de consumos y pendientes, los avances serán valorados más adelante en el Burndown Chart junto con las **diferentes actividades**.

3.6.2 Roles del Sprint 2: En este sprint intervienen el Product Owner y un desarrollador con el cual se establecieron los requerimientos para poder determinar las tareas a cumplir en este sprint en este caso enfocarse en el mantenimiento de la información de consumos y los pendientes de los socios.

3.6.3 Artefactos del Sprint 2: Una vez se cumpla lo establecido en el segundo sprint se definirá las secciones de consumo y pendientes.

3.6.4 Sprint Backlog 2

En este apartado podemos observar de manera detallada las historias de usuario propuestas para llevar a cabo los sprint número 3 y 4 y en base a ello definir las diferentes tareas que se realizarán para completar las historias de usuario.

Tabla XX Planificación del Sprint 2

Sprint 2	
Autor: Kevin Guaman	
Fecha de Inicio	01/Octubre/2022
Fecha de Fin	31/Octubre/2022
Duración	80 horas
Tareas a desarrollar	Mantenimiento de la información de los consumos Mantenimiento de los valores pendientes de cada socio

Tabla XXI Lista de tareas del Sprint 2

Sprint 2			Fecha Inicio 01-10-2022			Duración 4 semanas(80hrs)		
Delegado/Autor: Kevin Guaman						Días	05-15 oct	16-31 oct
H. Usu	Tarea	Estado	Hr	40	40			
3	Creación de la clase TipoPendiente	Completo				X		
3	Creación de la clase Pendientes	Completo				X		
3	Diseño de las capas de persistencia (JPA)	Completo				X		
3	Creación de las metodologías CRUD	Completo				X		
3	Creación de los DTO y los controladores	Completo				X		
3	Creación de la interfaz gráfica de la clase TipoPendiente y Pendientes	Completo				X		
4	Creación de la clase Consumo de Agua	Completo						X
4	Creación de la clase Lecturas	Completo						X
4	Diseño de las capas de persistencia (JPA)	Completo						X
4	Creación de las metodologías CRUD	Completo						X
4	Creación de los DTO y los controladores	Completo						X
4	Creación de la interfaz gráfica de la clase Lecturas	Completo						X

Tabla XXII Lista de Tareas completadas del Sprint 2

Nombre del Proyecto: JuntaMLP		Numero de tareas: 12		Sprint: 2	
Encargado de tares: Kevin Guaman					
Id	Tarea	Historia de Usuario	de	Fecha	
1	Creación de la clase TipoPendiente	3		06-octubre-2022	
2	Creación de la clase Pendientes	3		07-octubre-2022	
3	Diseño de las capas de persistencia (JPA)	3		09-octubre-2022	
4	Creación de las metodologías CRUD	3		10-octubre-2022	
5	Creación de los DTO y los controladores	3		11-octubre-2022	
6	Creación de la interfaz gráfica de la clase TipoPendiente y Pendientes	3		15-octubre-2022	
7	Creación de la clase Consumo de Agua	4		16-octubre-2022	
8	Creación de la clase Lecturas	4		17-octubre-2022	
9	Diseño de las capas de persistencia (JPA)	4		19-octubre-2022	

10	Creación de las metodologías CRUD	4	20-octubre-2022
11	Creación de los DTO y los controladores	4	22-octubre-2022
12	Creación de la interfaz gráfica de la clase Lecturas	4	29-octubre-2022

3.6.5 BurnDown Chart Sprint 2

En la figura 17 es donde se muestra el Burdow Chart de las actividades del segundo sprint correspondientes a las historias de usuario 3 y 4.

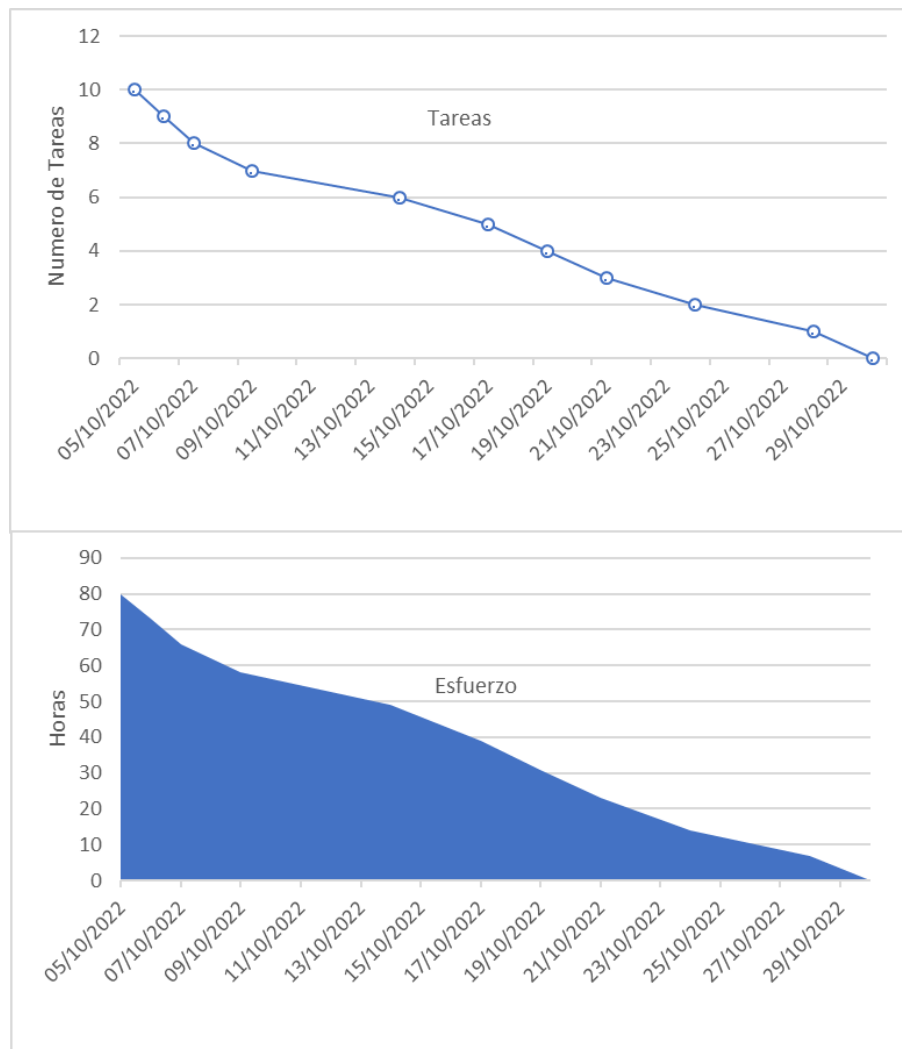
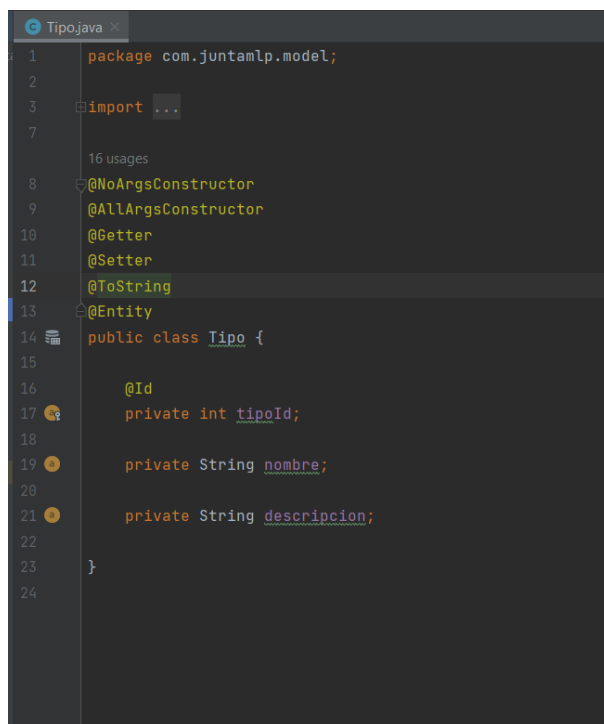


Fig. 17 Burn down chart Sprint 2

3.6.6 Justificación del sprint 2

En esta sección podemos observar los resultados obtenidos de la ejecución del sprint número 2, para lo cual se puede visualizar la figura número 17, donde se grafica el esfuerzo en función del tiempo destinado en cada día.

