



UNIVERSIDAD TÉCNICA DE AMBATO
FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E
INDUSTRIAL
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES E
INFORMÁTICOS

Tema:

DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE
FACTURACIÓN, TOMA DE LECTURA Y CONSULTA DE PLANILLAS, EN LA
JUNTA ADMINISTRADORA DE AGUA POTABLE MOLLEPAMBA DE LA
PARROQUIA PICAIHUA

Trabajo de Titulación Modalidad: Proyecto de Investigación, presentado previo la obtención del título de Ingeniero en Sistemas Computacionales e Informáticos

ÁREA: Software

LÍNEA DE INVESTIGACIÓN: Desarrollo de Software

AUTOR: Walter Danilo Semblantes Quinche.

TUTOR: Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

Ambato - Ecuador

agosto - 2021

APROBACIÓN DEL TUTOR

En calidad de tutor del Trabajo de Titulación con el tema: DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE FACTURACIÓN, TOMA DE LECTURA Y CONSULTA DE PLANILLAS, EN LA JUNTA ADMINISTRADORA DE AGUA POTABLE MOLLEPAMBA DE LA PARROQUIA PICAIHUA, desarrollado bajo la modalidad Proyecto de Investigación por el señor Walter Danilo Semblantes Quinche, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, me permito indicar que el estudiante ha sido tutorado durante todo el desarrollo del trabajo hasta su conclusión, de acuerdo a lo dispuesto en el Artículo 15 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y el numeral 7.4 del respectivo instructivo.

Ambato, agosto 2021.

Ing. Franklin Oswaldo Mayorga Mayorga, Mg.

TUTOR

AUTORÍA

El presente Proyecto de Investigación titulado: DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE FACTURACIÓN, TOMA DE LECTURA Y CONSULTA DE PLANILLAS, EN LA JUNTA ADMINISTRADORA DE AGUA POTABLE MOLLEPAMBA DE LA PARROQUIA PICAIHUA es absolutamente original, auténtico y personal. En tal virtud, el contenido, efectos legales y académicos que se desprenden del mismo son de exclusiva responsabilidad del autor.

Ambato, agosto 2021



Walter Danilo Semblantes Quinche

CI: 1804494142

AUTOR

APROBACIÓN TRIBUNAL DE GRADO

En calidad de par calificador del Informe Final del Trabajo de Titulación presentado por el señor Walter Danilo Semblantes Quinche, estudiante de la Carrera de Ingeniería en Sistemas Computacionales e Informáticos, de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial, de la Universidad Técnica de Ambato, bajo la Modalidad Proyecto de Investigación , titulado DESARROLLO DE UN SISTEMA WEB PARA EL PROCESO DE FACTURACIÓN, TOMA DE LECTURA Y CONSULTA DE PLANILLAS, EN LA JUNTA ADMINISTRADORA DE AGUA POTABLE MOLLEPAMBA DE LA PARROQUIA PICAIHUA , nos permitimos informar que el trabajo ha sido revisado y calificado de acuerdo al Artículo 17 del Reglamento para obtener el Título de Tercer Nivel, de Grado de la Universidad Técnica de Ambato, y al numeral 7.6 del respectivo instructivo. Para cuya constancia suscribimos, conjuntamente con la señora Presidenta del Tribunal.

Ambato, agosto 2021

Ing. Elsa Pilar Urrutia Mg.

PRESIDENTA DEL TRIBUNAL

Ing. Hernán Naranjo
PROFESOR CALIFICADOR


PhD. Julio Balarezo
PROFESOR CALIFICADOR

DERECHOS DE AUTOR

Autorizo a la Universidad Técnica de Ambato, para que haga uso de este Trabajo de Titulación como un documento disponible para la lectura, consulta y procesos de investigación.

Cedo los derechos de mi Trabajo de Titulación en favor de la Universidad Técnica de Ambato, con fines de difusión pública. Además, autorizo su reproducción total o parcial dentro de las regulaciones de la institución.

Ambato, agosto 2021



Walter Danilo Semblantes Quinche

CC: 1804494142

AUTOR

DEDICATORIA

Dedico este trabajo de investigación a mi familia, compañeros, amigos y a a todas las personas que de una u otra forma fueron un apoyo durante mis estudios.

Walter Semblantes

AGRADECIMIENTO

Agradezco a Dios, a mi familia, por el apoyo incondicional que me brindan cada día, a la Universidad por la oportunidad de superación al formarme como un profesional crítico y competente, y a mis maestros ya que gran parte del conocimiento que hoy poseo los he adquirido gracias a ellos.

Walter Semblantes

ÍNDICE

APROBACIÓN DEL TUTOR	ii	
AUTORÍA	iii	
APROBACIÓN COMISIÓN CALIFICADORA	iv	
DERECHOS DE AUTOR	v	
Dedicatoria	vi	
Agradecimiento	vii	
Introducción	xx	
CAPÍTULO I	MARCO TEÓRICO	1
1.1 Tema de Investigación		1
1.2 Antecedentes Investigativos		1
1.2.1 Contextualización del problema		1
1.2.2 Fundamentación teórica		2
1.2.2.1 Informática		2
1.2.2.2 Sistemas Informáticos		2
1.2.2.3 Sistemas web		2
1.2.2.4 Metodología ágil		3
1.2.2.5 Framework (Marco de desarrollo)		4
1.2.2.6 Patrón MVC		5
1.3 Objetivos		6
1.3.1 General		6
1.3.2 Específicos		6
CAPÍTULO II	METODOLOGÍA	7
2.1 Materiales		7
2.1.1 Ficha de observación		7

2.1.2	Formato de entrevista	7
2.2	Métodos	8
2.2.1	Modalidad Básica de la investigación	8
2.2.2	Población y muestra	8
2.2.3	Recolección de información	9
2.2.4	Procesamiento y análisis de datos	9
2.2.4.1	Observación	9
2.2.4.2	Entrevista	11
2.2.4.3	Conclusiones o interpretación de la entrevista y la observación	13
CAPÍTULO III		
	RESULTADOS Y DISCUSIÓN	15
3.1	Análisis y discusión de resultados	15
3.1.1	Proceso de recaudación de la junta de agua	15
3.1.2	Metodología de desarrollo de software	16
3.1.2.1	Metodologías ágiles para el desarrollo de un sistema web eficiente	17
3.1.2.2	Metodología ágil OpenUp	19
3.1.3	Elección del framework para el desarrollo web	21
3.1.3.1	Framework marco .NET	22
3.1.4	Aplicación de la metodología OpenUP en el desarrollo del proyecto	24
3.1.4.1	Plan de Desarrollo de Software	24
3.1.4.2	Resumen Plan del Proyecto OpenUP	25
3.1.4.3	Productos del proyecto	25
3.2	Especificación de requisitos de Software	27
3.2.1	Introducción	27
3.2.1.1	Propósito	27
3.2.1.2	Alcance	27
3.2.1.3	Definiciones, Acrónimos y Abreviaciones	27
3.2.1.4	Características de los usuarios	27
3.2.1.5	Restricciones	28
3.2.1.6	Suposiciones y Dependencias	29
3.2.2	Requisitos Funcionales	29
3.2.2.1	Módulo: Categorías	29
3.2.2.2	Módulo: Rubros	30
3.2.2.3	Módulo: Configuración	30
3.2.2.4	Módulo: Usuarios	30

3.2.2.5	Módulo: Clientes	31
3.2.2.6	Módulo: Lecturas	32
3.2.2.7	Módulo: Cuentas por cobrar	32
3.2.2.8	Módulo: Consultas planilla	33
3.2.2.9	Módulo: Facturación	33
3.2.2.10	Módulo: Reportes	34
3.2.3	Requisitos no funcionales	34
3.2.3.1	Requisitos de Rendimiento	34
3.2.3.2	Seguridad	35
3.3	Documento de arquitectura de software	36
3.3.1	Introducción	36
3.3.1.1	Propósito	36
3.3.1.2	Alcance	36
3.3.1.3	Organización del documento	36
3.3.2	Representación de la arquitectura	36
3.3.3	Objetivos y restricciones de la arquitectura	37
3.3.4	Vista de Casos de Uso	37
3.3.4.1	Introducción	37
3.3.4.2	Identificación de los Casos de Uso relevantes para la arquitectura	37
3.3.4.3	Descripción de los casos de uso relevantes para la arquitectura	45
3.3.5	Vista Lógica	45
3.3.5.1	Introducción	45
3.3.5.2	Descomposición en Subsistemas	46
3.3.5.3	Descripción de los Subsistemas	46
3.3.5.4	Diseño de Subsistemas	46
3.3.5.5	Diagrama de Clases	51
3.3.5.6	Diagrama de Entidad/Relación	52
3.3.6	Vista de Despliegue	52
3.3.6.1	Introducción	52
3.3.6.2	Distribución y Despliegue	53
3.3.7	Vista de Datos	53
3.4	Plan de Pruebas de Software	54
3.4.1	Introducción	54
3.4.1.1	Objetivos	54
3.4.1.2	Alcance	54

3.4.2	Instrumentos de Prueba	54
3.4.2.1	Módulos del Programa	54
3.4.3	Características a ser probadas	55
3.4.3.1	Pruebas Funcionales	56
3.4.3.2	Pruebas de Comportamiento	56
3.4.4	Proceso de pruebas	56
3.4.5	Desarrollo de las interfaces	65
3.4.6	Desarrollo de las interfaces utilizando Entity framework	70
3.4.6.1	Pantalla principal	74
3.5	Implantación del software	78
3.5.1	Introducción	78
3.5.2	Resumen	78
3.5.2.1	Requisitos de Hardware	78
3.5.2.2	Dominio y NET PREMIUM Seguridad	80
3.5.2.3	Análisis de la Aplicación Web.	84
3.5.2.4	SEO a la página web	85
3.5.2.5	Configuraciones y costos de servicios	85
3.5.2.6	Costos de la instalación	87
3.6	Aceptabilidad del software	87

CAPÍTULO IV	CONCLUSIONES Y RECOMENDACIONES	88
4.1	Conclusiones	88
4.2	Recomendaciones	88
Bibliografía		90
ANEXOS		91

ÍNDICE DE TABLAS

1	Manifiesto ágil.	4
2	Formato ficha de observación.	7
3	Formato entrevista.	8
4	Personal Administrativo J.A.A.P.M	8
5	Ficha de Observación Toma de Lecturas	10
6	Ficha de Observación Toma de Lecturas	10
7	Matriz de la entrevista presidente de la junta.	11
8	Matriz de la entrevista secretario de la junta.	12
9	Matriz de la entrevista lector de la junta.	13
10	Diferencias entre metodologías ágiles y no ágiles.	16
11	Ejemplos de metodologías.	16
12	Requerimientos metodología ágil.	17
13	Comparación entre Metodologías.	18
14	Requerimientos técnicos framework.	21
15	Tabla .NET vs Spring vs Laravel.	22
16	Productos del proyecto.	25
17	Desarrollador del proyecto de investigación.	26
18	Tutor del proyecto de investigación.	26
19	Propietario del software.	26
20	Plan de Fases.	26
21	Características de los usuarios.	28
22	Requisito funcional: CRUD categorías.	29
23	Requisito funcional: CRUD tarifas.	29
24	Requisito funcional: CRUD rubros.	30
25	Requisito funcional: configuración.	30
26	Requisito funcional: Ingreso al sistema.	30
27	Requisito funcional: Ingreso al sistema (restricción).	31
28	Requisito funcional: CRUD usuarios.	31
29	Requisito funcional: CRUD clientes.	31

30	Requisito funcional: medidores.	32
31	Requisito funcional: lecturas.	32
32	Requisito funcional: cuentas por cobrar.	33
33	Requisito funcional: Consulta de planilla.	33
34	Requisito funcional: Facturación.	34
35	Requisito funcional: Reportes.	34
36	Descripción de los Módulos del Programa	55
37	Características a ser Probadas	55
38	Prueba de Integración	56
39	Pruebas de Comportamiento	56
40	Prueba CRUD Tarifas	57
41	Prueba CRUD RUBROS	58
42	Prueba Usuarios	59
43	Prueba Clientes	60
44	Prueba Medidores	61
45	Prueba Lecturas	62
46	Prueba Cuentas por Cobrar	63
47	Prueba Cuentas por Cobrar	64
48	Prueba Facturación	64
49	Prueba Reportes	65
50	PLAN SELECCIONADO DEL HOSTING	80
51	Características del Alojamiento	80
52	Características de Hospedaje de windows server.	81
53	MySQL y SQL Server	81
54	Características de la lista de correo electrónico.	82
55	Características del Scripting	82
56	Componentes Instalados	83
57	Hardware	83
58	Seguridad	83
59	Costos de servicios anuales instalados.	87

ÍNDICE DE FIGURAS

1	Patrón MVC.	5
2	Proceso junta de agua de Mollepamba.	15
3	Elementos de OpenUP.	19
4	Ciclo de vida en OpenUP.	20
5	Practicas de OpenUP.	21
6	Microsoft blazor.	23
7	Resumen del Plan del Proyecto.	25
8	Características de los Usuarios	38
9	Requisito Funcional: CRUDtarifas	38
10	Requisito Funcional CRUD rubros	39
11	Requisitos Funcional Configuración	39
12	Requisito Funcional Ingreso al Sistema	40
13	Requisito funcional: Ingreso al sistema (restricción)	40
14	Requisito Funcional CRUD usuarios	41
15	Requisito funcional CRUD clientes	41
16	Requisito Funcional:Medidores	42
17	Requisito Funcional:Lecturas	42
18	Requisito Funcional Cuentas por Cobrar	43
19	Requisito Funcional: Consulta de planilla	43
20	Listar deudas del cliente	44
21	facturar consumo de agua potable	44
22	Reporte Facturación y administración	45
23	Descomposición en Subsistemas	46
24	Consulta de Planillas	46
25	Sistema de Facturación	47
26	Módulo Consulta de Planillas	47
27	Módulo Facturación	48
28	Módulo Reportes	48
29	Módulo Rubros	49
30	Módulo Configuración	49

31	Módulo Usuarios	50
32	Módulo Clientes	50
33	Módulo Lecturas	51
34	Diagrama de Clases	51
35	Diagrama de Entidad/Relación.	52
36	Distribución y Despliegue	53
37	Boceto Pantalla Principal	66
38	Boceto Conexión Usuario-Contraseña	66
39	Boceto Menú lateral	67
40	Boceto Módulo Usuarios	67
41	Boceto:Módulo Tarifas	68
42	Boceto:Módulo Configuraciones	68
43	Boceto:Módulo Rubros	68
44	Boceto:Módulo Clientes	69
45	Boceto: Módulo Cuentas por Cobrar	69
46	Boceto:Módulo Facturar	69
47	Boceto:Módulo Reportes	70
48	Boceto:Módulo Lecturas	70
49	EF y MVC.	71
50	EF modelo entidad.	72
51	EF vista entidad insertar o actualizar registro.	73
52	EF controlador función insertar.	73
53	Interfaz:Pantalla Principal	74
54	Interfaz Conexión-Usuario/Contraseña	75
55	Interfaz Usuario	75
56	Interfaz Tarifas	76
57	Interfaz Configuraciones	76
58	Interfaz Rubros	76
59	Interfaz Clientes	77
60	Interfaz Cuentas por Cobrar	77
61	Interfaz facturar.	78
62	Ecuahosting	79
63	smarterasp.net	79
64	Aplicación Web Análisis 1	84
65	Análisis 2 Rendimiento de la página web	84
66	Primer lugar en la Búsqueda del Sitio Web-Google.	85
67	Compra y Configuración del Dominio Mollepamba en el Hosting	86

68	Configuración del Certificado SSL para Protección de los Datos. . . .	86
69	Configuración del Cloudflare-Servicio de Seguridad en el Hosting. . .	86

RESUMEN EJECUTIVO

El tema del proyecto se centra en desarrollar un sistema web para el proceso de facturación, toma de lectura y consulta de planillas, en la junta administradora de agua potable “Mollepamba” de la parroquia Picaihua , permitiendo manejar de manera adecuada los datos obtenidos para el respectivo rubro a cobrarse.

Entre los pilares principales de este proyecto se encuentra buscar que la plataforma desarrollada esté enfocada a un fácil acceso y comprensión por parte de las personas que vayan a manejarlo la pantalla principal permitirá a los usuarios ingresar sus datos y consultar los valores a cancelarse sean estos por medio de sus datos personales o el número de medidor especificado. Por otro lado, se espera conseguir que dicha plataforma sea segura y esté a la par de las exigencias de los usuarios de la junta de Agua de Mollepamba, se debe agregar que este proyecto ha sido fruto del compendio de conocimientos que se han venido adquiriendo durante la vida universitaria y que han permitido poner en marcha un software efectivo seguro y accesible.

En cuanto al modelo de funcionamiento del sistema, este se fundamenta en pasos sistemáticos empezando con la elaboración de Plan de desarrollo de Software en donde se detallarán las especificaciones y los requisitos del Software, seguidamente se elaborará un documento donde se indique la arquitectura del Software, elaboración y puesta en marcha del sistema, así como posibles fallos y sus soluciones. Finalmente con la definición del Software se genera un primer prototipo el cual será evaluado bajo una tabulación de prueba y error que posteriormente será refinado y corrigiendo sus posibles fallos y dando como resultado un segundo prototipo funcional que derive en su Implantación.

Palabras clave: Junta de agua, sistema web, OPENUP, .net, MVC, entity framework.