A screenshot of an IDE window titled 'Tipo.java'. The code is as follows:

```
1 package com.juntamp.model;
2
3 import ...
4
5
6
7
8 @NoArgsConstructor
9 @AllArgsConstructor
10 @Getter
11 @Setter
12 @ToString
13 @Entity
14 public class Tipo {
15
16     @Id
17     private int tipoId;
18
19     private String nombre;
20
21     private String descripcion;
22
23 }
24
```

Fig. 18 Generación de la clase Tipo Pendiente

```

1 package com.juntamp.model;
2
3 import ...
4
5
6
7 @NoArgsConstructor
8 @AllArgsConstructor
9 @Getter
10 @Setter
11 @ToString
12 @Entity
13 public class Pendiente {
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private int pendienteId;
17
18     private int cantidad;
19
20     private String descripcion;
21
22     private double precio;
23
24     private String grabaIva;
25
26     private String pagado;
27
28     @ManyToOne
29     @JoinColumn(name = "tipo_id", nullable = false)
30     private Tipo tipo;
31 }

```

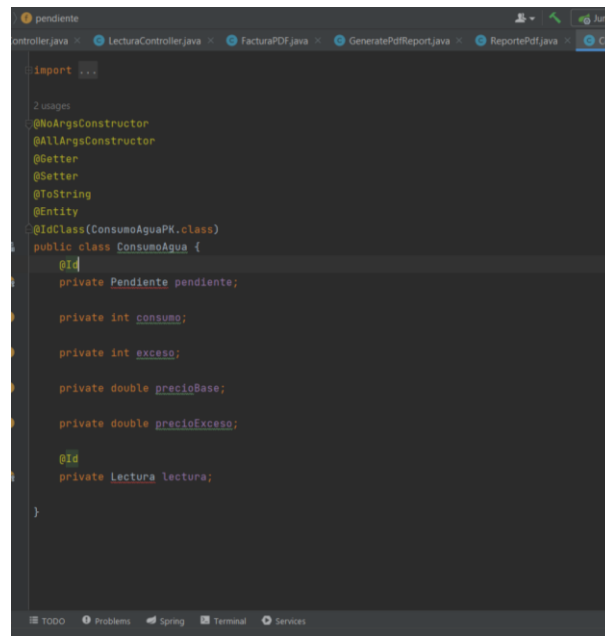
Fig. 19 Generación de la clase Pendiente

```

1 package com.juntamp.model;
2
3 import ...
4
5
6
7 @NoArgsConstructor
8 @AllArgsConstructor
9 @Getter
10 @Setter
11 @ToString
12 @Entity
13 public class Lectura {
14     @Id
15     @GeneratedValue(strategy = GenerationType.IDENTITY)
16     private int lecturaId;
17
18     private String mes;
19
20     private int año;
21
22     private int anterior;
23
24     private int actual;
25
26     @ManyToOne
27     @JoinColumn(name = "medidor_id", nullable = false)
28     private Medidor medidor;
29
30 }
31

```

Fig. 20 Generación de la clase Lectura



```
import javax.persistence.*;

import java.util.*;

@NoArgsConstructor
@AllArgsConstructor
@Getter
@Setter
@ToString
@Entity
@IdClass(ConsumoAguaPK.class)
public class ConsumoAgua {
    @Id
    private Pendiente pendiente;

    private int consumo;

    private int exceso;

    private double precioBase;

    private double precioExceso;

    @Id
    private Lectura lectura;
}
```

Fig. 21 Generación de la clase ConsumoAgua

3.6.7 Entregable Sprint 2

A continuación se pueden observar las figuras correspondientes a los resultados del segundo sprint donde se muestra las secciones de lecturas y pendientes:

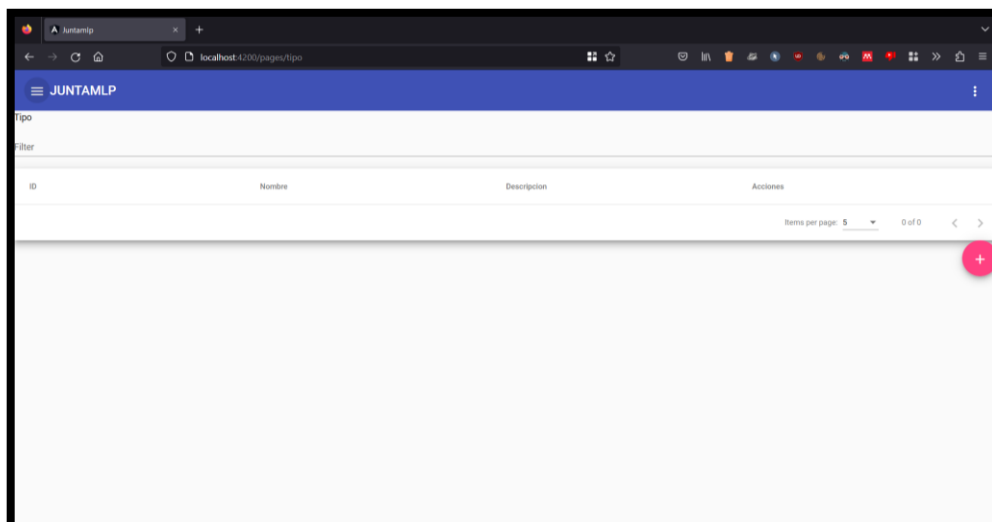


Fig. 22 Demostración de la ficha de TipoPendiente

The screenshot shows the 'Buscar Pendientes' (Search Pending) interface. At the top, there is a blue header with the JUNTAMLP logo and a menu icon. Below the header, the breadcrumb trail reads 'Factura' > 'Socio'. A blue button labeled 'Buscar Pendientes' is visible. The main area contains a table with the following columns: 'ID', 'Cantidad', 'Descripción', and 'precio'. The table is currently empty. On the right side of the table, there is a pagination control showing 'Items per page: 5' and '0 of 0', along with navigation arrows. A blue circular icon with a white document symbol is located in the bottom right corner of the table area.

Fig. 23 Demostración de la ficha de Pendiente

The screenshot shows the 'Lectura' (Reading) interface. At the top, there is a blue header with the JUNTAMLP logo and a menu icon. Below the header, the breadcrumb trail reads 'Lectura' > 'Filtro'. A red circular icon with a white plus sign is located in the top right corner. The main area contains a table with the following columns: 'ID', 'Año', 'Mes', 'Medidor', 'Anterior', 'Actual', and 'Acciones'. The table is currently empty. On the right side of the table, there is a pagination control showing 'Items per page: 5' and '0 of 0', along with navigation arrows.

Fig. 24 Demostración de la ficha Lectura

3.7 Sprint 3

3.7.1 Eventos del Sprint 3: con los eventos en el tercer sprint se continúa avanzando en el prototipo para cumplir con las historias de usuario establecidas para el tercer sprint que consiste en la generación de facturas y reportes y los avances serán valorados más adelante en el Burndown Chart junto con las diferentes actividades.

3.7.2 Roles del Sprint 3: En este sprint intervienen el Product Owner y un desarrollador con el cual se establecieron los requerimientos para poder determinar las tareas a cumplir en este sprint en este caso enfocarse en la generación de reportes y facturas.

3.7.3 Artefactos del Sprint 3: Una vez se cumpla lo establecido en el tercer sprint el prototipo sería capaz de generar las diferentes facturas de cada socio con su respectivo servicio y realizar informes mensuales para el análisis de información.

3.7.4 Sprint backlog 3

En esta sección se establecen las historias de usuario correspondientes al tercer sprint con el objetivo de definir las diferentes tareas que se realizaran para completar las historias de usuario:

Tabla XXIII Planificación del Sprint 3

Sprint 3	
Autor: Kevin Guaman	
Fecha de Inicio	01/Noviembre/2022
Fecha de Fin	31/Diciembre/2022
Duración	80 horas
Tareas a desarrollar	Generación de Reportes Generación de facturas

Tabla XXIV Lista de tareas del Sprint 3

Sprint 3			Fecha Inicio 01-10-2022			Duración 8 semanas(80hrs)		
Delegado/Autor: Kevin Guaman						Días	01-30 Nov	01-31 Dic
H. Usu	Tarea	Estado	Hr	40	40			
5	Creación de la clase Factura Cabecera y Factura Detalle	Completo		X				
5	Creación de las clases auxiliares (Tarifas, Iva, Numero de factura y Empresa)	Completo		X				
5	Diseño de las capas de persistencia (JPA)	Completo		X				
5	Creación de las metodologías CRUD	Completo		X				
5	Creación de los DTO y los controladores	Completo		X				
5	Creación de la interfaz gráfica de las clases de Facturación	Completo		X				
6	Creación de la interfaz gráfica de las clases de Facturación	Completo						X
6	Creación de la clase Factura	Completo						X
6	Creación de clase Reporte	Completo						X
6	Creación de las clases para la generación de facturas y reportes	Completo						X
6	Creación de una página para las facturas generadas	Completo						X
6	Creación de una página para la generación de reportes	Completo						X
6	Pruebas para la impresión de facturas y generación de reportes	Completo						X

Tabla XXV Lista de Tareas completadas del Sprint 3

Nombre del Proyecto: JuntaMLP		Numero de tareas:		Sprint: 3		Encargado	
de tares: Kevin Guaman							
Id	Tarea	Historia de Usuario	Fecha				
1	Creación de la clase Factura Cabecera y Factura Detalle	5	05-noviembre-2022				
2	Creación de las clases auxiliares (Tarifas, Iva, Numero de factura y Empresa)	5	07-noviembre-2022				
3	Diseño de las capas de persistencia (JPA)	5	08-noviembre-2022				
4	Creación de las metodologías CRUD	5	10-noviembre-2022				
5	Creación de los DTO y los controladores	5	13-noviembre-2022				
6	Creación de la interfaz gráfica de las clases de Facturación	5	27-noviembre-2022				
7	Creación de la interfaz gráfica de las clases de Facturación	6	02-diciembre-2022				
8	Creación de la clase Factura	6	05-diciembre-2022				
9	Creación de clase Reporte	6	09-diciembre-2022				
10	Creación de las clases para la generación de facturas y reportes	6	14-diciembre-2022				
11	Creación de una página para las facturas generadas	6	28-diciembre-2022				
12	Creación de una página para la generación de reportes	6	31-diciembre-2022				

3.7.5 BurnDown Chart Sprint 3

En la figura 3.20 es donde se muestra el Burdow Chart de las actividades del segundo sprint correspondientes a las historias de usuario 5 y 6.

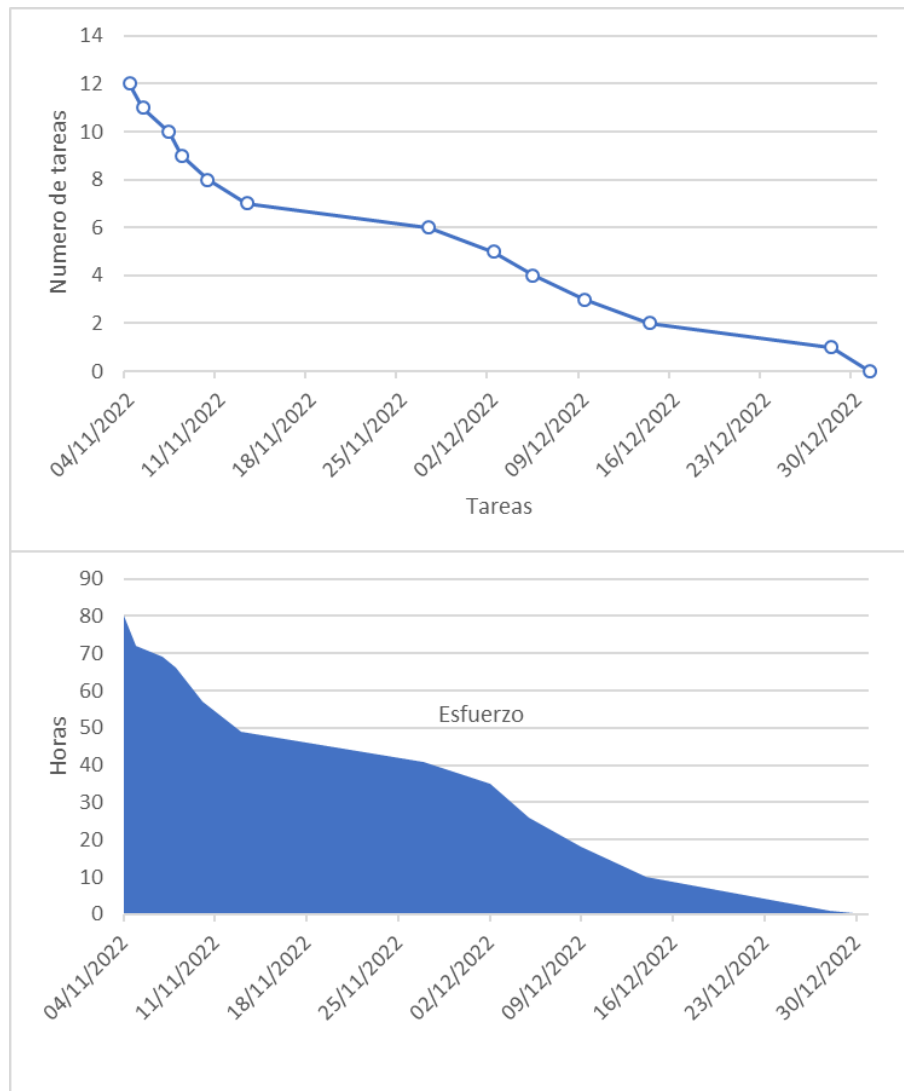


Fig. 25 Burn down chart Sprint 3

3.7.6 Justificación del sprint 3

En esta sección podemos observar las gráficas que comprenden los resultados correspondientes al tercer sprint.

```

@Setter
@ToString
@Entity
public class FacturaCabecera {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int cabeceraId;

    private LocalDate fecha;

    private double subTotalB;

    private double subTotalIVA;

    private double valorIVA;

    private double ivaCalculado;

    private double total;

    @ManyToOne
    @JoinColumn(name = "id_socio", nullable = false)
    private Socio socio;

    private int numeroFactura;

    @OneToMany(mappedBy = "facturaCabecera", cascade = {CascadeType.ALL}, orphanRemoval = true, fetch = FetchType.LAZY)
    private List<FacturaDetalle> facturasDetalle;
}

```

Fig. 26 Generación de la clase Factura Cabecera

```

@Entity
public class FacturaDetalle {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int detalleId;

    private int cantidad;

    private String descripcion;

    private double pu;

    private double subTotal;

    private String graba_iva;

    @ManyToOne
    @JoinColumn(name = "id_factura_cabecera", nullable = false, foreignKey = @ForeignKey(name = "FK_CABECERA_DETALLE"))
    @JsonIgnore
    private FacturaCabecera facturaCabecera;

    @OneToOne
    @JoinColumn(name = "id_pendiente", nullable = false)
    @JsonIgnore
    private Pendiente pendiente;
}

```



Fig. 27 Generación de la clase Factura Detalle

3.7.7 Entregable Sprint 3

En la siguiente sección se adjuntan las figuras correspondientes a los resultados del tercer sprint donde se muestran las secciones de facturación.