ABSTRACT

The subject of the project is focused on developing a web system for the billing process, reading and consulting the spreadsheets, in the Picaihua parish drinking water administration board "Mollepamba", allowing the data to be handled properly obtained for the respective item to be collected. Among the main pillars of this project is to find that the developed platform is focused on easy access and compression by the people who are going to handle it.

The main screen will allow users to enter their data and consult the values to be canceled are these by means of your personal data or the specified meter number. On the other hand, it is expected that said platform is safe and meets the demands of the users of the Mollepamba Water Board, it should be added that this project has been the result of the compendium of knowledge that has been acquired during university life and that have made it possible to implement safe and accessible effective software.

Regarding the operating model of the system, it is based on systematic steps starting with the development of the Software Development Plan where the specifications and requirements of the Software will be detailed, then a document will be prepared indicating the architecture of the Software, development and commissioning of the system, as well as possible failures and their solutions. Finally, with the definition of the Software, a first prototype is generated which will be evaluated under a trial and error tabulation that will later be refined, correcting its possible failures and resulting in a second functional prototype that results in its Implementation.

keywords: Water board, web system, OPENUP, .net, MVC, entity framework.

INTRODUCCIÓN

El tema del proyecto se centra en desarrollar un sistema web para el proceso de facturación, toma de lectura y consulta de planillas, en la junta administradora de agua potable “Mollepamba” de la parroquia Picaihua , permitiendo manejar de manera adecuada los datos obtenidos tanto de las lecturas de los medidores, así como el respectivo rubro a cobrarse. Entre los ejes principales de este proyecto están que la plataforma desarrollada sea de fácil acceso y comprensión para las personas que vayan a manejarlo y a su vez de la misma manera ser intuitivo para los usuarios ya que podrán entrar a la página web consultar de forma clara y sencilla los valores a pagarse mediante el tipeo de sus datos personales o el número de medidor correspondiente. En cuanto al modelo de funcionamiento de nuestro sistema, este se fundamenta en la elaboración de Plan de desarrollo de Software en donde se detallarán las especificaciones y los requisitos del Software, seguido a este paso se realizará un documento de arquitectura del Software en el que se detallaran todos los pasos para la elaboración y puesta en marcha del sistema, así como posibles fallos y sus soluciones.

Terminado este proceso se empezará con la definición de pruebas para el Software mediante el desarrollo del primer prototipo, mediante tabulación de prueba y error se generará un primer prototipo funcional el cual será puesto en Refinamiento de los requerimientos funcionales generando un segundo prototipo funcional para finalmente derivar en su Implantación.

Actualmente el mundo se encuentra cada vez más globalizado, esto se ve reflejado en los grandes avances tecnológicos los cuales están en están enfocados en mejorar y facilitar los procesos que anteriormente se lo realizaban de forma manual lo que acarrea posibles errores humanos y que demanden una cantidad de tiempo mayor . En el presente la conectividad a los servicios digitales cada año está ganando espacio y se ha vuelto una política no solo de las grandes ciudades si no por el contrario cada año está abarcando a las zonas rurales haciendo partícipes a sus habitantes de los beneficios de implementar la tecnología y que las personas se involucren más a fondo con estos sistemas informáticos para la agilización de procesos.

Bajo esta premisa se busca el desarrollo e implementación de una plataforma cuyo principal objetivo está enfocado en subsanar los problemas de carácter logístico y técnico que suponía el cobro del valor correspondiente de agua en función a los datos obtenidos de consumo arrojados por el medidor. La gran mayoría de estos problemas se

basan en un mal manejo de los datos, rubros mal calculados, errónea identificación de los datos del usuario entre otros lo que favorece a que el proceso sea tardío e ineficiente. La SWP abreviatura para sistema web que mejore el proceso de facturación, toma de lectura y consulta de planillas, de la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua.

El SWP se implantará en un servidor web con soporte para aplicaciones web desarrolladas en la plataforma .Net, manejando su base de datos en SQL Server 2017 Express de igual manera esta plataforma se soportará en cualquier navegador con las características de proporciona la plataforma .NET, estos son: Internet Explorer 9+, Microsoft Edge, Safari, Firefox, Google Chrome, Opera. Es importante tomar en cuenta las limitaciones que tiene SQL Server Express especialmente en el tamaño máximo que nos proporciona para la base de datos que son 10GB para poder realizar una análisis de comparación con los datos que almacenamos. Finalmente, el sistema se valorará según la aceptación que tenga con los miembros de la Junta de agua de Mollepamba quienes serán los principales actores y beneficiarios en el uso de esta plataforma la cual contará con un fácil acceso y comprensión para los usuarios evitando así el tedioso trabajo de facturar de forma manual y muchas veces con errores.

CAPÍTULO I

MARCO TEÓRICO

1.1. Tema de Investigación

Sistema web para el proceso de facturación, toma de lectura y consulta de planillas, en la junta administradora de agua potable “Mollepamba” de la parroquia Picaihua.

1.2. Antecedentes Investigativos

1.2.1. Contextualización del problema

Carolina Elizabeth Ruiz Chicaiza Y Silvia Janeth Paguay Sánchez de la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO en su trabajo de tesis “SISTEMA DE COBRO DE AGUA POTABLE DE LA COMUNIDAD EL SOCORRO UTILIZANDO EL FRAMEWORK ASP.NET MVC”,2017(Pág. 64) concluye que el método automatizado para realizar la facturación permitió una mejora del 95,67 % que el método tradicional-manual, por lo que el sistema logro disminuir de manera significativa inconvenientes presentados en la junta de agua potable [1].

Erika Magdalena Llerena Ortiz, de la UNIVERSIDAD TÉCNICA DE AMBATO en su trabajo de investigación titulado “SISTEMA DE FACTURACIÓN PARA EL CONTROL AUTOMATIZADO DE LAS TARIFAS RECAUDADAS EN LAS JUNTAS ADMINISTRADORAS DE AGUA POTABLE ADSCRITAS AL PARLAMENTO AGUA DEL GOBIERNO PROVINCIAL DE TUNGURAHUA”,2011(Pág. 143) manifiesta que el contar con un modelo de facturación ayuda a manejar la información de manera rápida y eficiente, lo que permite brindar mejor atención a los contribuyentes[2].

En la tesis “Sistema Web de Gestión de Procesos para una Junta de Agua Potable utilizando las tecnologías de software libre, JSF.” 2014 (Pag.25), de la UNIVERSIDAD TÉCNICA DEL NORTE, desarrollada por Franklin Andrés Cheza Luna, señala que la automatización de los procesos es una ventaja necesaria y prioritaria dentro de las empresas, debido a los beneficios que estos provocan[3].

El siguiente trabajo realizado por KERLY PATRICIA CHABLA VINTIMILLA de la UNIVERSIDAD TECNOLÓGICA INDOAMÉRICA, septiembre 2017 (Pg. 84) Nos dice que un sistema web de facturación puede brindar reportes indispensables como; deudas por cobrar, cobros mensuales, multas, descuentos, etc., y que contar con esta información es indispensable en una empresa para la toma de decisiones Teniendo en cuenta estos antecedentes podemos decir que el desarrollo del proyecto mejorará de manera significativa el proceso de facturación que se viene realizando de forma manual y con paquetes de ofimática muy básicos [4].

1.2.2. Fundamentación teórica

1.2.2.1. Informática

Ciencia que se dedica a estudiar el tratamiento de la información mediante medios automáticos, es decir, la ciencia de la información automática [5]. Se trata de una unión de conocimientos científicos y de técnicas que posibilitan el tratamiento automático de la información mediante el uso de computadoras.

1.2.2.2. Sistemas Informáticos

Es el conjunto de partes interrelacionadas, una computadora que usa dispositivos programables para capturar, almacenar y procesar datos. La computadora personal junto con la persona que lo maneja y los periféricos que los envuelven, resultan de por sí un ejemplo de un sistema informático. Incluso la computadora más sencilla se clasifica como un sistema informático, porque al menos dos componentes (hardware y software) tienen que trabajar unidos. Pero el genuino significado de "sistema informático" viene mediante la interconexión. Muchos sistemas informáticos pueden interconectarse, esto es, unirse para convertirse un sistema mayor. La interconexión de sistemas informáticos puede tornarse difícil debido a incompatibilidades[6]. A veces estas dificultades ocurren a nivel de hardware, mientras que en otras ocasiones se dan entre programas informáticos que no son compatibles entre sí.

1.2.2.3. Sistemas web

Se denomina sistema web a aquellas aplicaciones de software que puede utilizarse accediendo a un servidor web a través de Internet o de una intranet mediante un navegador [6]. Las aplicaciones web son muy usadas hoy en día, debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo.

1.2.2.4. Metodología ágil

Ágil es el conjunto de buenos valores y buenas prácticas para el desarrollo de proyectos. Todos los métodos ágiles están fundamentados en el manifiesto ágil, que es el resultado del trabajo de un grupo de expertos, siendo estos los mismos creadores de las metodologías ágiles. Teniendo como objetivo acordar y definir valores que ayuden a los equipos de trabajos en la gestión de sus proyectos de manera eficiente, rápida y con adaptación a los cambios[7]. Estas metodologías son flexibles y pueden ser modificadas para que se ajusten a la realidad de cada proyecto en el que se esté aplicando. Los proyectos ágiles se subdividen en más pequeños mediante una lista ordenada de características y cada uno es tratado de manera independiente desarrollando características durante un tiempo determinado, además durante el desarrollo la comunicación con el cliente es constante[8].

Como ejemplos de metodologías ágiles tenemos SCRUM, OPENUP, XP (Programación Extrema). Estas están basadas en el manifiesto ágil que establece un conjunto de políticas que diferencia una metodología ágil de una tradicional. En la siguiente tabla se establecen los 12 principios del manifiesto ágil.

Principios de manifiesto ágil
Nuestra principal prioridad es satisfacer al cliente a través de la entrega temprana y continua de software con valor.
Aceptamos que los requisitos cambien, incluso en etapas tardías del desarrollo. Los procesos ágiles aprovechan el cambio para proporcionar ventaja competitiva al cliente.
Entregamos software funcional frecuentemente, entre dos semanas y dos meses, con preferencia al período de tiempo más corto posible.
Los responsables del negocio y los desarrolladores trabajamos juntos de forma cotidiana durante todo el proyecto.
Los proyectos se desarrollan en torno a individuos motivados. Hay que darles el entorno y el apoyo que necesitan, y confiarles la ejecución del trabajo.
El método más eficiente y efectivo de comunicar información al equipo de desarrollo y entre sus miembros es la conversación cara a cara.
El software funcionando es la medida principal de progreso.
Los procesos ágiles promueven el desarrollo sostenido. Los promotores, desarrolladores y usuarios debemos mantener un ritmo constante de forma indefinida.
La atención continua a la excelencia técnica y al buen diseño mejora la agilidad.
La simplicidad, o el arte de maximizar la cantidad de trabajo no realizado, es esencial.
A intervalos regulares, el equipo reflexiona sobre cómo ser más efectivo para, a continuación, ajustar y perfeccionar su comportamiento en consecuencia.

Tabla 1: Manifiesto ágil.

Elaborado por: [7].

1.2.2.5. Framework (Marco de desarrollo)

El concepto framework se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones Web. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito. En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta [9].

Los objetivos principales que persigue un framework son: acelerar el proceso de desarrollo, reutilizar código ya existente y promover buenas prácticas de desarrollo como el uso de patrones. Un framework Web, por tanto, podemos definirlo como un conjunto de componentes (por ejemplo clases en C# y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de aplicaciones web.

1.2.2.6. Patrón MVC

El patrón Modelo-Vista-Controlador es una guía para el diseño de arquitecturas de aplicaciones que ofrezcan una fuerte interactividad con usuarios. Este patrón organiza la aplicación en tres modelos separados, el primero es un modelo que representa los datos de la aplicación y sus reglas de negocio, el segundo es un conjunto de vistas que representa los formularios de entrada y salida de información, el tercero es un conjunto de controladores que procesa las peticiones de los usuarios y controla el flujo de ejecución del sistema[9].

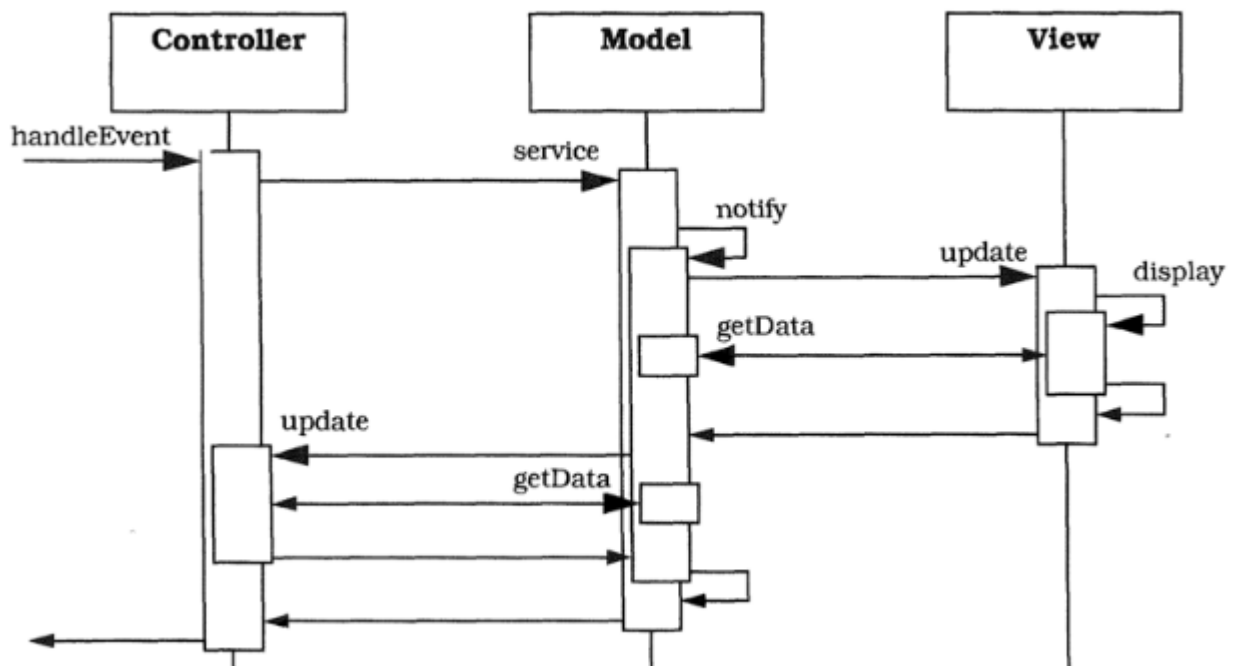


Figura 1: Patrón MVC.
Elaborado por: [9].

1.3. Objetivos

1.3.1. General

Desarrollar un sistema web, para mejorar el proceso de facturación, toma de lectura y consulta de planillas, en la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua.

1.3.2. Específicos

- Determinar los procesos para la recaudación de tarifas por suministro de Agua Potable.
- Definir la metodología de desarrollo ágil adecuada que facilite la implementación del proyecto planteado
- Comparar los principales Frameworks para desarrollar un sistema web eficiente.
- Implementar un sistema web que mejore el proceso de facturación, toma de lectura y consulta de planillas, de la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua.

CAPÍTULO II

METODOLOGÍA

2.1. Materiales

Las técnicas e instrumentos utilizados para la recolección información fueron la ficha de observación y entrevista.

La observación nos ayuda a tomar información para luego analizarla y procesarla, por su parte la entrevista nos permite tener un dialogo con las personas que tienen conocimiento sobre el tema que estamos investigando, lo cual nos ayuda a tener en cuenta las principales características para resolver el problema.

2.1.1. Ficha de observación

Se establece el siguiente formato a aplicar en la Junta Administradora de Agua Potable Mollepamba.

Ficha de observación	
Número	
Lugar	
Fecha	
Elaborado por	
Categoría	Lo observado
categoría 1	detalle observación categoría 1
...
...

Tabla 2: Formato ficha de observación.

Elaborado por: El investigador.

2.1.2. Formato de entrevista

Se establece el siguiente formato a aplicar con las personas que dirigen de la Junta Administradora de Agua Potable Mollepamba.

Formato entrevista	
Lugar	
Fecha	
Entrevistados	
Pregunta	Respuesta
...	...
...	...

Tabla 3: Formato entrevista.

Elaborado por: El investigador.

2.2. Métodos

2.2.1. Modalidad Básica de la investigación

El presente trabajo tiene las siguientes modalidades de investigación:

Modalidad Bibliográfica o Documentada Se considera esta modalidad ya que se recurre a diferentes fuentes obtenidas de libros, artículos científicos, tesis desarrolladas en Universidades para profundizar enfoques con respecto al tema de la investigación.

Modalidad aplicada Por la utilización de los conocimientos adquiridos a lo largo de la carrera universitaria.

2.2.2. Población y muestra

De acuerdo al tema de investigación no se requiere realizar el cálculo de población y muestra, ya que se solo se trabajará con la parte administrativa de la Junta Administradora de Agua Potable Mollepamba. En la siguiente tabla se muestra el número de personas con las cuales se trabajará para el desarrollo de la investigación. Para la siguiente investigación no se tomará en cuenta a los usuarios finales ya que el sistema casi en su totalidad será revisado y manejado por la parte administrativa, Los usuarios finales son parte del proceso de facturación, pero no entran al manejo de datos del sistema como si lo hace la parte administrativa para ingresar datos, actualizar datos, sacar reportes etc.