Socio

Filter


ID	Cédula	Nombre	Apellido	Telefono	Correo	Direccion	Estado	Acciones
1	0302366281	Kevin	Guaman	0900000006	kguaman@gmail.com	Cuenca	activo	 

Items per page: 5 1 - 1 of 1 < >

Fig. 28 Demostración del ingreso de un socio

Lectura

Filter

ID	Año	Mes	Medidor	Anterior	Actual	Acciones
1	2023	Enero	xxx	150	166	

Items per page: 5 1 - 1 of 1 < >

Fig. 29 Demostración de ingreso de datos en la ficha de Lectura

Factura

Usuario: Kevin Guaman

Buscar Pendientes

ID	Cantidad	Descripción	precio
1	1	Alcantarillado Enero 2023	1
2	1	Agua Enero 2023 ANT: 150 ACT: 166 CON: 16	3.15

Items per page: 5 1 - 2 of 2 < >

Fig. 30 Demostración de resultados de pendientes de un Usuario

JUNTA ADMINISTRADORA DE AGUA POTABLE Y SANEAMIENTO MONSEÑOR LEONIDAS PROAÑO GESTIÓN COMUNITARIA DEL AGUA Y ALCANTARILLADO Dirección: Guapán. Telf.: 2207208			
Transacción: 1 Fecha: 2023-02-10 Socio: Kevin Guaman Cédula: 0302366281		Telefono: 0900000006 Dirección: Azogues	

CANT	DETALLE	V.U.	TOTAL
1	Alcantarillado Febrero 2023	1.0	1.0
1	Agua Febrero 2023 ANT: 150 ACT: 180 CON: 30	7.45	7.45
		Subtotal 0%: 8.45	
		Subtotal IVA: 0.0	
		IVA: 0.0	
		Total: 8.45	

Forma de pago: EFECTIVO = 8.45

Recibi conforme

Firma Autorizada

Fig. 31 Demostración de la ficha de Factura de un Usuario

Reporte de facturación del mes de: Enero	
TOTAL FACTURADO	8.45
Recibi conforme	Entrego conforme

Fig. 32 Demostración de la ficha de Reportes

3.8 Sprint 4

3.8.1 Eventos del Sprint 4: gracias a los eventos en este sprint y en base a los resultados obtenidos en el primer sprint se continúa avanzando a la etapa final con la carga del prototipo a un servicio en la nube para cumplir con las historias de usuario establecidas para el cuarto sprint y los avances serán valorados más adelante en el Burndown Chart junto con las diferentes actividades.

3.8.2 Roles del Sprint 4: En este sprint intervienen el Product Owner y un desarrollador con el cual se establecieron los requerimientos para poder determinar las tareas a cumplir en este sprint en este caso se realiza la carga del sistema en la nube para su acceso público y almacenamiento de información.

3.8.3 Artefactos del Sprint 4: Una vez se cumpla lo establecido en el cuarto sprint el prototipo estaría finalizado y listo para iniciar la fase de pruebas.

3.8.4 Sprint Backlog 4

Esta sección está destinada a contemplar las historias de usuario correspondientes al cuarto sprint; donde se integran las historias de usuario 7 y 8 para definir las tareas.

Tabla XXVI Planificación del Sprint 4

Sprint 4	
Autor: Kevin Guaman	
Fecha de Inicio	01/Enero/2023
Fecha de Fin	31/Enero/2023
Duración	80 horas
Tareas a desarrollar	Almacenamiento

Tabla XXVII Lista de tareas del Sprint 4

Sprint	Fecha Inicio	Duración	Días	01-15 Ene	16-31 Ene
4	01-01-2023	8 semanas(80hrs)			
Delegado/Autor: Kevin Guaman					
H. Usu	Tarea	Estado	Hr	40	40
7	Creación de una cuenta AWS	Completo		X	
7	Creación de una instancia en AWS	Completo		X	
7	Configuración de la Instancia	Completo		X	
7	Subir el programa a la instancia	Completo			X
7	Pruebas del programa en la instancia en la nube	Completo			X

Tabla XXVIII Lista de Tareas completadas del Sprint 4

Nombre del Proyecto: JuntaMLP		Numero de tareas:		Sprint:
Encargado de tareas: Kevin Guaman				
Id	Tarea	Historia Usuario	de	Fecha
1	Creación de una cuenta AWS	7		03-enero-2023
2	Creación de una instancia en AWS	7		06-enero-2023
3	Configuración de la Instancia	7		09-enero-2023
4	Subir el programa a la instancia	7		29-enero-2023
5	Pruebas del programa en la instancia en la nube	7		30-enero-2023

3.8.5 BurnDown Chart Sprint 4

En la figura 3.28 es donde se muestra el Burdow Chart de las actividades del cuarto sprint correspondientes a las historias de usuario 5 y 6.

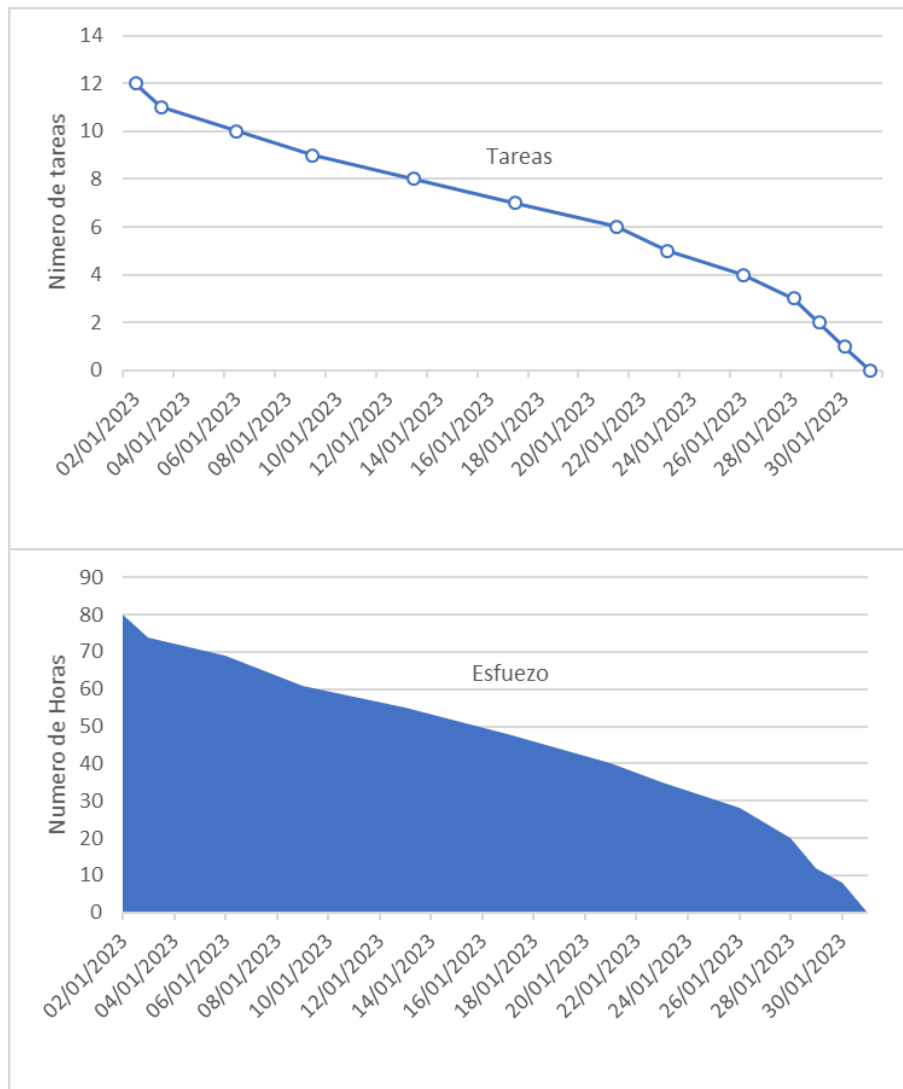


Fig. 33 Burn down chart Sprint 4

3.8.6 Justificación del sprint 4

En esta sección se muestra los resultados del cuarto sprint con las siguientes figuras se puede contemplar los resultados obtenidos.

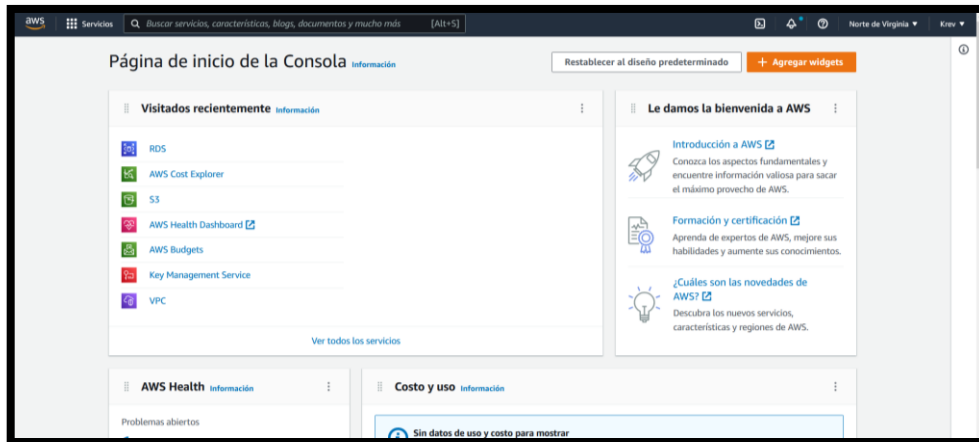


Fig. 34 Creación da la cuneta en aws

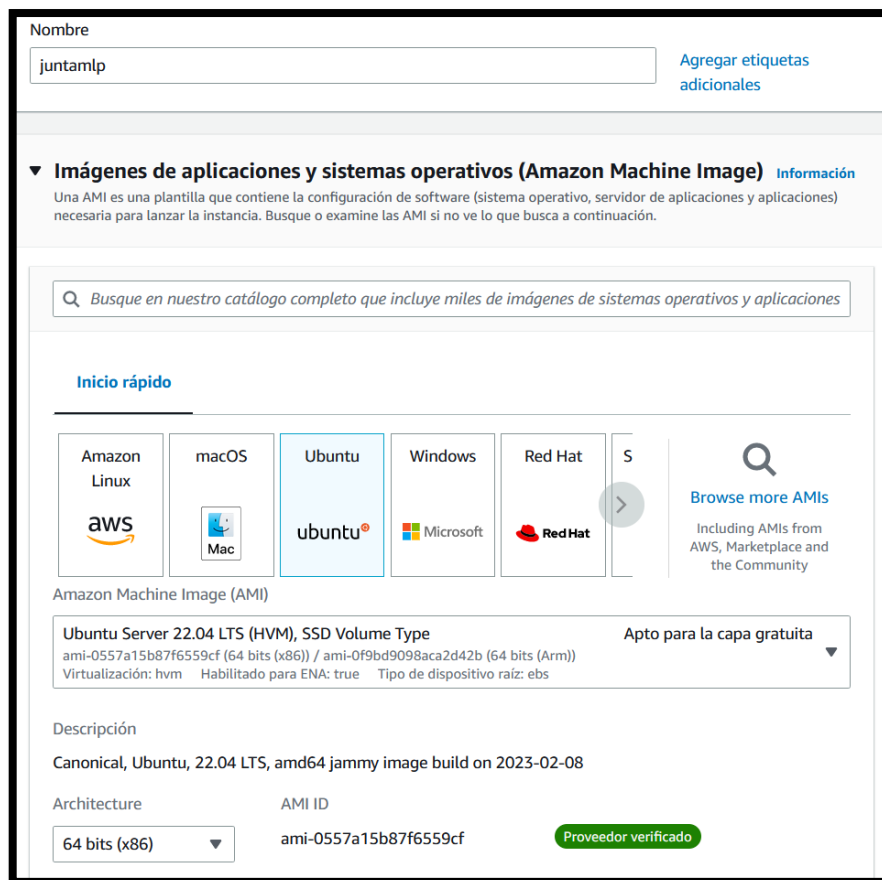


Fig. 35 Creación de una Instancia en AWS

▼ Configurar almacenamiento Información Avanzado

1x GiB Volumen raíz (Sin cifrar)

ⓘ Los clientes que cumplan los requisitos de la capa gratuita pueden obtener hasta 30 GB de almacenamiento magnético o de uso general (SSD) de EBS ✕

Agregar un nuevo volumen

La AMI seleccionada contiene más volúmenes de almacén de instancias de los que permite la instancia. Solo se podrá obtener acceso desde la instancia a los primeros 0 volúmenes de almacén de instancias de la AMI

0 x sistemas de archivos Editar

Fig. 36 Configuración de la Instancia

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\AP\angularp> ng build
✓ Browser application bundle generation complete.
✓ Copying assets complete.
✓ Index html generation complete.