Nombres	Cargo	Número	Porcentaje
Sr. Criollo Jose	Presidente	1	33.3 %
Sr. Luis Chango	Secretario	1	33.3 %
Sr. Juan Palate	Lector/Medidores	1	33.3 %
	Total	3	100 %

Tabla 4: Personal Administrativo J.A.A.P.M

Elaborado por: El investigador.

2.2.3. Recolección de información

Las técnicas para la recolección de información necesaria en el proyecto de investigación son por observación y entrevista.

Observación

Esta técnica determinará el procedimiento manual que se lleva a cabo antes durante y después del caso en estudio, mediante esta técnica se podrá encontrar y corregir las principales falencias en la parte administrativa. Dentro de la observación se fijan detalles que serán tomadas en cuenta para establecer prioridades en las historias de usuario que serán documentadas dentro del análisis del proyecto.

Entrevista

Esta técnica proporcionará una forma objetiva de conseguir requerimientos. Mediante esta técnica podemos obtener información directa de la parte administrativa ya que la entrevista se realiza con las personas que poseen más conocimiento sobre los procesos de facturación mensual.

2.2.4. Procesamiento y análisis de datos

Una vez realizada la entrevistas a los miembros de la Junta Administradora de Agua Potable Mollepamba, junto con la información recolectada mediante la observación, podemos concluir lo siguiente.

2.2.4.1. Observación

La observación se realizó en la oficina de la Junta Administradora de Agua Potable Mollepamba, en la cual se obtuvo información acerca del proceso de facturación que se realiza actualmente, cuya información se encuentra en la siguiente ficha de la tabla 2.4 y 2.5

Ficha de Observación	
Número	1
Lugar	Oficina Junta Administradora de Agua Potable Mollepampa
Fecha	05/08/2020
Elaborado por	Walter Semblantes
Categoría	Lo observado
Registro de Usuario	Se registra los datos básicos en formato Excel, y a mano.
Registro Medidor	Se asigna manualmente.
Facturación	Revisión manual de facturas pendientes
	No disponen de un sistema automatizado.
	Los cálculos para determinar los cobros no son precisos.
Reportes	No disponen de reportes automatizados, se lo realiza manualmente
	El tiempo de respuesta cuando un usuario requiere información no es oportuna.

Tabla 5: Ficha de Observación Toma de Lecturas

Elaborado por: El investigador.

Análisis e interpretación de resultados: Se puede evidenciar que la Junta Administradora de Agua no dispone de un sistema web eficiente que permita visualizar el proceso de cobros de planillas.

Ficha de Observación	
Número	2
Lugar	Oficina Junta Administradora de Agua Potable Mollepampa
Fecha	05/08/2020
Elaborado por	Walter Semblantes
Categoría	Lo observado
Registro de Lecturas	La lectura de medidores se lo realiza de manera manual
	Se registra en hojas/fichas
	La información que se recaba de la lectura de medidores se entrega a la persona responsable de los cobros.
	Ficha de ingreso de lecturas ordenado por ruta.

Tabla 6: Ficha de Observación Toma de Lecturas

Elaborado por: El investigador.

Análisis e interpretación de resultados: La información relacionada con los usuarios se

almacenan en archivos físicos, la misma que es archivada de manera diaria, haciéndose notorio la carencia de un sistema web eficiente que permita a los encargados administrar los recursos de la junta

2.2.4.2. Entrevista

La entrevista efectuada para saber la viabilidad del desarrollo del sistema que es el tema principal de este proyecto evidencia los problemas que presenta la Junta Administradora de agua potable de Mollepamba.

Entrevista personal Administrativo Junta Administradora de Agua Potable Mollepamba	
Lugar	Junta Administradora de Agua Potable Mollepamba
Fecha	05/08/2020
Entrevistado	Sr. Jose Criollo
	Respuesta
¿La Junta Administradora de Agua cuenta con un sistema informático para registrar el cobro de su planilla?	No se cuenta con ningún sistema informático, desde que se inició la vida jurídica de la Junta Administradora todas las actividades se lo ha realizado de manera manual
¿De qué manera se conserva o se mantiene el archivo con la información de los usuarios?	La información se almacenan en los archivos físicos, tarea que se lo efectúa de manera diaria con la finalidad de mantener la información disponible para cuando un usuario lo solicite.
¿Considera que el proceso que emplean actualmente para realizar el cobro de los valores por el consumo de agua a los usuarios se realiza de manera rápida y oportuna?	No se puede realizar los cobros de forma rápida puesto que todo se lo realiza de manera manual

Tabla 7: Matriz de la entrevista presidente de la junta.

Elaborado por: El investigador.

Análisis e interpretación de resultados: Se puede evidenciar que la información obtenida como producto de la lectura de los medidores se registra de manera manual y en fichas.

Entrevista personal Administrativo Junta Administradora de Agua Potable Mollepamba	
Lugar	Junta Administradora de Agua Potable Mollepamba
Fecha	05/08/2020
Entrevistado	Sr. Luis Chango (Secretario)
	Respuesta
¿ Existe riesgo de pérdida de los datos de consumo y pagos de los moradores del sector?	Hay un elevado riesgo dado que los registros se lo hacen de forma manual ciertos documentos por el paso del tiempo se están deteriorando lo que puede generar problemas a futuro, además estos datos no los tienen respaldado digitalmente.
¿ Ha existido problemas al momento de registrar el consumo correcto de Agua Potable utilizado por los moradores del Sector?	La información se almacenan en los archivos físicos, tarea que se lo efectúa de manera mensual con la finalidad de mantener la información disponible para cuando un usuario lo solicite.
¿Considera que el proceso que emplean actualmente para realizar el cobro de los valores por el consumo de agua a los usuarios se realiza de manera rápida y oportuna?	Si a existido inconvenientes sobre todo en los datos tomados por los medidores que muchas lecturas son erróneas además la personas no pueden consultar en tiempo real el valor a pagar por su consumo

Tabla 8: Matriz de la entrevista secretario de la junta.

Elaborado por: El investigador.

Análisis e interpretación de resultados: Se determina que los cobros por el consumo de agua no son eficientes puesto que todos los procesos lo realizan de manera manual.

Entrevista personal Administrativo Junta Administradora de Agua Potable Mollepamba	
Lugar	Junta Administradora de Agua Potable Mollepamba
Fecha	05/08/2020
Entrevistado	Sr. Juan Palate (Lector)
	Respuesta
¿ Cómo identifica a los medidores de los usuarios de la junta de agua potable?	Se los identifica mediante el nombre del usuario, por cada uno de los diferentes sectores.
¿ Existe algún identificador único para cada medidor escrito físicamente que le facilite la toma del consumo?	No, todo se base en el orden de la lista generada por el secretario y en ese sentido se basa en mi memoria de la persona que vive en el domicilio.
¿Cuánto tiempo le toma la recolección de los datos?	De dos a tres días, son 5 diferentes sectores de la parroquia que hay que recorrer.
¿Cuánto tiempo le toma la recolección de los datos?	Toma entre 2 a tres días.

Tabla 9: Matriz de la entrevista lector de la junta.

Elaborado por: El investigador.

Análisis e interpretación de resultados: Se evidencia la falta de organización de los datos sobre las lecturas, medidores y socios. Este con lleva a los problemas evidenciados que se darán solución con el desarrollo de un sistema web eficiente.

2.2.4.3. Conclusiones o interpretación de la entrevista y la observación

De acuerdo a la entrevista realizada con los interviniente y mediante la observación se encontraron los siguientes resultados:

- La junta de agua no cuenta con ningún sistema informático en la actualidad para

realizar el trabajo de recaudación por el servicio prestado.

- La junta de agua no lleva su información de forma digital, al contrario, la tienen de forma física.
- La junta de agua considera que la forma de cobro manual no es efectiva y rápida en la actualidad.
- La junta de agua engloba a la solución de su problema, y con la mejora de los servicios con la implantación de un sistema informático.
- La junta de agua define al ingreso de las lecturas de los medidores de agua potable como el proceso más engorroso actualmente.

CAPÍTULO III

RESULTADOS Y DISCUSIÓN

3.1. Análisis y discusión de resultados

Es preciso mencionar que la implantación de un sistema web eficiente que automatice el proceso para la emisión de pagos permitirá mejorar los servicios a sus usuarios por lo que se concluye que la implantación de un sistema de información mejorará positivamente en la calidad de los servicios y se brindará una atención eficiente a sus usuarios. En este contexto es necesaria una metodología ágil para el desarrollo del sistema web . Esta será determinada en secciones posteriores bajo un análisis que determine su adaptabilidad a este proyecto.

3.1.1. Proceso de recaudación de la junta de agua

Se establece un proceso de manejo simple y entendible para la gestión de cobros de la Junta Administradora de agua potable de Mollepamba, proceso que instituciones similares lo manejan para realizar los cobros de servicios básicos.

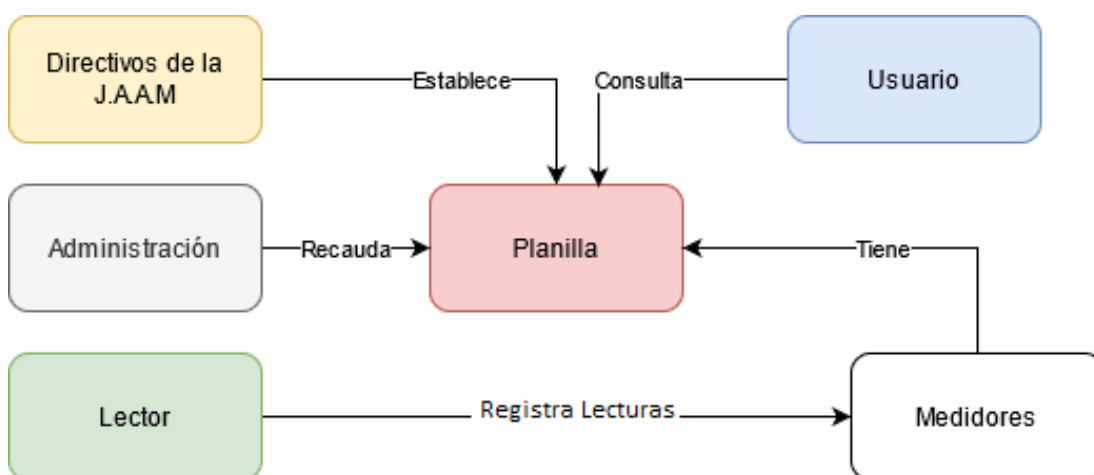


Figura 2: Proceso junta de agua de Mollepamba.
Fuente: Elaborado por: El investigador.

El proceso establecido con el que se maneja la Junta Administradora de agua potable de Mollepamba será automatizado con la implementación de un sistema web que en las siguientes secciones será desarrollado y documentado.

3.1.2. Metodología de desarrollo de software

Las principales diferencias de una Metodología ágil respecto de las Metodologías Tradicionales, en la ágil los cambios forman parte del proceso, al contrario de las tradicionales que su naturaleza es resistirse al cambio , En la Tabla 3.1, se presentan las diferencias de cada una de ellas.

Metodología Ágil	Metodología Tradicional
Pocos Artefactos. El modelado es prescindible, modelos desechables.	Más Artefactos. El modelado es esencial, mantenimiento de modelos.
Pocos Roles, más genéricos y flexibles.	Más Roles, más específicos.
No existe un contrato tradicional, debe ser bastante flexible.	Existe un contrato prefijado.
Cliente es parte del equipo de desarrollo.	El cliente interactúa con el equipo de desarrollo mediante reuniones.
Orientada a proyectos pequeños. Corta duración (o entregas frecuentes), equipos pequeños (< 10 integrantes) y trabajando en el mismo sitio.	Aplicables a proyectos de cualquier tamaño, pero suelen ser especialmente efectivas/usadas en proyectos grandes y con equipos posiblemente dispersos.
La arquitectura se va definiendo y mejorando a lo largo del proyecto.	Se promueve que la arquitectura se defina tempranamente en el proyecto.
Énfasis en los aspectos humanos: el individuo y el trabajo en equipo.	Énfasis en la definición del proceso: roles, actividades y artefactos.
Se esperan cambios durante el proyecto.	Se espera que no ocurran cambios de gran impacto durante el proyecto.

Tabla 10: Diferencias entre metodologías ágiles y no ágiles.

Fuente: [10].

El desarrollo ágil de software nos permite realizar un proyecto más flexible y eficaz además de involucrar de manera más directa al cliente. En la Tabla (11) se presentan ejemplos de metodologías.

Metodología Ágil	Metodología Tradicional
Desarrollo de Software Adaptable (ASD)	Prototipato
OpenUP	Espiral
Scrum	Cascada

Tabla 11: Ejemplos de metodologías.

Elaborado por: El investigador.

3.1.2.1. Metodologías ágiles para el desarrollo de un sistema web eficiente

En la siguiente tabla se exponen con los aspectos más importantes a tomar en cuenta para la comparación e inmediata elección de la metodología ágil que permita documentar el desarrollo del sistema web eficiente para la Junta de Agua Potable Mollepamba.

Aspecto	Detalle
Trabajo unipersonal	Todas las etapas del desarrollo del software, se deben manejar por una sola persona.
Generación prototipo	La metodología debe permitir documentar avances incrementales basados en prototipos.
Integración del cliente	El cliente debe estar presente y participar en la revisión de los prototipos.
Documentación eficaz	Se debe permitir documentar lo estrictamente necesario. Se debe basar en el proceso y protección de código y datos. La documentación en ningún punto de la misma debe vulnerar la propiedad de los datos de la Junta y de sus socios.
Tecnicismos	Los aspectos técnicos del sistema web deben ser protegidos. Debe permitir adaptar nuevos documentos a la metodología.

Tabla 12: Requerimientos metodología ágil.

Elaborado por: El investigador.

Pos análisis de los requerimientos para la elección de una metodología ágil para el desarrollo de un sistema web eficiente, se realiza una comparativa de tres metodologías que se encuentran en la línea de los aspectos requeridos, esas son: Scrum, OpenUP y Extreme Programming. De acuerdo a las características del presente proyecto y realizando una comparativa de las metodologías se seleccionará la más adecuada para el desarrollo del sistema web eficiente.

En la tabla (13) se exponen características sobre cada una.

Scrum	OpenUP	Extreme Programming (XP)
Es apropiado para proyectos con equipos competitivos productivos.	Es apropiado para proyectos pequeños y de bajos recursos.	Es apropiado para proyectos startups o empresas en procesos de consolidación, puesto que se centra en la planificación con el cliente.
Es iterativo con tiempos de incrementos fijos.	Permite detectar errores tempranos a través de un ciclo iterativo[10].	Permite un desarrollo rápido, lo habitual el desarrollo se realiza en pareja.
Demostración de los resultados de proyecto en cada iteración.	Evita la elaboración de documentación, diagramas e iteraciones innecesarios requeridos en la metodología RUP.	Se tiene que documentar en el instructivo parte del código fuente de cualquier proyecto.
Gestión regular de las expectativas del cliente.	Por ser una metodología ágil tiene un enfoque centrado al cliente y con iteraciones cortas.	Iteraciones, y desarrollo del software en parejas.
Entregales funcionales cada 30 días.	Los avances del proyecto se realizan con micro-incrementos[10].	Micro avances modulares basados en las pruebas de código que van a ser implementados.

Tabla 13: Comparación entre Metodologías.

Elaborado por: El investigador.

Scrum y Extreme Programming son metodologías de desarrollo de software ágil que aplican de manera regular un conjunto de prácticas para trabajar en equipo, y obtener el mejor resultado posible en un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos. En este contexto, la metodología contempla un amplio registro en trabajos de larga duración con seguimiento de productividad personales.

En la presente investigación se considera los argumentos de la tabla 13 y de la tabla 10 para definir la metodología de desarrollo de software. En consideración al tamaño del proyecto, tiempo de entrega, número de personas que intervienen y de todos los requisitorios que conllevan su elección. En este contexto, se establece a OpenUP como la mejor metodología. Además es claro mencionar que esta metodología propone la generación de documentos estrictamente necesarios para el desarrollo del mismo; de esta forma se establecen controles iterativos directamente con el cliente en forma de microincrementos. Por estas razones se escogió OpenUp como metodología de desarrollo del Sistema web para el proceso de facturación, toma de lectura y consulta de planillas, en la junta administradora de agua potable “Mollepamba” de la parroquia Picaihua.

3.1.2.2. Metodología ágil OpenUp

OpenUP es un Proceso Unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado, utiliza una filosofía ágil que se enfoca en la naturaleza de colaboración en el desarrollo de software. Es una herramienta que puede extenderse para hacer frente a una amplia variedad de proyectos. Está basado en casos de uso, la gestión de riesgos, y una arquitectura centrada a impulsar el desarrollo[11].

Elementos OpenUP se organiza en dos dimensiones: Contenido metodológico y contenido procedimental. El contenido metodológico es el que define elementos metodológicos, tales como: Disciplinas, tareas, artefactos y procesos, independientemente de cómo se usen estos o se combinen[11]. El contenido procedimental, por el contrario, es donde se aplican todos estos elementos metodológicos dentro de una dimensión temporal, pudiéndose crear multitud de ciclos de vida diferentes a partir del mismo subconjunto de elementos metodológicos. En la Figura (3), se presentan los elementos que son parte de metodología que se usara en este proyecto.



Figura 3: Elementos de OpenUP.
Fuente: [11].

Ciclo de vida En todo proyecto que se desarrolle con la metodología OpenUP consta de cuatro fases, que son: Inicio, elaboración, construcción y transición. Cada una de estas se divide a su vez en iteraciones[11]. En la Figura (3.1.2.2), se expone las diferentes fases y las relaciones entre las mismas.

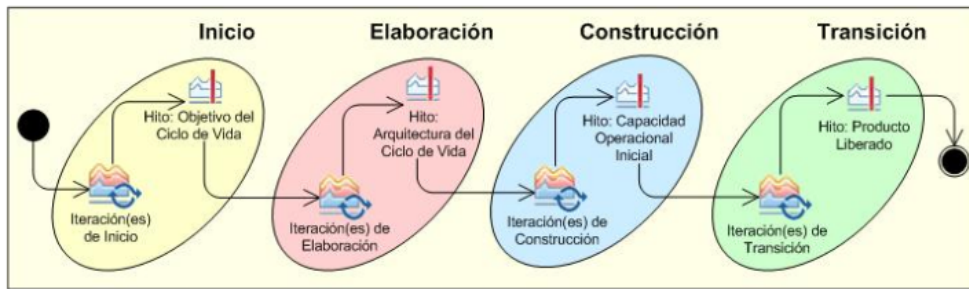


Figura 4: Ciclo de vida en OpenUP.
Fuente: [11].

- **Fase de inicio:** En esta fase, las necesidades de cada participante del proyecto son tomadas en cuenta y plasmadas en objetivos del proyecto. Se definen para el proyecto: el ámbito, los límites, el criterio de aceptación, los casos de uso críticos, una estimación inicial del coste y un boceto de la planificación[10].
- **Fase de elaboración:** En esta fase se realizan tareas de análisis del dominio y definición de la arquitectura del sistema. Se debe elaborar un plan de proyecto, estableciendo unos requisitos y una arquitectura estables. Por otro lado, el proceso de desarrollo, las herramientas, la infraestructura a utilizar y el entorno de desarrollo también se especifican en detalle en esta fase. Al final de la fase se debe tener una definición clara y precisa de los casos de uso, los actores, la arquitectura del sistema y un prototipo ejecutable de la misma[10].
- **Fase de construcción:** Todos los componentes y funcionalidades del sistema que faltan por implementar son realizados, probados e integrados en esta fase. Los resultados obtenidos en forma de incrementos ejecutables deben ser desarrollados de la forma más rápida posible sin dejar de lado la calidad de lo desarrollado[10].
- **Fase de transición:** Esta fase corresponde a la introducción del producto en la comunidad de usuarios, cuando el producto está lo suficientemente maduro. La fase de la transición consta de las subfases de pruebas de versiones, y capacitación de los usuarios finales. En función de la respuesta obtenida por los usuarios puede ser necesario realizar cambios en las entregas finales o implementar alguna funcionalidad más[11].

Prácticas OpenUP es una metodología basada en RUP (Proceso Unificado de Desarrollo), y por lo tanto, comparte las mismas prácticas que subyacen por debajo del flujo de trabajo y los roles de OpenUP[?]. En la Figura (3.1.2.2), se presentan las prácticas de trabajo de esta metodología.

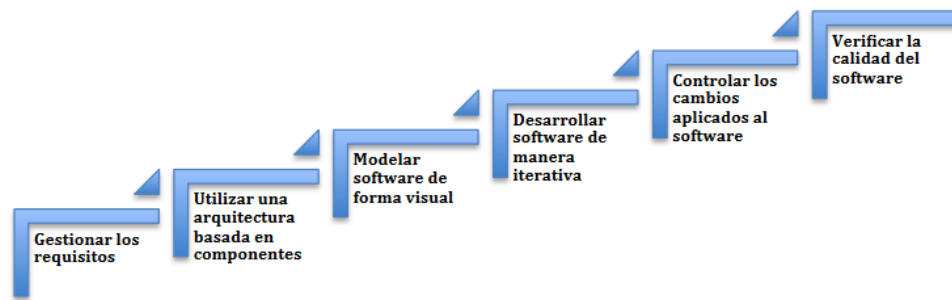


Figura 5: Practicas de OpenUP.
Fuente: [11].

3.1.3. Elección del framework para el desarrollo web

Un framework es una herramienta de desarrollo web que, por lo general, se define como una aplicación o conjunto de módulos que permiten el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya creadas, estas funcionalidades nos ayuda a optimizar tiempo de desarrollo y a tener sistemas más eficientes [12].

En la siguiente tabla define aspectos técnicos generales que son primordiales para el desarrollo de un sistema web eficiente.

Aspecto	Detalle
Acceso a base de datos	Se debe poder realizar migraciones y acceder a los datos a través de un ORM.
Generación de estructura de ingreso al sistema	Se debe poder generar un scaffold para el acceso al usuario sin que este tenga que ser desarrollado desde su inicio.
Arquitectura	El framework debe manejar una estructura MVC.
Seguridad para publicación	Lenguaje con código fuertemente tipado, en especial C#.
Integración modulares	El framework debe poder integrarse por módulos y componentes .Net

Tabla 14: Requerimientos técnicos framework.

Elaborado por: El investigador.

En el contexto de esta investigación se nombrará tres opciones de herramientas basadas en los requerimientos técnicos generales definidos en la tabla anterior. Además de abstraer la información del marco teórico del presente proyecto.

En la siguiente tabla se muestra una comparativa de los marcos de desarrollo: .NET, Spring y Laravel. Estos son los más usados para el desarrollo de sistemas web eficientes.

Características	.NET	Spring	Laravel
Definición	.NET es una plataforma de desarrollo de propósito general. Con .NET, puede utilizar varios idiomas, editores y bibliotecas para crear aplicaciones nativas para web, dispositivos móviles, escritorio, juegos e IoT para Windows, macOS, Linux, Android y más [13].	Spring Framework proporciona un modelo integral de programación y configuración para aplicaciones empresariales modernas basadas en Java, en cualquier tipo de plataforma de implementación. [13].	Es un marco de aplicación web con una sintaxis elegante y expresiva. Intenta aliviar la molestia del desarrollo al facilitar las tareas comunes que se utilizan en la mayoría de los proyectos web, como la autenticación, el enrutamiento, las sesiones y el almacenamiento en caché [14].
Lenguaje	C#	JAVA	PHP
Protección del código	Código fuertemente tipado [15].	Código multiplataforma [13].	Fuente abierta[15].
Documentación	Documentación intuitiva con talleres en línea [15].	Documentación basada en versiones e inyección de bibliotecas[13].	Documentación dependiente de paquetes externos[15].
Curva de aprendizaje	Media	Media	Baja
Mapeo de datos	Entity Framework Core	Dependiente de bibliotecas externas.	Eloquent
Experiencia	Media	Baja	Baja
Ofertas de empleo	Media [16].	Media[16].	Baja [16].

Tabla 15: Tabla .NET vs Spring vs Laravel.

Elaborado por: El investigador.

Bajo los requerimientos de la tabla 14, además de los argumentos presentados en la tabla 15 y los proyectos relacionados en el apartado antecedentes investigación 1.2 se determina a .NET como marco de desarrollo de este proyecto. En síntesis, este framework tiene muchas ventajas importantes como son: disminuir el tiempo de desarrollo de los proyectos, ya que tiene implementadas librerías y funcionalidades que se puede reutilizar durante todo el proyecto, además de las ventajas de la plataforma en: documentación, protección de código, curva de aprendizaje, ofertas de empleo y mi experiencia con el entorno de desarrollo son hechos que justifican su selección.

3.1.3.1. Framework marco .NET

.NET es una plataforma de desarrollador de código abierto gratuita y multiplataforma para crear muchos tipos diferentes de aplicaciones. Con .NET, puede usar varios

lenguajes, editores y bibliotecas para crear juegos y aplicaciones para la Web, dispositivos móviles y de escritorio, y el Internet de las cosas (IoT)[17].

Interfaz de usuario web interactiva con C # Blazor le permite crear interfaces de usuario web interactivas utilizando C # en lugar de JavaScript. Las aplicaciones Blazor se componen de componentes de interfaz de usuario web reutilizables implementados con C #, HTML y CSS. Tanto el código del cliente como el del servidor están escritos en C #, lo que le permite compartir código y bibliotecas[18].

Blazor es una característica de ASP.NET, el popular marco de desarrollo web que amplía la plataforma de desarrollo .NET con herramientas y bibliotecas para crear aplicaciones web.



```
<table class="table">
  <thead>
    <tr>
      <th>@Html.DisplayNameFor(model => model.Name)</th>
      <th>@Html.DisplayNameFor(model => model.PhoneNumber)</th>
      <th>@Html.DisplayNameFor(model => model.Email)</th>
    </tr>
  </thead>
  <tbody>
    @foreach (var item in Model) {
      <tr>
        <td>@Html.DisplayFor(modelItem => item.Name)</td>
        <td>@Html.DisplayFor(modelItem => item.PhoneNumber)</td>
        <td>@Html.DisplayFor(modelItem => item.Email)</td>
      </tr>
    }
  </tbody>
</table>
```

The screenshot shows a Blazor component with a table. The table has three columns: Name, PhoneNumber, and Email. The columns are rendered using `@Html.DisplayNameFor` and the rows are rendered using `@Html.DisplayFor` within a `@foreach` loop. A tooltip is visible over the `item.Email` property, showing the type `string Person.Email { get; set; }` and a list of methods: Email, Equals, GetHashCode, and GetType.

Figura 6: Microsoft blazor.

Fuente: [18].

Integración con datos La popular biblioteca de acceso a datos de Entity Framework (EF) le permite interactuar con bases de datos utilizando objetos fuertemente tipados.

Se admiten las bases de datos más populares, incluidas SQLite, SQL Server, MySQL, PostgreSQL, DB2 y más, así como tiendas no relacionales como MongoDB, Redis y Azure Cosmos DB[17].

Modelo Con EF Core, el acceso a datos se realiza mediante un modelo. Un modelo se compone de clases de entidad y un objeto de contexto que representa una sesión con

la base de datos. Este objeto de contexto permite consultar y guardar datos. Para más información, vea Creación de un modelo.

EF admite los siguientes métodos de desarrollo de modelos:

Generar un modelo a partir de una base de datos existente. Codificar un modelo manualmente para que coincida con la base de datos. Una vez creado un modelo, usar Migraciones de EF para crear una base de datos a partir del modelo. Migraciones permite que la base de datos evolucione a medida que el modelo va cambiando[19].

Consultas Las instancias de las clases de entidad se recuperan de la base de datos por medio de Language Integrated Query (LINQ)[19]

Para documentación completa sobre el framework visita la página oficial [19].

3.1.4. Aplicación de la metodología OpenUP en el desarrollo del proyecto

En esta sección se generarán los documentos necesarios para la gestión del proyecto. Estos son: Documento de especificación de requisitos, documento de arquitectura de software, plan de pruebas, instalación del software, aceptabilidad del software.

3.1.4.1. Plan de Desarrollo de Software

Propósito En el plan de desarrollo de software el objetivo es describir las actividades que se desarrollarán en cada una de las fases de la metodología aplicada para el sistema web que mejore el proceso de facturación, toma de lectura y consulta de planillas, de la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua.

Alcance En el Plan de Desarrollo de software se especifica las actividades más importantes a ser documentadas durante el desarrollo de un sistema web que mejore el proceso de facturación, toma de lectura y consulta de planillas, de la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua.

Definiciones, Acrónimos y Abreviaciones

- **Casos de Uso:** Los casos de uso indican la interacción del usuario con el software .
- **Iteración:** Iteración es el acto de repetir un proceso para así cumplir con un objetivo.
- **OpenUP:** metodología de desarrollo de software.

- **Recurso:** Los recursos son los elementos disponibles para el desarrollo de un proyecto.
- **Usuario:** El usuario es la persona que utilizará el software, en este caso son clasificados por el alcance del proyecto.

3.1.4.2. Resumen Plan del Proyecto OpenUP

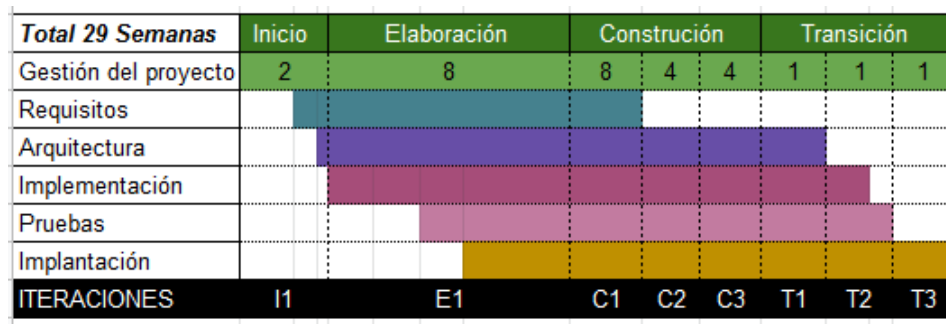


Figura 7: Resumen del Plan del Proyecto.

3.1.4.3. Productos del proyecto

Actividad	Iteración
Plan de desarrollo de Software	1
Especificación de los requisitos del Software	3
Documento de arquitectura del Software	1
Definición de pruebas para el Software	1
Desarrollo del primer prototipo	3
Desarrollo de pruebas	3
Primer prototipo funcional	3
Refinamiento de los requerimientos funcionales	3
Refinamiento de las pruebas	3
Segundo prototipo funcional	3
Implantación	1

Tabla 16: Productos del proyecto.

Organización del proyecto En la organización del proyecto hay que tener en cuenta a las personas con las que estaremos trabajando durante el proyecto y tener claro cuáles son sus funciones durante el mismo. Para el presente proyecto hemos identificado a los siguientes roles.

Nombre	Walter Semblantes
Rol	Desarrollador del Proyecto de Investigación
Información de contacto	danilosemblantes@gmail.com

Tabla 17: Desarrollador del proyecto de investigación.

Nombre	Franklin Mayorga
Rol	Tutor del Proyecto de Investigación
Información de contacto	fmayorga@uta.edu.ec

Tabla 18: Tutor del proyecto de investigación.

Nombre	José Criollo
Rol	Propietario del producto de software
Información de contacto	josecriollo@gmail.com

Tabla 19: Propietario del software.

Estimación del proyecto Los costos generados por el proyecto están detallados en la propuesta del mismo. No obstante se estimó un valor total de 1320 dólares americanos.

Se estima una entrega del proyecto con un 50% de funcionalidad al finalizar la primera iteración. El 100% del proyecto será entregada en la tercera iteración.

Plan de proyecto El desarrollo del proyecto está basado en fases, donde en cada fase vamos cumpliendo con la documentación y estas tienen un número de interacciones, es decir las iteraciones son las repeticiones que se puede ir dando en cada fase de desarrollo. La tabla 20 muestra las fases con sus iteraciones y el tiempo estimado que llevara cada una.

Fase	No. Iteraciones	Duración
Fase de Inicio	1	2 Semanas
Fase de Elaboración	1	8 Semanas
Fase de Construcción	3	16 Semanas
Fase de Transición	3	3 Semanas

Tabla 20: Plan de Fases.

3.2. Especificación de requisitos de Software

3.2.1. Introducción

3.2.1.1. Propósito

El documento de especificación de requisitos de software tiene como objetivo definir los requisitos funcionales, no funcionales de un sistema web que mejore el proceso de facturación, toma de lectura y consulta de planillas, de la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua que será utilizada por los clientes, administradores, y encargados del mismo.

3.2.1.2. Alcance

El sistema web que mejore el proceso de facturación, toma de lectura y consulta de planillas, de la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua se basará su desarrollo en los requerimientos funcionales que lleven a la aceptabilidad del software.

3.2.1.3. Definiciones, Acrónimos y Abreviaciones

- **SWP:** abreviatura para sistema web que mejore el proceso de facturación, toma de lectura y consulta de planillas, de la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua.
- **ERS:** abreviatura de Especificación de Requisitos Software.
- **CRUD:** acrónimo para: Crear, Leer, Actualizar y Borrar registros sobre un módulo específico

3.2.1.4. Características de los usuarios

El SWP contara con los tipos de usuarios: Recaudador, Lector, Cliente descritos en las siguiente tabla.

Tipo de Usuario	Formación	Habilidades	Actividades
Recaudador	Persona con conocimientos sobre el manejo de Excel y sistemas contables.	Conocimientos sobre manejo de un computador, manejo de office.	Persona encargada en cobrar a los clientes o socios el valor por consumo de agua potable. Además de realizar operaciones sobre los distintos módulos para una fácil administración de la Junta de Agua Potable. Control total.
Lector	Persona con conocimientos sobre consultas y manejos de sitios web y redes sociales	Conocimientos sobre aplicaciones móviles, plomería, y construcción.	Acceso total al módulo de lecturas. Este le permitirá crear y actualizar las lecturas de los medidores de agua potable.
Cliente	Persona cliente de la Junta Administradora de Agua Potable “Mollepamba” de la parroquia Picaihua.	Sin habilidades definidas.	Consulta de las planillas de agua potable adeudar según el número de medidor.

Tabla 21: Características de los usuarios.

3.2.1.5. Restricciones

- **Políticas Regulatorias:** El SWP se soportará en cualquier navegador con las características de proporciona la plataforma .NET, estos son: Internet Explorer 9+, Microsoft Edge, Safari, Firefox, Google Chrome, Opera.
- **Limitaciones de Hardware:** El SWP podrá ser utilizado en cualquier equipo de cómputo que soporte los navegadores mencionados en las políticas antes mencionadas.
- **Consideraciones de Seguridad:** La seguridad del SWP hereda las clases de conexión, validación, y restricciones que maneja el framework entity framework. Sin embargo, al poner en marcha el producto final se establecerán parámetros de seguridad definidas por google para los desarrolladores web.