main.c9f2f02b97add7e9.js | main | 167.88 kB | 45.21 kB
polyfills.974cbae822cc61b6.js | polyfills | 33.07 kB | 10.63 kB
runtime.d4ef1bbee9d9bdcc.js | runtime | 1.04 kB | 602 bytes
styles.ef46db3751d8e999.css | styles | 0 bytes | -

```

Fig. 37 Compilación del Frontend

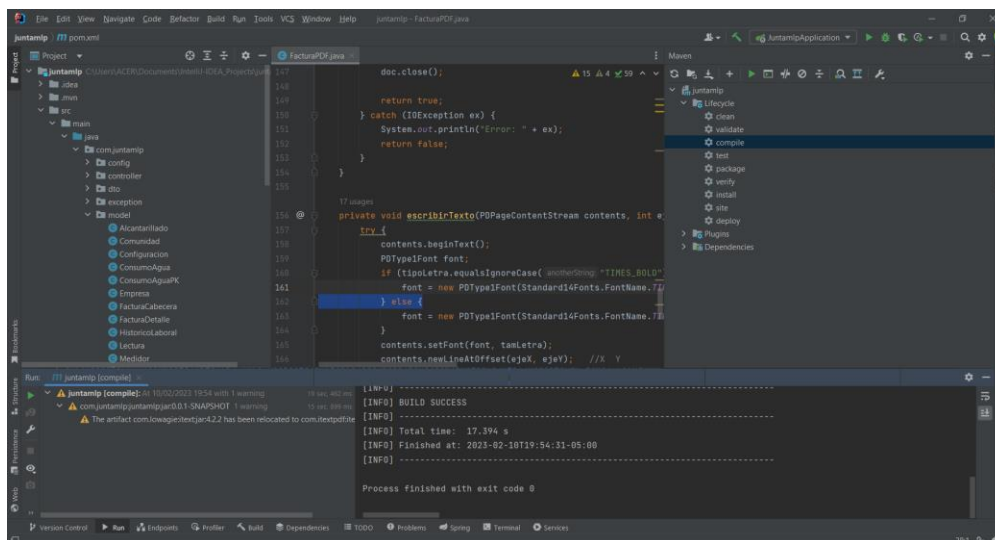


Fig. 38 Compilación del Backend

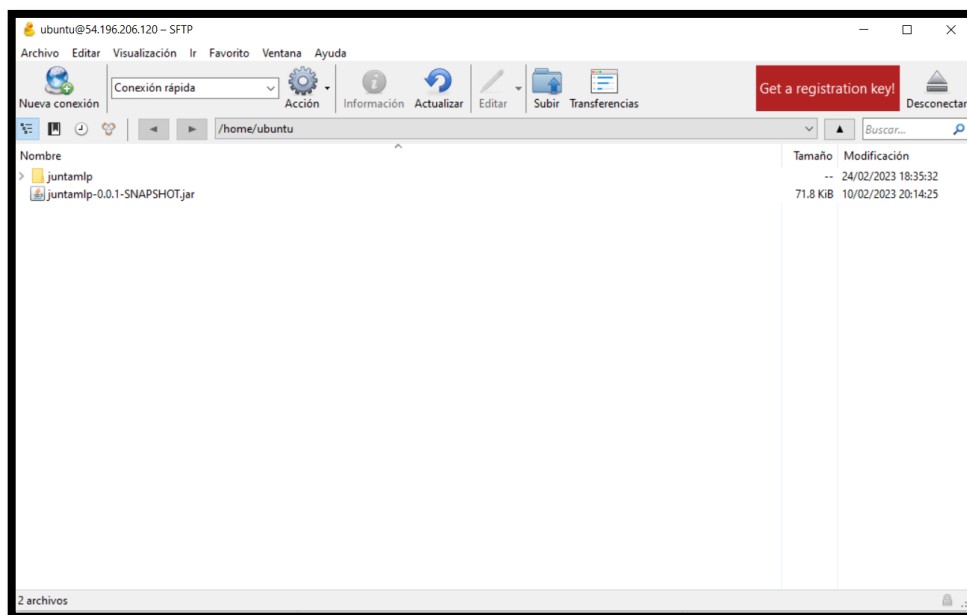


Fig. 39 Carga de los archivos compilados

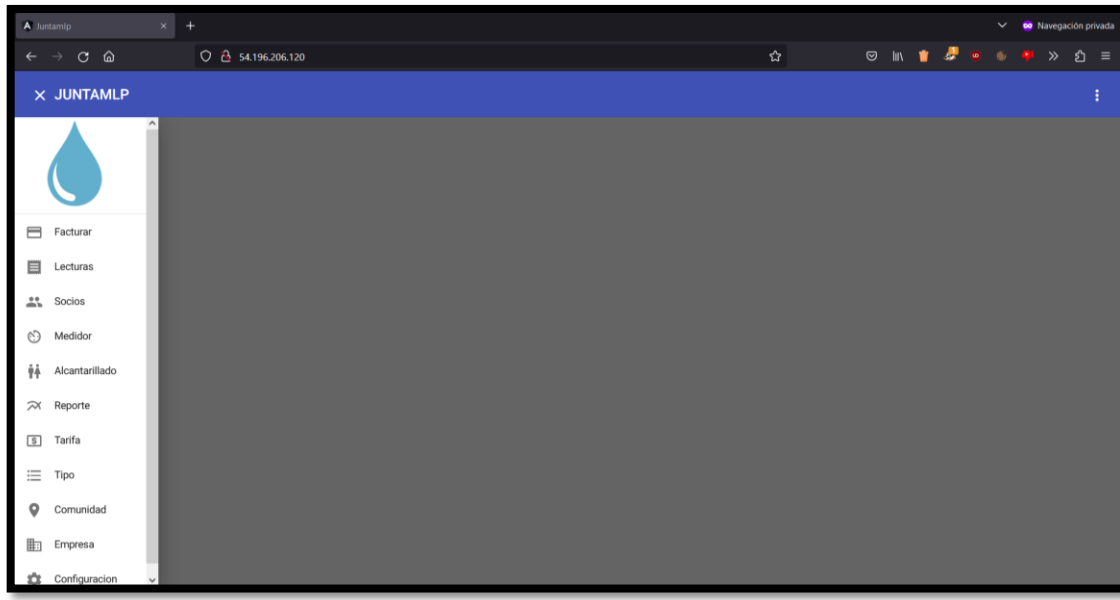


Fig. 42 Resultados del servicio Levantado

3.9 Pruebas del prototipo

Una vez concluido el desarrollo del prototipo se procede a realizar diferentes pruebas con el área de facturación de la Junta de Agua. La realización de pruebas del es una etapa crucial en el proceso de desarrollo de software. Las pruebas permitieron validar la funcionalidad de prototipo y asegurar el cumplimiento de los requisitos establecidos.

Se optó por realizar pruebas de aceptación para evaluar la funcionalidad, usabilidad y rendimiento, las cuales dieron como resultado la aceptabilidad de versión actual del prototipo por del usuario. A continuación, se detalla informe de las pruebas realizadas (Ver Tabla XXIX)

Informe de resultados de las pruebas			
Tareas establecidas	Respuesta del programa	Cumple con la tarea establecida	Observaciones
- Almacenamiento de la información de los socios dentro de una base de datos.	Carga la información	Si	Ninguna
- Capacidad de realizar reportes de forma diaria, mensual y anual para el análisis de datos.	Carga la información	Si	Ninguna
- Capacidad de impresión de facturas.	Imprime facturas normalmente	Si	Ninguna
- Almacenamiento de los datos de los medidores de los socios.	Carga la información	Si	Ninguna
- Almacenamiento de las lecturas de los medidores de los socios.	Carga la información	Si	Ninguna
- Almacenamiento de la información sobre el consumo de cada socio.	Carga la información	Si	Ninguna
- Registro y guardado del RUC de las facturas.	Carga la información	Si	Ninguna

Tabla XXIX Informe de resultados de las pruebas aplicadas.

CAPITULO 4. RESULTADOS OBTENIDOS

4.1 Resultados

A lo largo del desarrollo de sistema se logran obtener varios resultados que se resumen en los siguientes puntos:

Gestión de la información de los usuarios: el prototipo cumple con los parámetros de integridad y consistencia de la información, debido que se usó el motor de la base de datos de AWS y que el diseño se realizó con un modelo entidad relación en el cual en cada entidad se especificó el tamaño, tipo de dato, campos clave y sus relaciones.

Agilizar el proceso de búsqueda: para reducir los tiempos de búsqueda de los pendientes de un usuario, el sistema ofrece la capacidad de realizar búsquedas en cada página de un elemento específico ya sea un socio, una lectura, un medidor, un pendiente, entre otros.

Almacenamiento en la nube: Además de que el prototipo ofrece la posibilidad de obtener información ordenada, y rápida de cada entidad, lo que agiliza las consultas y la búsqueda de información; el alojamiento en la nube con AWS incrementa la seguridad, escalabilidad, flexibilidad, respaldos automáticos entre otros.

Eficiencia: ahora se puede realizar facturas de forma eficiente y rápida, así como consultar los valores pendientes que tenga un socio y la capacidad de seleccionar qué valores desea el socio que se le sean facturados.

Generación de Reportes: El prototipo permite generar reportes de manera sencilla, puesto que se mostrará la información general de los pagos que se realizan mensual y anualmente. Los resultados obtenidos facilitarán el análisis de la información y la detección de inconsistencias con los pagos de los socios.

4.2 Conclusiones

- Después de revisar y analizar la fundamentación teórica, se pudo determinar las herramientas y metodologías necesarias para el desarrollo del prototipo de software. Todos estos elementos desempeñan un rol fundamental, por ello, es importante elegir correctamente las herramientas que se utilizaran a lo largo del proyecto, entre ellas: la metodología de desarrollo de software que es fundamental para llevar a cabo las diferentes etapas de generación del aplicativo, las herramientas de programación necesarias para la creación del frontend y el backend del prototipo y las herramientas de Cloud Computing para el alojamiento y almacenamiento de datos.
- Una vez concluida con la etapa de desarrollo, se pudo evidenciar que efectivamente la metodología SCRUM es apropiada para la creación de prototipos por ser de naturaleza iterativa e incremental, facilitando la sistematización y organización del trabajo para producir las funcionalidades que se establecieron hasta conseguir la versión actual del prototipo. Una parte fundamental de esta metodología fueron los eventos, donde el tiempo destinado para este fin, fue aprovechado para realizar reuniones tanto virtuales como presenciales, con las personas encargadas de facturación para poner en su conocimiento los resultados obtenidos y validar los datos obtenidos como resultado de las operaciones.