3.2.1.6. Suposiciones y Dependencias

- El SWP se implantará en un servidor web con soporte para aplicaciones web desarrolladas en la plataforma .Net.
- El SWP se maneja su base de datos en SQLServer 2017 Express.
- Los clientes usarán el SWP para consultar el valor que adeudan por consumo de agua potable.

3.2.2. Requisitos Funcionales

3.2.2.1. Módulo: Categorías

CRUD categorías

Número de requisito	RF0001
Nombre de requisito	CRUD categorías
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir: Crear nuevas categorías con los siguientes datos: nombre, valor mínimo, valor máximo. Actualizar los datos de la categoría. Leer las categorías usando filtros de sus datos.

Tabla 22: Requisito funcional: CRUD categorías.

CRUD tarifas

Número de requisito	RF0002
Nombre de requisito	Ingreso al sistema
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Lector
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir: Crear nuevas tarifas según una categoría con los siguientes datos: nombre, valor base, valor exceso. Actualizar los datos de la tarifa. Leer las tarifas usando filtros de sus datos.

Tabla 23: Requisito funcional: CRUD tarifas.

3.2.2.2. Módulo: Rubros

CRUD rubros

Número de requisito	RF0003
Nombre de requisito	CRUD rubros
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir: Crear rubros con los siguientes datos: nombre, valor, descripción. Actualizar los datos de los rubros. Leer los rubros. Desactivar o activar rubros.

Tabla 24: Requisito funcional: CRUD rubros.

3.2.2.3. Módulo: Configuración

Configuración de número de facturas

Número de requisito	RF0004
Nombre de requisito	Configuración de número de facturas
Tipo	<input checked="" type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir actualizar los campos de inicio y fin del secuencial de los números de la factura.

Tabla 25: Requisito funcional: configuración.

3.2.2.4. Módulo: Usuarios

Ingreso al sistema

Número de requisito	RF0005
Nombre de requisito	Ingreso al sistema
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP será capaz de permitirle ingresar al sistema a través de sus credenciales, es decir mediante un usuario y contraseña.

Tabla 26: Requisito funcional: Ingreso al sistema.

Número de requisito	R0001
Nombre de requisito	Ingreso al sistema
Tipo	<input type="checkbox"/> Requisito <input checked="" type="checkbox"/> Restricción
Actor	Lector
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP será capaz de permitirle ingresar al sistema a través de sus credenciales, es decir mediante un usuario y contraseña. Con acceso solo al módulo de lecturas para la ejecución de sus actividades previamente definidas en la tabla 3.2.1.4.

Tabla 27: Requisito funcional: Ingreso al sistema (restricción).

CRUD usuarios

Número de requisito	RF0006
Nombre de requisito	CRUD usuarios
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP será capaz de permitir crear un nuevo usuario con los siguientes datos: nombre, correo electrónico, cédula, contraseña y rol. Se debe permitir actualizar todos sus datos. Se debe permitir eliminar usuarios o desactivarlos.

Tabla 28: Requisito funcional: CRUD usuarios.

3.2.2.5. Módulo: Clientes

CRUD clientes

Número de requisito	RF0007
Nombre de requisito	CRUD clientes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Programador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir: Crear nuevo clientes con los siguientes datos: nombres, apellidos, dirección, fecha de nacimiento y estado. Actualizar los datos del cliente. Leer los clientes usando filtros de sus datos. Desactivar y activar un usuario.

Tabla 29: Requisito funcional: CRUD clientes.

CRUD medidores

Número de requisito	RF0008
Nombre de requisito	CRUD medidores
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir: Crear uno o más medidores por cliente con los siguientes datos: código, categoría, observación. Leer los medidores por cliente. Actualizar los datos del medidor. Desactivar o activar uno o todos los medidores de un cliente.

Tabla 30: Requisito funcional: medidores.

3.2.2.6. Módulo: Lecturas

CRUD lecturas

Número de requisito	RF0009
Nombre de requisito	CRUD lecturas
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Recaudador, Lector
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir: Crear una lectura por medidor mensualmente con los siguientes datos: fecha, lectura anterior, lectura actual, consumo, estado, observación. Actualizar todos sus datos. Leer las lecturas por medidor. Eliminar lecturas por medidor.

Tabla 31: Requisito funcional: lecturas.

3.2.2.7. Módulo: Cuentas por cobrar

CRUD cuentas por cobrar

Número de requisito	RF0010
Nombre de requisito	Listar las vistas de una base datos
Tipo	<input checked="" type="checkbox"/> Requisito [] Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Eencial [] Media/Deseado [] Baja/Opcional
Descripción	El SWP debe permitir: Crear una cuenta por cobrar por medidor con los siguientes datos: fecha, medidor, cuotas, valor, saldo, estado, lectura actual, observación. Crear los detalles de la cuenta por cobrar basados en los registros del módulo de rubros. Leer las cuentas por cobrar por medidor y/o cliente según el caso. Eliminar cuentas por cobrar según su estado.

Tabla 32: Requisito funcional: cuentas por cobrar.

3.2.2.8. Módulo: Consultas planilla

Listar deudas del cliente

Número de requisito	RF0012
Nombre de requisito	Consultas de planillas
Tipo	<input checked="" type="checkbox"/> Requisito [] Restricción
Actor	Cliente
Prioridad	<input checked="" type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado [] Baja/Opcional
Descripción	El SWP debe permitir: Debe mostrar las deudas pendientes de consumo de agua potable según el número de medidor consultado.

Tabla 33: Requisito funcional: Consulta de planilla.

3.2.2.9. Módulo: Facturación

facturar consumo de agua potable

Número de requisito	RF0012
Nombre de requisito	Facturación
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir: seleccionar un medidor y desplegar los consumos por cobrar del mismo. seleccionar y agregar a la factura cuentas por cobrar en cómodas cuotas. listar y revisar las facturas del medidor seleccionado. anular las facturas

Tabla 34: Requisito funcional: Facturación.

3.2.2.10. Módulo: Reportes

Reportes de facturación y administración

Número de requisito	RF0013
Nombre de requisito	Reportes
Tipo	<input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción
Actor	Recaudador
Prioridad	<input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional
Descripción	El SWP debe permitir: extraer información sobre cada uno de los módulos existente en el sistema. incluir gráficos estadísticos, cálculos, filtros, y más que reflejen la realidad de la Junta administradora de Agua Potable de Mollepamba

Tabla 35: Requisito funcional: Reportes.

3.2.3. Requisitos no funcionales

3.2.3.1. Requisitos de Rendimiento

El SWP deberá poder conectarse a internet y mantener la conexión durante las operaciones necesarias que los usuarios deseen, en este punto el rendimiento del SWP es directamente proporcional a la capacidad de respuesta del servidor, ya sea que el mismo se encuentre trabajando en un servicio contratado o en equipos

propios. También hay que considerar varias características, por ejemplo número de usuarios conectados simultáneamente, teniendo en cuenta que todos los sitios Web son diferentes debido a su complejidad hay que verificar características indispensables como CPU, RAM, DISCO. Sin embargo para poder tener una referencia de las características mínimas que necesitamos podemos decir que en promedio un sitio web básico es decir, con solo los plugins necesarios, caché, CDN, imágenes optimizadas y sin videos en autoplay en un Plan 1 puede recibir hasta unas 150 visitas simultaneas, mientras que un Plan Premium 1 puede recibir al menos 400, y un VPS 3 unas 1000 simultáneas, características que debemos tomar en cuenta al momento de seleccionar el servicio de hosting.

3.2.3.2. Seguridad

El SWP es dependiente de la seguridad del servidor web en el que se implante el sistema, además de la seguridad de la base de datos seleccionada y los entornos y procedimientos utilizados en el desarrollo del mismo. A nivel de programación en ASP.NET Core implementamos varios métodos que nos ayuda mediante tokens de seguridad a prevenir ataques CSRF (Cross-Site Request Forgery), ASP.NET MVC usa tokens anti-falsificación (anti-forgery tokens), también llamados tokens de verificación de solicitud. Los anti-forgery tokens funcionan de la siguiente manera:

- El cliente solicita una página HTML que contiene un formulario.
- El servidor incluye dos tokens en la respuesta. Se envía un token como cookie. El otro se coloca en un campo de formulario oculto. Los tokens se generan aleatoriamente para que un adversario no pueda adivinar los valores.
- Cuando el cliente envía el formulario, debe enviar ambos tokens al servidor. El cliente envía el token de la cookie como una cookie y envía el token del formulario dentro de los datos del formulario. (Un cliente de navegador hace esto automáticamente cuando el usuario envía el formulario).
- Si una solicitud no incluye ambos tokens, el servidor rechaza la solicitud

De esta manera podemos evitar los ataques CSRF, ya que de manera oculta también validamos que la información sea enviada de un formulario de nuestro sitio web.

3.3. Documento de arquitectura de software

3.3.1. Introducción

3.3.1.1. Propósito

El presente documento, tiene como propósito presentar la arquitectura del sistema para el proceso de facturación, toma de lectura y consulta de planillas, en la junta administradora de agua potable.

3.3.1.2. Alcance

La descripción del documento se encuentra basado en diferentes perspectivas lógicas que ayuda a describir los aspectos más importantes de la arquitectura para la SWP para el proceso de facturación, toma de lectura y consulta de planillas, en la junta administradora de agua potable.

3.3.1.3. Organización del documento

El documento se organiza en base a la plantilla Software Architecture Document del proceso de desarrollo elaborado por OpenUp, la misma que es adaptada a las características particulares del tipo de proyecto en desarrollo.

3.3.2. Representación de la arquitectura

El modelo propuesto utiliza las siguientes vistas:

- **Vista de casos de usos:** Los casos de usos representa funcionalidades centrales del SWP, describen las funcionalidades del sistema, funciones o acciones que debe realizar el sistema.
- **Vista lógica:** da una representación de los datos, muestra un subconjunto significativo arquitectónicamente del modelo de diseño, es decir, un subconjunto de las clases, subsistemas y paquetes, y realizaciones de guiones de uso
- **Vista de despliegue:** ilustra la distribución de procesos en un conjunto de nodos del sistema, es decir, muestra la comunicación y relaciones existentes entre los diferentes nodos que componen los escenarios antes mencionados
- **Vista de implementación:** Nos permite llevar el control de todos los artefactos que se utiliza para poner el sistema en producción.
- **Vista de datos:** Nos muestra los elementos principales del Modelo de Datos, dicho modelo es llevado a un conjunto de tablas .

3.3.3. Objetivos y restricciones de la arquitectura

El desarrollo del SWP está orientado a ser un software de calidad que sea de fácil manejo para el usuario final, teniendo en cuenta aspectos que faciliten el mantenimiento o reutilización de los componentes del mismo.

3.3.4. Vista de Casos de Uso

3.3.4.1. Introducción

La vista de Casos de Uso describe el comportamiento de los sistemas o software, mas no describen alguna funcionalidad interna

3.3.4.2. Identificación de los Casos de Uso relevantes para la arquitectura

Para el diseño de la SWP que genera de forma automática el proceso de facturación, toma de lectura y consulta de planillas, se identifican como casos de uso importantes a los siguientes:

- **Características de los Usuarios**
- **CRUD Categorías**
- **CRUD Tarifas**
- **CRUD Rubros**
- **Configuración del Número de Facturas**
- **Ingreso al Sistema**
- **CRUD Usuarios**
- **CRUD Clientes**
- **CRUD Medidores**
- **CRUD Lecturas**
- **CRUD Cuentas por Cobrar**
- **CRUD Listar deudas del cliente**
- **Consulta de Planillas**
- **Facturar Consumo de Agua Potable**

■ **Reporte Facturación y administración**

CARACTERÍSTICAS DE LOS USUARIOS

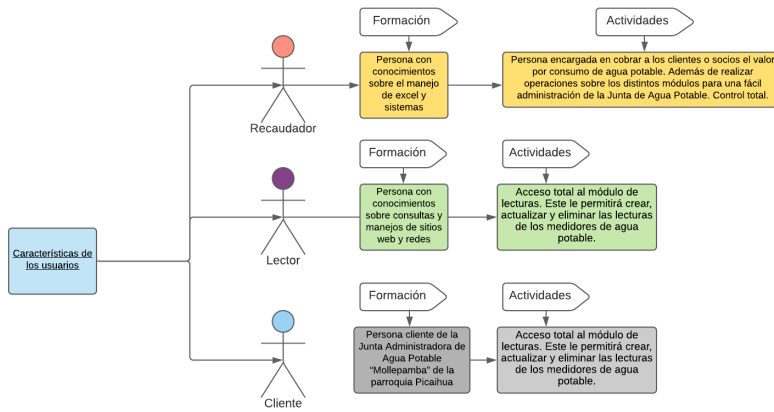


Figura 8: Características de los Usuarios

CRUD TARIFAS

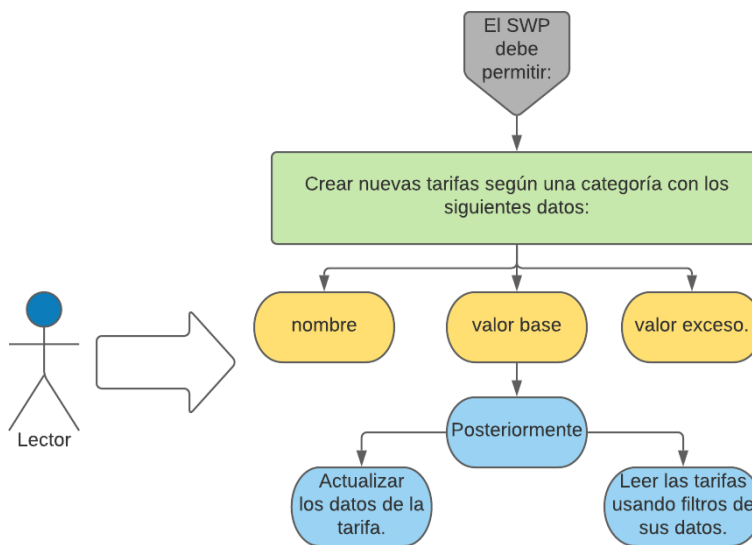


Figura 9: Requisito Funcional: CRUDtarifas

CRUD RUBROS

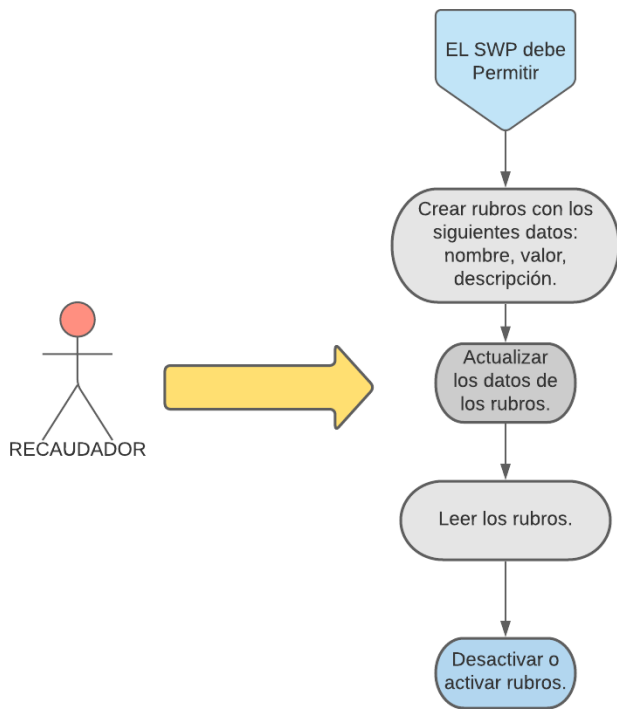
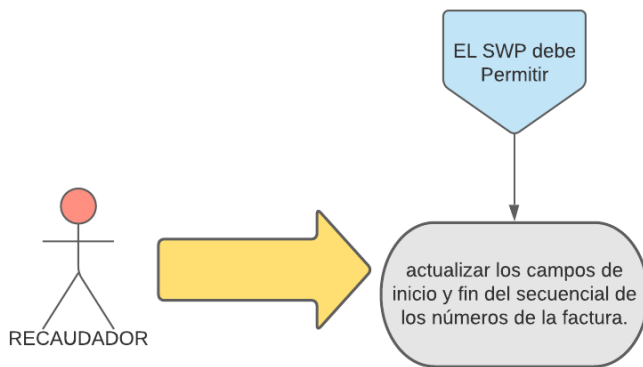


Figura 10: Requisito Funcional CRUD rubros

CONFIGURACIÓN DEL NÚMERO DE FACTURAS

Figura 11: Requisitos Funcional Configuración



INGRESO AL SISTEMA

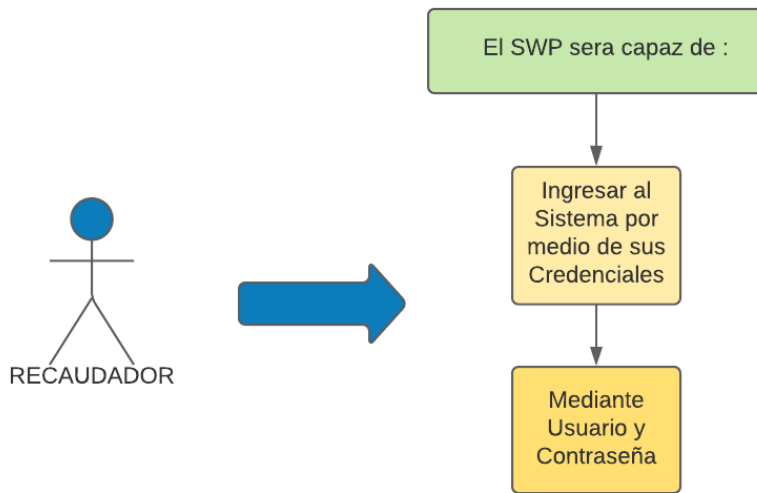


Figura 12: Requisito Funcional Ingreso al Sistema

INGRESO AL SISTEMA RESTRICCIÓN

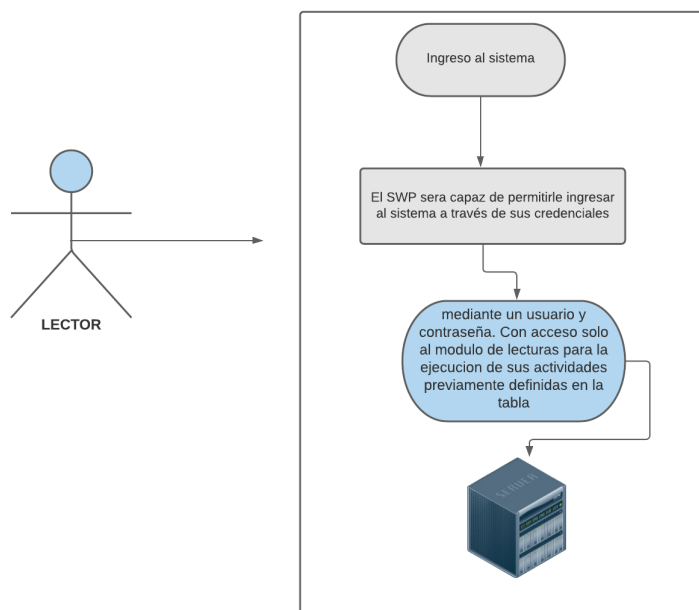


Figura 13: Requisito funcional: Ingreso al sistema (restricción)

CRUD USUARIOS

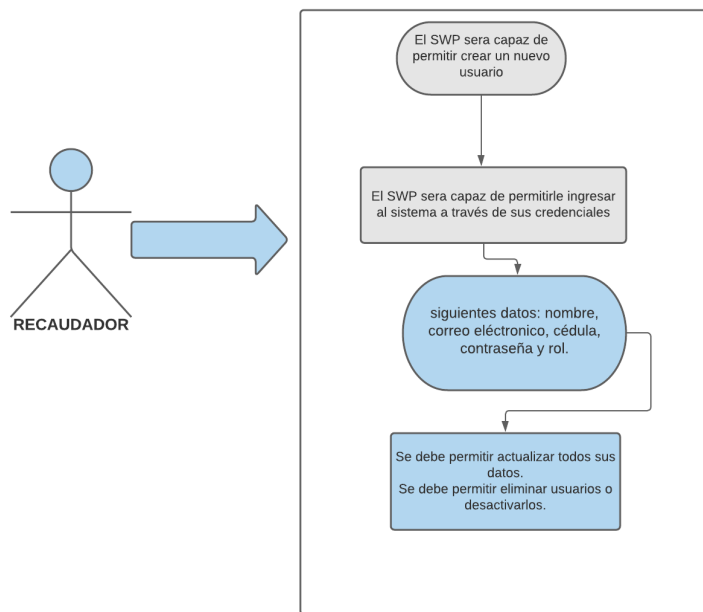


Figura 14: Requisito Funcional CRUD usuarios

CRUD CLIENTES

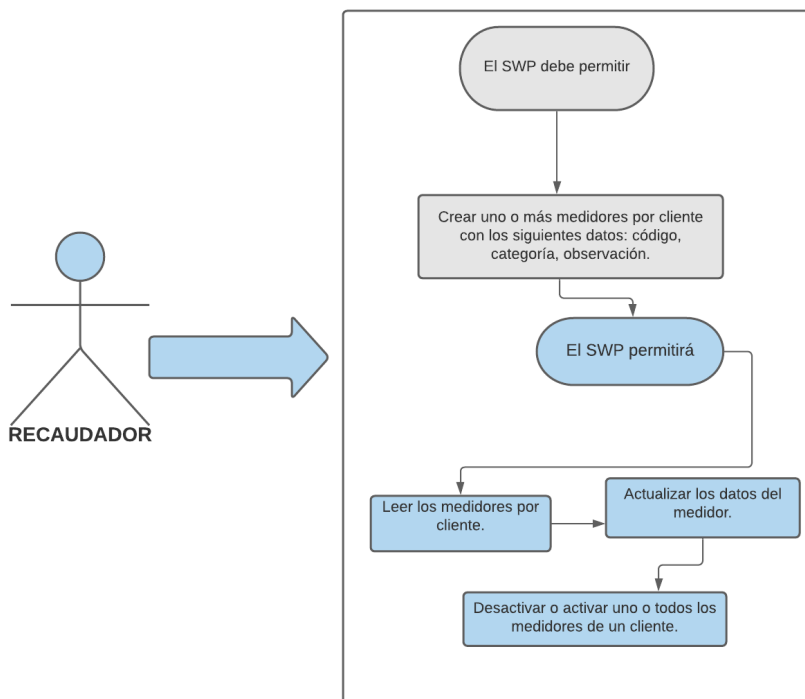


Figura 15: Requisito funcional CRUD clientes

CRUD MEDIDORES

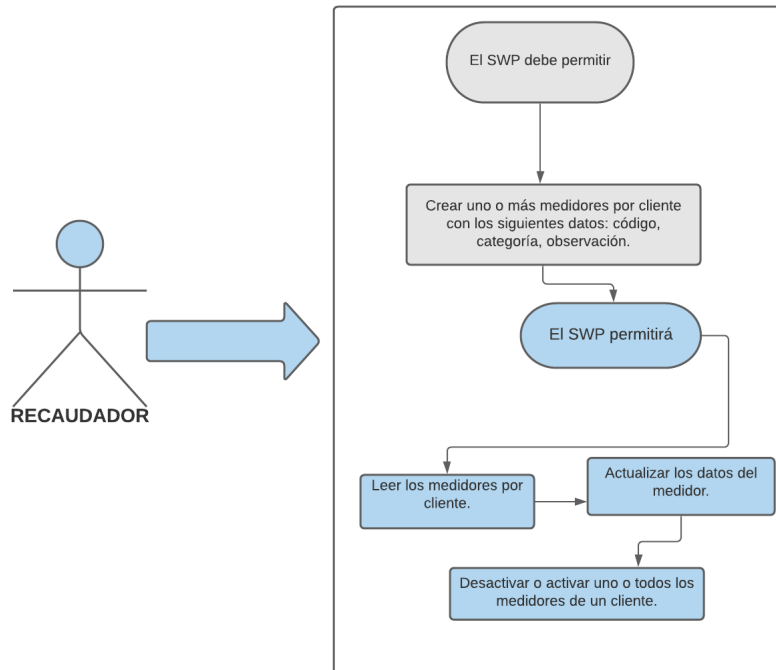


Figura 16: Requisito Funcional: Medidores

CRUD LECTURAS

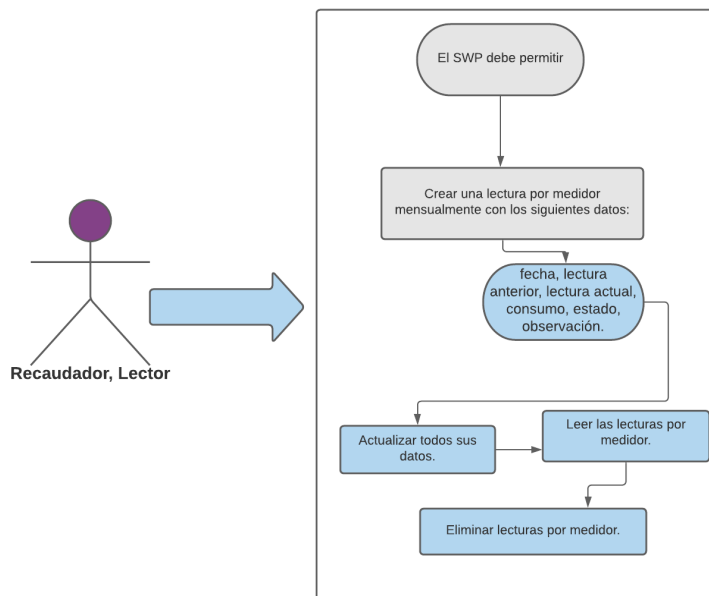


Figura 17: Requisito Funcional: Lecturas

CRUD CUENTAS POR COBRAR

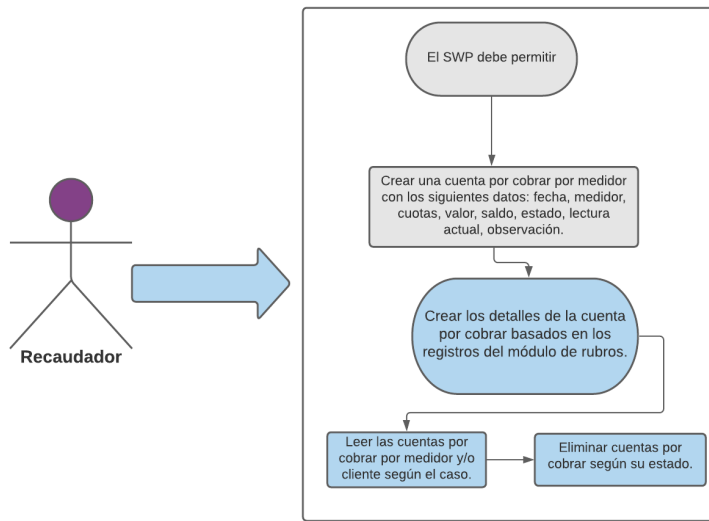


Figura 18: Requisito Funcional Cuentas por Cobrar

CRUD FACTURACIÓN

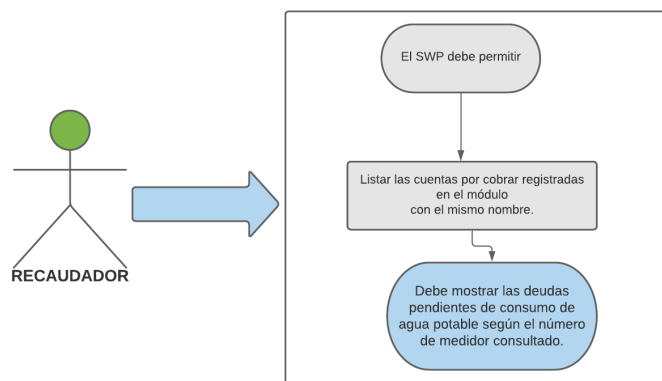
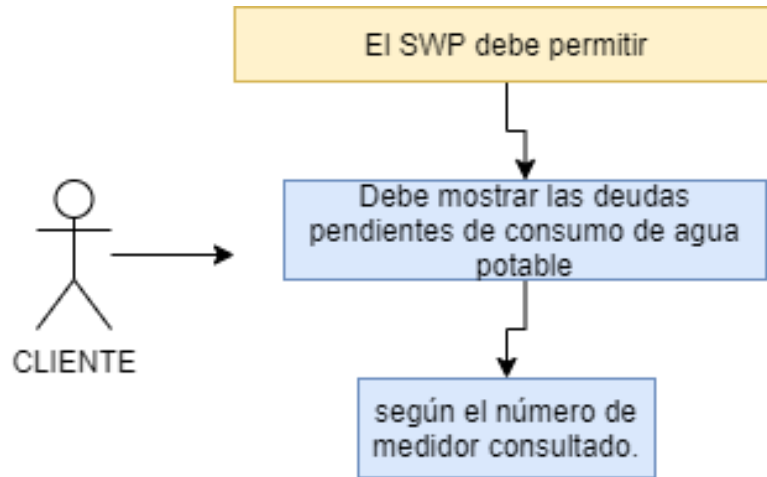


Figura 19: Requisito Funcional: Consulta de planilla

MODULO CONSULTAS DE PLANILLA

Figura 20: Listar deudas del cliente



MODULO FACTURACIÓN

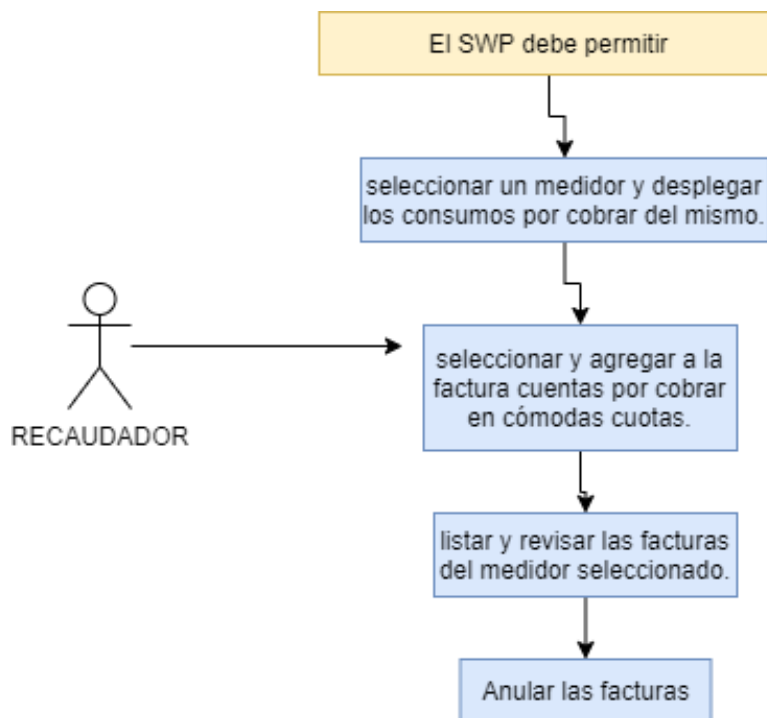


Figura 21: facturar consumo de agua potable

MODULO REPORTES

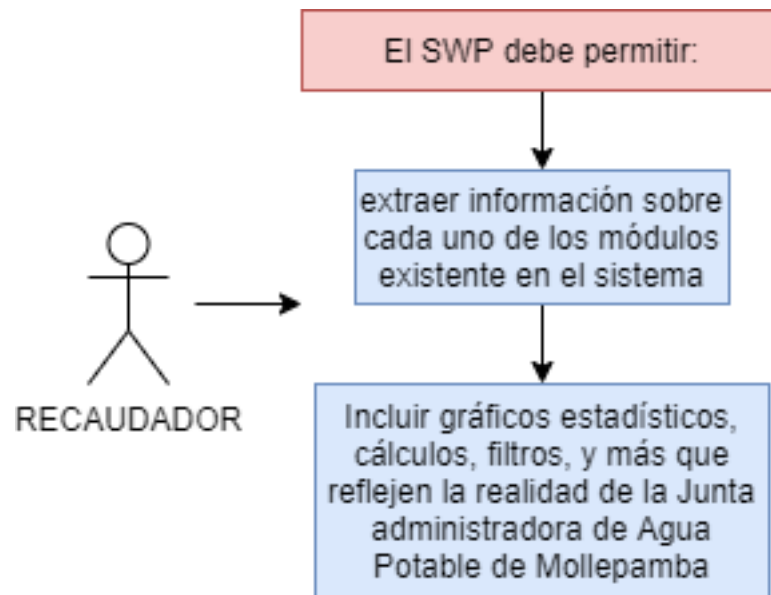


Figura 22: **Reporte Facturación y administración**

3.3.4.3. Descripción de los casos de uso relevantes para la arquitectura

Remitirse a la sección 3.2.2 Requisitos Funcionales del documento de Especificación de Requisitos de Software.

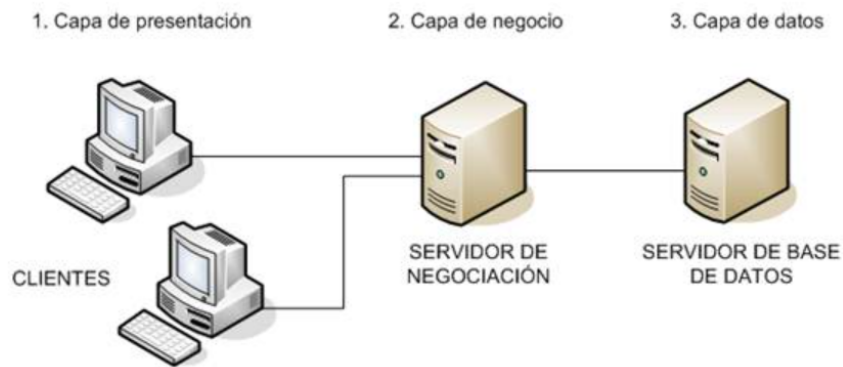
3.3.5. Vista Lógica

3.3.5.1. Introducción

Consiste en la descomposición de la SWP en subsistemas cada uno de estos tiene un grupo de diferentes funcionalidades relacionadas entre sí, es decir si unimos los subsistemas podemos funcionar como un solo sistema.