- Una vez realizadas las pruebas de aceptación con las autoridades de la Junta de Agua podemos concluir que este proceso ha sido fundamental para el prototipo ya que permite encontrar problemas de funcionalidad o accesibilidad, lo que a su vez contribuirá a ofrecer nuevas mejoras al prototipo en un futuro. Las pruebas también han proporcionado información valiosa sobre el rendimiento y usabilidad. En este caso el prototipo logra

cubrir los requisitos y expectativas del usuario que fueron establecidas en el levantamiento de requerimientos pero este carece aun de funciones para que llegue a ser un software operativo.

4.3 Recomendaciones

- El prototipo de software desarrollado cumple con los requerimientos levantados en base a las necesidades de la Junta de Agua “Monseñor Leónidas Proaño”. por tanto, a la universidad se recomienda dar continuidad a este proyecto para mejorar este proyecto.
- Se recomienda incorporar nuevas funciones de seguridad como un login para los usuarios la junta, de esa manera resguardar la información personal de los socios y de la junta. A la vez implementar la facturación electrónica para facilitar la gestión y el control de los registros contables y tributarios.
- Debido a la rápida evolución de las redes sociales y sistemas de comunicación a través de internet, se recomienda implementar mensajería instantánea en el proyecto para comunicar a los socios sobre datos particulares de manera rápida y eficiente para lo cual se podría hacer uso del Cloud Api de Whastapp.

Bibliografía

- [1] M. Fonte and C. Ranaboldo, “Desarrollo rural, territorios e identidades culturales. Perspectivas desde America Latina y la union Europea.,” *Rev. Opera*, no. 7, pp. 9–32, 2007.
- [2] M. C. López Pérez, “EL USO DE LAS TICS Y SU INCIDENCIA EN EL PROCESO DE ENSEÑANZA APRENDIZAJE DE LOS ESTUDIANTES DE SEGUNDO A SÉPTIMO AÑO DE EDUCACIÓN BÁSICA DE LA EDUCACIÓN BÁSICA DE LA ESCUELA RUBÉN SILVA DEL CANTÓN PATATE RUBÉN SILVA DEL CANTÓN PATATE PROVINCIA DE TUNGURA,” Ambato, 2012.
- [3] J. Pérez Porto and M. Merino, “Definición de ruralidad,” 2014.
<https://definicion.de/ruralidad/> (accessed May 03, 2022).
- [4] “¿Qué son los servicios de nube?” <https://www.redhat.com/es/topics/cloud-computing/what-are-cloud-services> (accessed May 03, 2022).
- [5] Á. Arias and I. T. C. Academy, *Computación en la Nube: 2ª Edición*. sn, 2015.
- [6] M. Pozo Benitez, “AWS: El mayor proveedor de Cloud para Startups,” 2021.
- [7] V. L. Lazo Dután, “Implementación del sistema contable para la junta administradora de agua potable Mobiloil & Mosquera.,” Azogues, 2010. Accessed: May 03, 2022. [Online]. Available: <https://dspace.ucacue.edu.ec/handle/ucacue/2524>.
- [8] J. A. Garay Bravo and S. M. Moscoso Cevallos, “Aplicación móvil nativa en Android 2.2-4.0.3 para dinamizar las consultas de facturación de Empresas de Servicios Básicos y Telecomunicaciones para evitar las multas o cortes inesperados,” Universidad Católica de Santiago de Guayaquil, Guayaquil, 2013.

- [9] M. J. Sánchez González, “Desarrollo e implementación de sistema web de registro de pagos de alícuotas para automatización de control de acceso de vehículos en urbanización privada.,” Universidad Católica de Santiago de Guayaquil, Guayaquil, 2019.
- [10] J. A. Peña Peña and D. X. González Miranda, “Sistema de informacion web para la gestión de datos referente a la recolecc...: Metabuscador Universidad Católica de Cuenca,” Universidad Católica de Cuenca, Azogues, 2021.
- [11] P. Prescott and A. Gutiérrez Sánchez, *HTML 5*. Babelcube Inc, 2015.
- [12] “¿Qué es Java y por qué lo necesito?”
https://www.java.com/es/download/help/whatis_java.html (accessed Sep. 21, 2022).
- [13] J. R. M. Ríos, N. M. L. Mora, M. P. Z. Ordóñez, and E. L. L. Sojos, “Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python,” *Rev. Latinoam. Ing. Softw.*, vol. 4, no. 4, pp. 201–207, Sep. 2016, doi: 10.18294/RELAIS.2016.201-207.
- [14] M. Miranda Chinlli and C. G. Rochina Rochina, “Análisis de los Frameworks Yii y Zend para mejorar la Usabilidad y la Compatibilidad del Sitio Web de la Cooperativa de Ahorros y Crédito Minga Ltda.,” Aug. 2015, Accessed: Sep. 21, 2022. [Online]. Available: <http://dspace.epoch.edu.ec/handle/123456789/4581>.
- [15] C. Mateu, *Desarrollo de aplicaciones web*, 1st ed., vol. 1. Catalunya: Fundació para la Universitat Oberta de Catalunya, 2004., 2004.
- [16] S. Graciela, P. Ibarra, R. Quispe, F. F. Mullicundo, D. A. Lamas, and L. Presente, “Herramientas y tecnologías para el desarrollo web desde el FrontEnd al BackEnd,” *XXIII Work. Investig. en Ciencias la Comput. (WICC 2021, Chilecito, La Rioja)*, no.

- August 2021, pp. 963–968, 2021, Accessed: Sep. 21, 2022. [Online]. Available: <http://sedici.unlp.edu.ar/handle/10915/120476>.
- [17] I. Fernandez-Corbaton, X. Zambrana-Puyalto, and G. Molina-Terriza, “Helicity and angular momentum: A symmetry-based framework for the study of light-matter interactions,” *Phys. Rev. A - At. Mol. Opt. Phys.*, vol. 86, no. 4, Oct. 2012, doi: 10.1103/PHYSREVA.86.042103/FIGURES/4/MEDIUM.
- [18] E. Haro, T. Guarda, A. O. Z. Peñaherrera, and G. N. Quiña, “Desarrollo backend para aplicaciones web, Servicios Web Restful: Node.js vs Spring Boot TT - Backend development for web applications, Restful Web Services: Node.js vs Spring Boot,” *Rev. Ibérica Sist. e Tecnol. Informação*, no. E17, pp. 309–321, Jan. 2019, [Online]. Available: <https://www.proquest.com/scholarly-journals/desarrollo-backend-para-aplicaciones-web/docview/2195127847/se-2>.
- [19] P. Plecinski, N. Bokla, T. Klymkovych, M. Melnyk, and W. Zabierowski, “Comparison of Representative Microservices Technologies in Terms of Performance for Use for Projects Based on Sensor Networks,” *Sensors (Basel)*, vol. 22, no. 20, Oct. 2022, doi: 10.3390/S22207759.
- [20] “Funcionalidades - IntelliJ IDEA.” <https://www.jetbrains.com/es-es/idea/features/> (accessed Feb. 06, 2023).
- [21] M. Kimberlin, “Reducing Boilerplate Code with Project Lombok.” <https://objectcomputing.com/resources/publications/sett/january-2010-reducing-boilerplate-code-with-project-lombok> (accessed Jan. 30, 2023).
- [22] J. F. Smart and others, “An introduction to Maven 2,” *JavaWorld Mag. Available*

<http://www.javaworld.com/javaworld/jw-12-2005/jw-1205-maven.html>, p. 27, 2005.

- [23] E. Jendrock, R. Cervera-Navarro, I. Evans, K. Haase, and W. Markito, *The Java EE 7 Tutorial, Band 1*, 5th ed., vol. 1. Addison-Wesley Professional, 2014.
- [24] “PostgreSQL: About.” <https://www.postgresql.org/about/> (accessed Sep. 21, 2022).
- [25] “Ingeniería del Software StarUML,” Accessed: Sep. 21, 2022. [Online]. Available: <http://cnx.org/content/m15092/latest/>.
- [26] “home – Instituto Nacional de Estadística y Censos.” <https://www.ecuadorencifras.gob.ec/institucional/home/> (accessed Mar. 20, 2023).

CAPITULO 5. Anexos

CHH
10 de 07. 2022


Levantamiento de requerimientos

En este documento se presenta una lista de las diferentes funcionalidades con las que el software ofrecido contara y que se realizaran durante el proceso de desarrollo para que finalmente se pueda entregar un producto de buena calidad al cliente, en este caso se entregara para el servicio de facturación de la junta de agua Monseñor Leónidas Proaño. El programa contara con los siguientes elementos:

- Almacenamiento de la información de los socios dentro de una base de datos.
- Capacidad de realizar reportes de forma diaria, mensual y anual para el análisis de datos.
- Capacidad de impresión de facturas.
- Almacenamiento de los datos de los medidores de los socios.
- Almacenamiento de las lecturas de los medidores de los socios.
- Almacenamiento de la información sobre el consumo de cada socio en.
- Registro y guardado del RUC de las facturas.

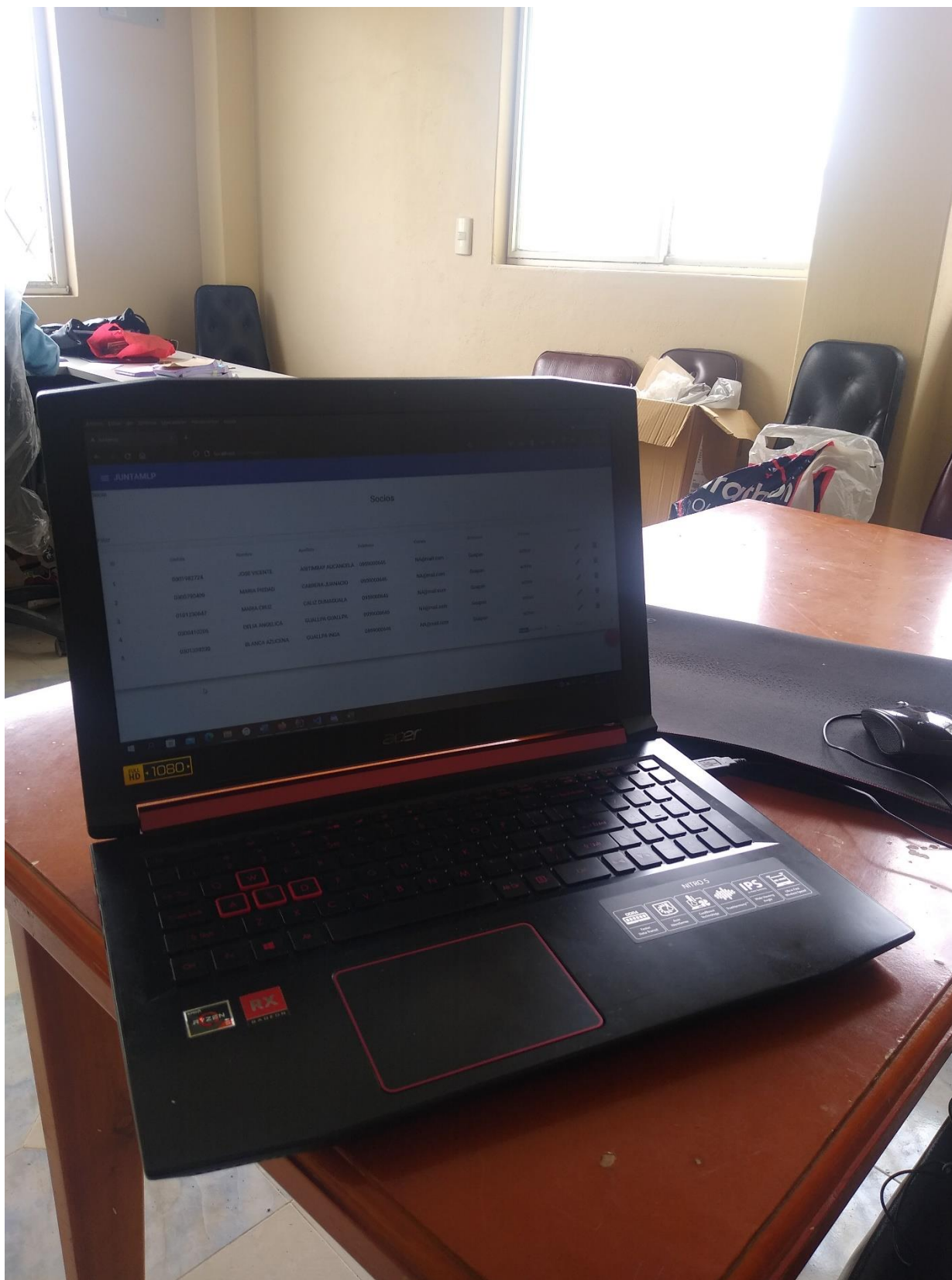
Una vez finalizado y entregado el sistema de forma definitiva no se le dará mantenimiento al sistema, ya que al ser un proyecto piloto este a la vez de ser entregado no se le podrá hacer modificaciones.

Firma



Kevin Raúl Guaman Arenillas
Ced: 0302366281

Anexo A. Levantamiento de Requerimientos



Anexo B. Pruebas del Prototipo 1



Anexo C. Pruebas del Prototipo 2




Anexo D. Pruebas del Prototipo 3

Guapán, 03 de Abril del 2023

Carta de Aceptación

Yo Lucia Dolores Ortega Gualpa con número de cédula 0301555371, tesorera y encargada del cobro de consumo mensual de la Junta de Agua Potable "Monseñor Leónidas Proaño" de la comunidad de Guartanchun. Manifiesto lo siguiente: que el estudiante Kevin Raúl Guamán Arenillas con numero de cédula 0302366281, realizó la demostración del sistema de facturación que esta desarrollando, el mismo que servirá en el futuro para implantar dicho sistema y mejorar nuestra forma de facturación de los usuarios de nuestra institución, en virtud de la cual procedo a realizar la **ACEPTACIÓN** del mencionado sistema de facturación, con el conocimiento y autorización del señor Cristóbal Ochoa presidente de la Junta de Agua Potable.

Es todo cuanto puedo informar en honor a la verdad para que el mencionado estudiante pueda hacer el uso necesario de dicha carta de aceptación.



Lucia Dolores Ortega Gualpa
TESORERA DE LA JUNTA

Anexo E. Carta de Aceptación del Prototipo



Universidad
Católica
de Cuenca

**AUTORIZACIÓN DE PUBLICACIÓN EN EL
REPOSITORIO INSTITUCIONAL**

CÓDIGO: F – DB – 30
VERSION: 01
FECHA: 2021-04-15
Página 1 de 1

Kevin Raúl Guamán Arenillas portador de la cédula de ciudadanía N° **0302366281**. En calidad de autor y titular de los derechos patrimoniales del trabajo de titulación **"Propuesta y desarrollo de un prototipo para el registro de pagos en la Junta de Agua Monseñor "Leónidas Proaño"** de conformidad a lo establecido en el artículo 114 Código Orgánico de la Economía Social de los Conocimientos, Creatividad e Innovación, reconozco a favor de la Universidad Católica de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos y no comerciales. Autorizo además a la Universidad Católica de Cuenca, para que realice la publicación de éste trabajo de titulación en el Repositorio Institucional de conformidad a lo dispuesto en el artículo 144 de la Ley Orgánica de Educación Superior.

Azogues, 21 de marzo de 2023

F:

Kevin Raúl Guamán Arenillas

C.I. 0302366281