3.3.5.2. Descomposición en Subsistemas

Figura 23: Descomposición en Subsistemas



3.3.5.3. Descripción de los Subsistemas

1. Sistema de Consulta de Planillas: Este subsistema permitirá la consulta de la planilla con el consumo real, mediante el ingreso a la plataforma Web.
2. Sistema de facturación, toma de lectura y consulta de Planillas de la Junta de Agua Potable: El sistema garantizará el proceso de facturación mediante la toma de lectura y consulta de las planillas correspondientes a cada uno de los usuarios que conforman o hacen uso de los recursos hídricos administrados por la Junta de Agua Potable..

3.3.5.4. Diseño de Subsistemas

Sistema de Consulta de Planillas

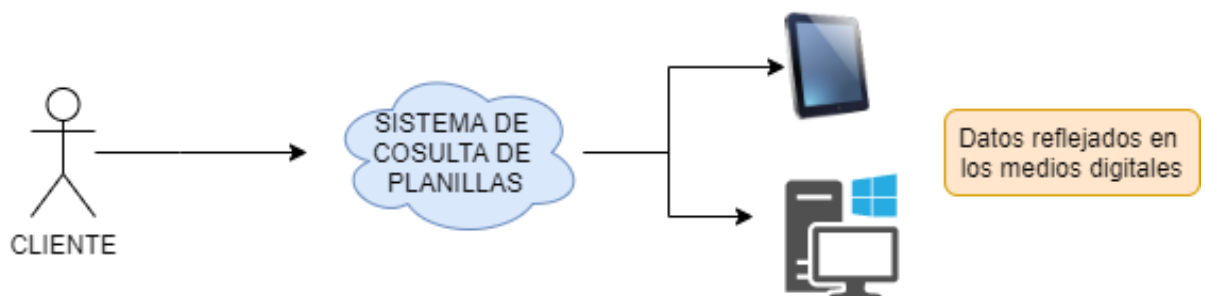


Figura 24: Consulta de Planillas

Sistema de Facturación, toma de lectura y consulta de Planillas de la Junta de Agua Potable

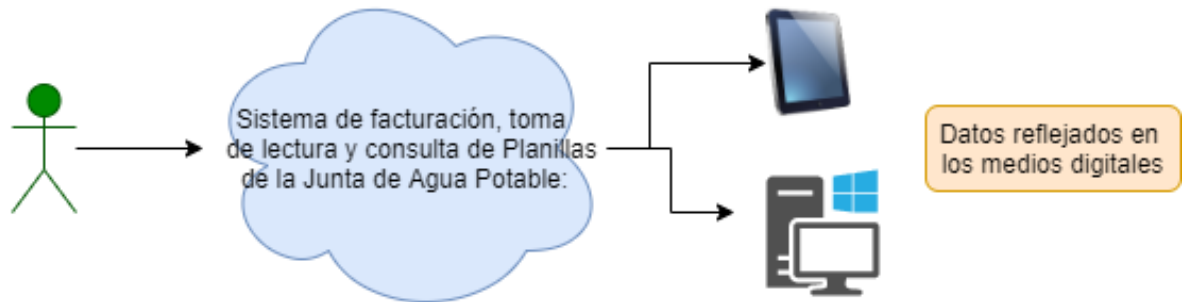


Figura 25: Sistema de Facturación

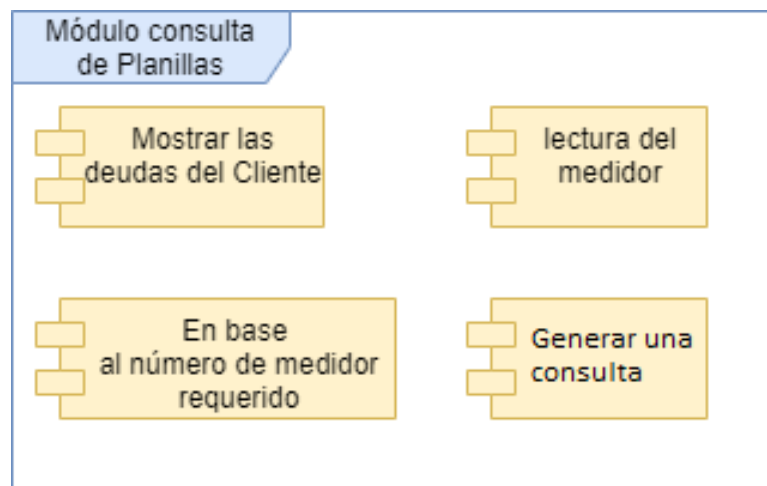


Figura 26: Módulo Consulta de Planillas

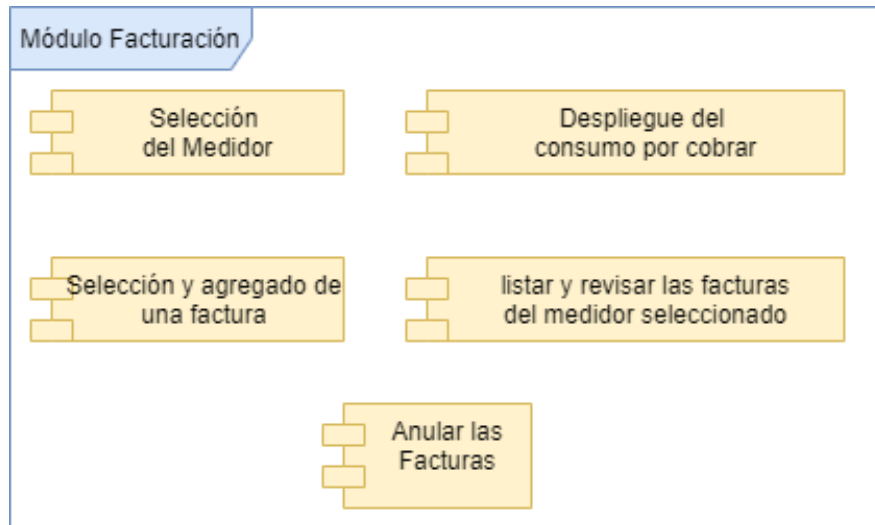


Figura 27: Módulo Facturación

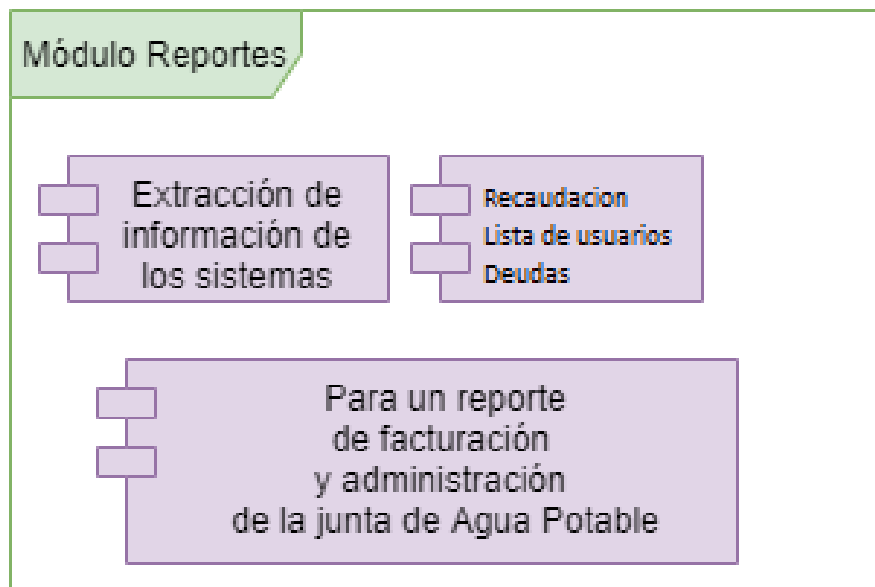


Figura 28: Módulo Reportes

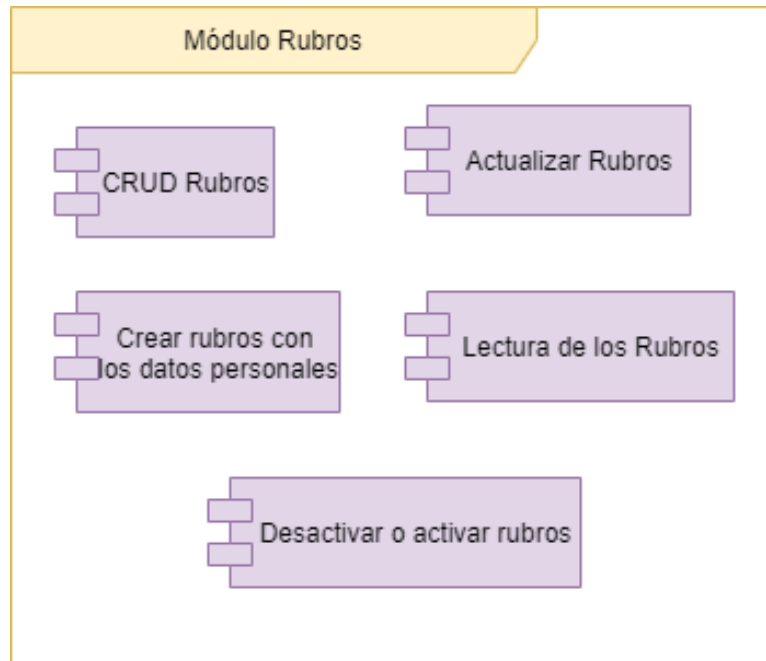


Figura 29: Módulo Rubros

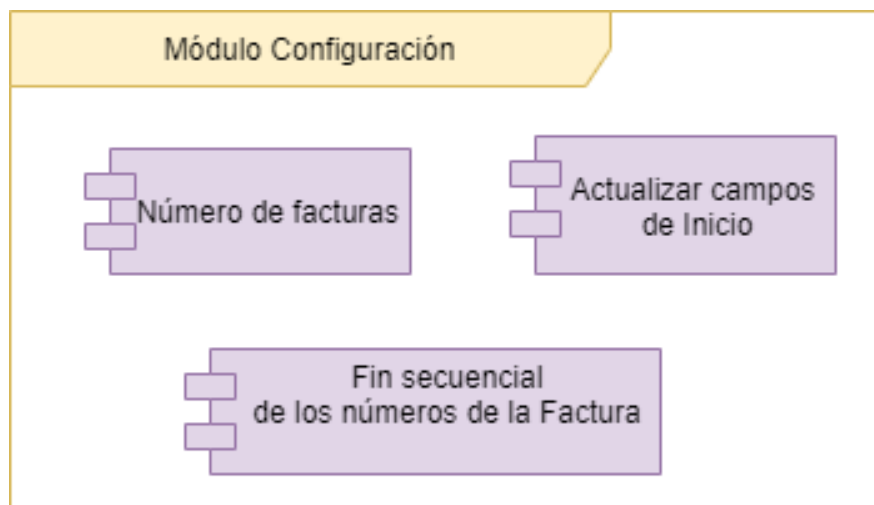


Figura 30: Módulo Configuración

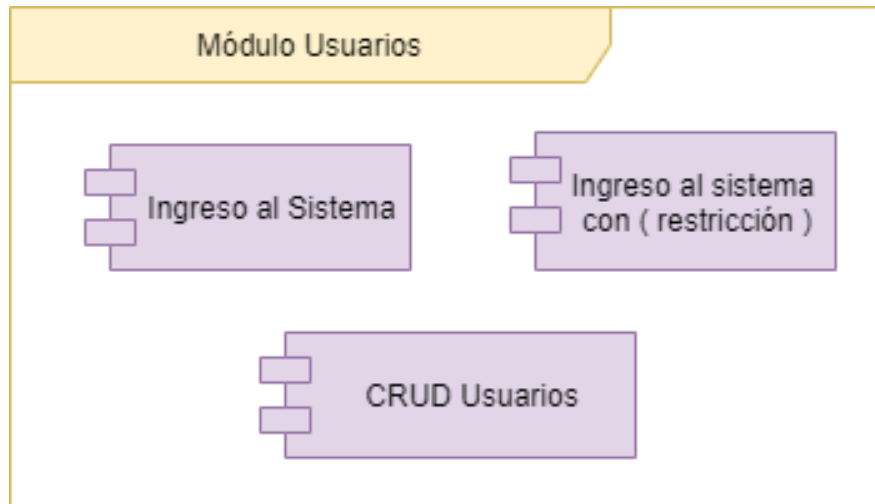


Figura 31: Módulo Usuarios

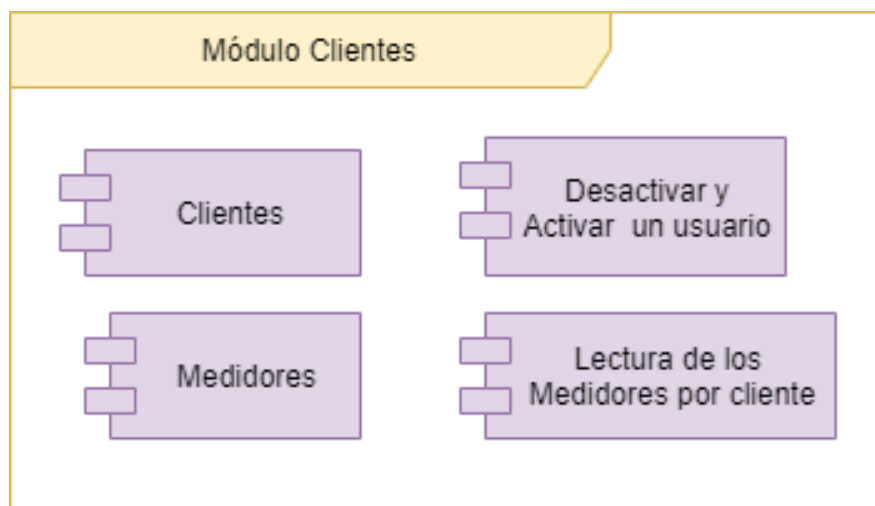


Figura 32: Módulo Clientes

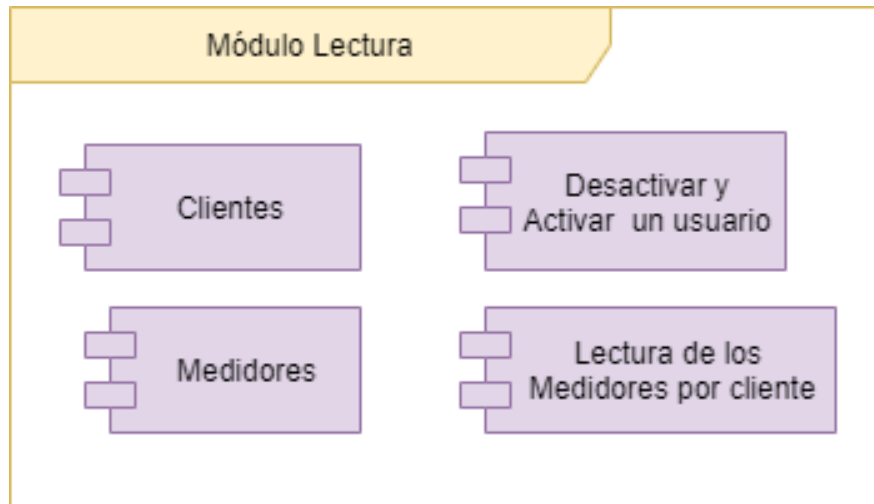


Figura 33: Módulo Lecturas

3.3.5.5. Diagrama de Clases

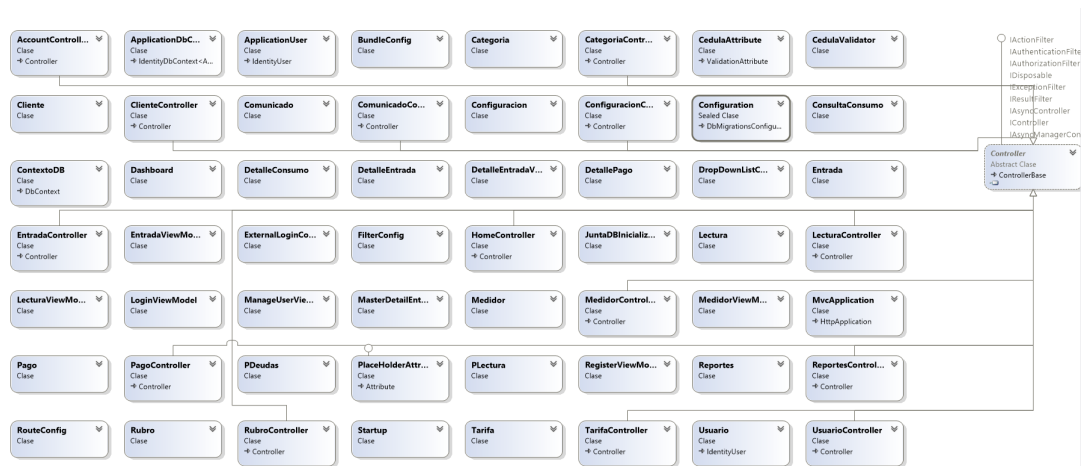


Figura 34: Diagrama de Clases

3.3.5.6. Diagrama de Entidad/Relación

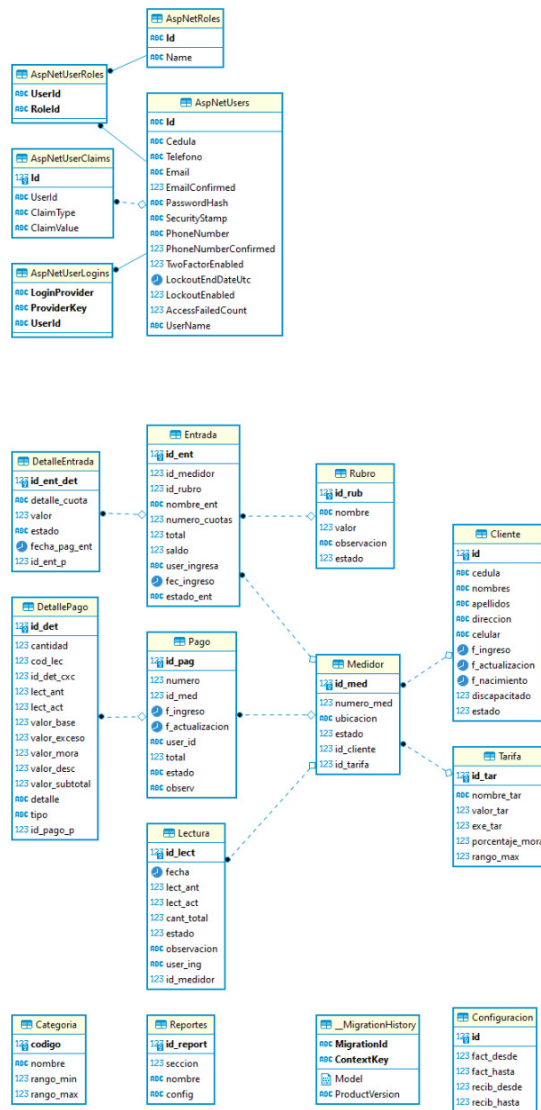


Figura 35: Diagrama de Entidad/Relación.

3.3.6. Vista de Despliegue

3.3.6.1. Introducción

Describe las configuraciones y despliegue del software y donde se lo ejecuta como puede ser computadora de escritorio, laptop, servidor web entre otros, además de la infraestructura necesaria para su instalación y funcionamiento óptimo del sistema

3.3.6.2. Distribución y Despliegue

El siguiente esquema indica el escenario esperado de distribución para la integración del SWP.

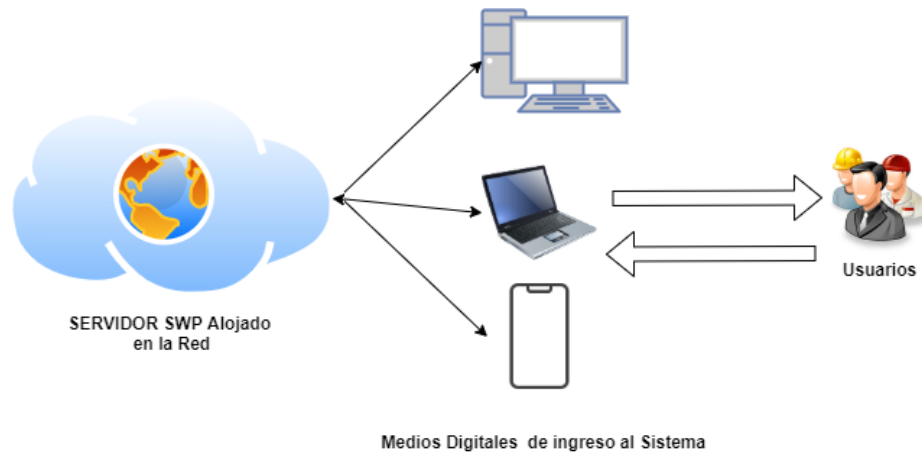


Figura 36: Distribución y Despliegue

Escenario del Entorno

- **Servidor Web** : es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente, en este caso por medio de puentes digitales tales como teléfonos inteligentes, laptop, Pc de escritorio.
- **Interfaz gráfica de usuario (GUI)**: esta interfaz usa tecnologías dentro de la plataforma .Net.
- **Cliente**: es el usuario de la GUI podrá administrar cada uno de los módulos basados en las restricciones y requerimientos establecidos en el documento anterior.

3.3.7. Vista de Datos

Remitirse a la sección 3.3.5.6 referente al Diagrama de Entidad/ relación, la misma da una visión de los datos a los que se puede acceder.

3.4. Plan de Pruebas de Software

3.4.1. Introducción

3.4.1.1. Objetivos

El plan de pruebas de software, es un conjunto de casos de pruebas con la finalidad de probar las funcionalidades del sistema, los mismo que son probados por módulos para saber si el software cumple o no con las necesidades de los usuarios finales.

3.4.1.2. Alcance

Las pruebas funcionales de la SWP se realizarán después de la implementación de cada módulo, es decir una vez finalizada alguna funcionalidad del sistema, se llevará a cabo su caso de prueba respectiva.

3.4.2. Instrumentos de Prueba

3.4.2.1. Módulos del Programa

En esta sección se presentan los módulos que componen la Interfaz además de las especificaciones de las pruebas a realizar en cada una de ellas

MODULO	PRUEBAS	DESCRIPCIÓN
Interfaz Gráfica de Usuario	Facilidad de uso	Esto permite evaluar si el usuario, en cada momento, tiene conocimiento sobre lo que puede y debería ser. En el caso de la interfaz gráfica, esta sirve para evaluar las funcionalidades y propiedades que brinda la SWP.
	Look & feel	Look & feel se refiere, a la apariencia que se le proporciona al usuario de la interfaz que integra la SWP de forma implícita para las pruebas.
Lógica de Negocio	Funcionalidad	La SWP debe cumplir con todos los requerimientos ya detallados en el documento referente a las especificaciones requeridas para el.
No funcionales	No funcionales	La SWP debe cumplir con los requerimientos no funcionales que se han especificado teniendo en cuenta el diseño.

Tabla 36: Descripción de los Módulos del Programa

3.4.3. Características a ser probadas

En esta sección se encuentran las características de la SWP a ser probadas, conforme al alcance del Proyecto.

Característica	Descripción	Módulo
Requerimientos Funcionales	Se hace analiza los casos de uso, para verificar si el sistema cumple o no con las funciones de los casos de uso establecidos en el proyecto	El módulo que permite probar esta característica es: Lectura de Planillas y consumo de agua

Tabla 37: Características a ser Probadas

3.4.3.1. Pruebas Funcionales

Estas son aplicadas a los requerimientos funcionales del sistema, se pone a prueba los aspectos funcionales y se estudia el comportamiento del usuario. Se desarrolló una Interfaz de Usuario (GUI) para poner a prueba los aspectos funcionales

Pruebas de Integración	Código: INT01
Actividades	Validación de los requerimientos funcionales
Herramientas	Prototipo del Sistema.
Tiempo estimado	40 minutos por prueba
Entregable	Acordar con el dueño del Producto

Tabla 38: Prueba de Integración

3.4.3.2. Pruebas de Comportamiento

En las pruebas de comportamiento se garantiza la calidad del sistema, ayuda a determinar si nuestro sistema es eficiente ante los requerimientos del usuario.

Pruebas de Comportamiento	Código: INT01
Actividades	Validación de los requerimientos funcionales
Herramientas	Captura del comportamiento del usuario.
Tiempo estimado	45 minutos por prueba
Entregable	Informe de las pruebas funcionales realizadas

Tabla 39: Pruebas de Comportamiento

3.4.4. Proceso de pruebas

Para realizar las respectivas pruebas se tomará en cuenta los casos de usos, En las tablas se muestra los casos de prueba respectivos

Nombre	CRUD TARIFAS /PRUEBA TARIFAS 02
Propósito	Verificar si es posible conectarse a una base de datos para la generación de las tarifas correspondientes.
Prerrequisitos	Generar las tarifas previamente
Ubicación	Base de datos Tarifas
Entrada	Acciones que se pueden realizar: - Generación de la Tarifa - Probar Tarifa - Establecer Tarifa - Modificar Tarifa
Oráculo	<ol style="list-style-type: none"> 1. CREACIÓN DE LA TARIFA : Mostrar los campos necesarios para la generación de la tarifa. 2. AÑADIR TARIFA: Probar si los datos son correctos . 3. ESTABLECER LA TARIFA : El programa mostrará mensaje de creación de la tarifa. 4. MODIFICAR O ELIMINAR TARIFA: Leer los datos usando los filtros de datos y con la posibilidad de modificarlos
Pasos	<ol style="list-style-type: none"> 1. CREACIÓN DE LA TARIFA- Abrir el programa - Clic en la opción tarifa del menú principal 2. AÑADIR TARIFA - Nos desplazamos hacia el botón nuevo para añadir una tarifa con los siguientes datos: nombre de la tarifa, valor de la tarifa, excedente de la tarifa, porcentaje de mora, rango máximo. 3. ESTABLECER LA TARIFA - el sistema podrá actualizar las tarifas respectivas 4. MODIFICAR O ELIMINAR TARIFA- dirigirse a la sección Acciones y se observará un icono de color azul para poder editar o una X en color rojo para eliminar dicha tarifa.
Posible Error	Controlado campo solo numérico y valor positivo.
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 40: Prueba CRUD Tarifas

Nombre	CRUD RUBROS /PRUEBA RUBROS 03
Propósito	Crear rubros con los siguientes datos: Nombre,Valor,Descripción
Prerrequisitos	Tener los datos necesarios previamente
Ubicación	Base de datos Rubros
Entrada	Acciones que se pueden realizar: - El SWP debe permitir actualizar los campo Rubro.
Oráculo	<ol style="list-style-type: none"> 1. CREACIÓN DE LOS RUBROS : Mostrar los campos necesarios para la generación de los rubros 2. ESTADO RUBROS: Probar si los datos son correctos . 3. ESTABLECER LOS RUBROS: El programa mostrará el cálculo del valor del rubro 4. MODIFICAR RUBROS: Leer los datos usando los filtros de datos y con la posibilidad de modificarlos
Pasos	<ol style="list-style-type: none"> 1. CREACIÓN DE LOS RUBROS : Click en la pestaña nuevo para añadir un nuevo rubro , se indicará el nombre el valor del rubro y si existe una observación de ser el caso. 2. ESTADO DE RUBROS: Verificar el estado con el mensaje Activo. 3. ESTABLECER LOS RUBROS: Se añade los datos para el cálculo correspondiente. 4. MODIFICAR RUBROS: en la pestaña editar podemos cambiar los datos correspondientes.
Posible Error	Controlado campo solo numérico y valor positivo.
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 41: Prueba CRUD RUBROS

Nombre	CRUD USUARIOS /PRUEBA USUARIOS 04
Propósito	El SWP será capaz de permitirle ingresar al sistema a través de sus credenciales, es decir mediante un usuario y contraseña
Prerrequisitos	Tener todos los datos previamente
Ubicación	Base de datos usuarios
Entrada	Acciones que se pueden realizar: El SWP será capaz de permitir crear un nuevo usuario con los siguientes datos: nombre, correo electrónico, cédula, contraseña y rol. Se debe permitir actualizar todos sus datos. Se debe permitir eliminar usuarios o desactivarlos.
Oráculo	<ol style="list-style-type: none"> 1. CREACIÓN DE USUARIOS: Mostrar los campos necesarios de ingreso de usuarios. 2. PROBAR USUARIOS: Probar si los datos son correctos 3. ESTABLECER USUARIOS: El programa ingresará el nuevo usuario 4. MODIFICAR USUARIOS: Se actualiza la información para del usuario seleccionado.
Pasos	<ol style="list-style-type: none"> 1. CREACIÓN DE USUARIOS- Abrir el programa - Clic en la opción usuarios del menú principal - Llenar el formulario con los datos de los usuarios 2. PROBAR USUARIOS - Seleccionamos ingresar usuarios 3. ESTABLECER USUARIO - Listar y verificar en la lista los usuarios 4. MODIFICAR USUARIO - Clic en la opción editar para modificar los datos del usuario.
Posible Error	Controlado campo solo letras, numerico, cedula
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 42: Prueba Usuarios

Nombre	CRUD CLIENTES /PRUEBA CLIENTES 05
Propósito	<p>Crear nuevos clientes con los siguientes datos: Nombres, Apellidos,dirección fecha de nacimiento y estado.</p> <p>Actualizar los datos del cliente.</p> <p>Leer los clientes usando filtros de sus datos.</p> <p>Desactivar y activar Usuario</p>
Prerrequisitos	Generar los clientes correspondientes
Ubicación	Base de datos clientes
Entrada	Acciones que se pueden realizar: - Creación de clientes - Probar clientes - Establecer cliente - Desactivar y Activar cliente
Oráculo	<p>1. CREACIÓN DEL CLIENTE: Mostrar los campos necesarios de ingreso del cliente.</p> <p>2. PROBAR USUARIO: Probar si los datos son correctos .</p> <p>3. DESACTIVAR Y ACTIVAR USUARIO: Se actualiza la información del cliente.</p>
Pasos	<p>1. CREACIÓN DEL CLIENTE- Pulsar el botón “nuevo” para añadir un cliente, se añadirán los datos como nombres número de cédula, teléfono.</p> <p>2. PROBAR CLIENTE - Para probar el cliente nos dirigiremos nuevamente al menú cliente y observamos que el cliente se encuentre agregado.</p> <p>3. DESACTIVAR Y ACTIVAR CLIENTE - En la pestaña estado podremos observar si el usuario está activo en color verde o desactivado en color rojo.</p>
Posible Error	Controlado campo solo letras, numerico, cedula
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 43: Prueba Clientes

Nombre	CRUD MEDIDORES /PRUEBA MEDIDORES 06
Propósito	<p>Crear uno o más medidores por cliente con los siguientes datos: Código,Categoría,Observación</p> <p>Leer los medidores por Cliente</p> <p>Actualizar Datos del Medidor</p> <p>Desactivar o Activar uno o todos los medidores de un cliente</p>
Prerrequisitos	Generar las categorías previamente
Ubicación	Base de datos usuarios y medidores
Entrada	Acciones que se pueden realizar: - Creación de Lectura del Medidor - Actualizar datos del Medidor
Oráculo	<p>1.CREACIÓN DE MEDIDOR : Nos permite crear medidor con datos como cliente,numero de medidor, tarifa.</p> <p>2.LECTURA DEL MEDIDOR: Observar el estado del medidor.</p> <p>3. ACTUALIZAR DATOS DEL MEDIDOR : Modificar los números del medidor.</p>
Pasos	<p>1.CREACION DE CATEGORIAS : Pulsar el botón nuevo para añadir un medidor.</p> <p>2.LECTURA DEL MEDIDOR: Seleccionamos la tarifa</p> <p>3. ACTUALIZAR DATOS DEL MEDIDOR : En la pestaña editar podemos modificar el número del medidor así como la tarifa.</p>
Posible Error	Controlado campo solo letras, numérico
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 44: Prueba Medidores

Nombre	CRUD LECTURAS /PRUEBA LECTURAS 07
Propósito	<p>Crear una lectura por medidor mensualmente con los siguientes datos: fecha, lectura anterior, lectura actual, consumo, estado y observación.</p> <p>Actualizar todos sus datos</p> <p>Leer las lecturas por medidor</p> <p>Eliminar lecturas por medidor</p>
Prerrequisitos	Generar los clientes y medidores
Ubicación	Base de datos usuarios y medidores
Entrada	Acciones que se pueden realizar: - Actualizar Datos; Leer Datos de Lecturas, Actualizar Lecturas.
Oráculo	<p>1.ACTUALIZAR DATOS: Mostrar los campos necesarios de ingreso para la lectura correspondiente.</p> <p>2. LEER DATOS DE LECTURAS: Probar si los datos son correctos .</p> <p>3. ELIMINAR LECTURAS: Se actualiza la información para añadir o actualizar la lectura de un medidor.</p>
Pasos	<p>1.ACTUALIZAR DATOS: Seleccionamos el periodo de consumo que queremos observar ejemplo: consumo Diciembre 2020</p> <p>2. LEER DATOS DE LECTURAS: El sistema arroja los datos con los datos correspondientes : medidor, ubicación, fechas, lecturas etc.</p> <p>3. ELIMINAR LECTURAS: En el recuadro acciones podemos actualizar el registro.</p>
Posible Error	Controlado campo solo numérico y valor positivo.
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 45: Prueba Lecturas

Nombre	CRUD CUENTAS POR COBRAR 08
Propósito	<p>Crear una cuenta por cobrar por medidor con los siguientes datos: fecha, medidor, cuotas, valor, saldo, estado, lectura actual observación.</p> <p>Crear los detalles de la cuenta por cobrar basados en los registros del módulo de rubros.</p>
Prerequisitos	Generar clientes y medidores
Ubicación	Base de datos clientes y CxCobrar
Entrada	Acciones que se pueden realizar: - Creación Cuenta por cobrar- Probar Cuenta por cobrar - Establecer Cuenta por cobrar - Modificar Cuenta por cobrar
Oráculo	<p>1. CREACIÓN CXCOBRAR: Mostrar los campos necesarios de ingreso de una cuenta por cobrar.</p> <p>2. PROBAR CXCOBRAR : Probar si los datos son correctos .</p> <p>3. ESTABLECER CUENTA POR COBRAR: El programa mostrará la base de datos a la que se conectó.</p> <p>4. MODIFICAR CUENTA POR COBRAR: Se actualiza la información para establecer una nueva categoría.</p>
Pasos	<p>1. CREACIÓN CXCOBRAR - Abrir el programa - Clic en la opción excobrar del menú principal - Pulsar botón nuevo para añadir una excobrar.</p> <p>2. PROBAR CXCOBRAR - Seleccionamos excobrar y verificamos en la lista su ingreso.</p> <p>3. MODIFICAR CXCOBRAR Clic en la opción editar para poder editar.</p>
Posible Error	Controlado campo solo numérico y valor positivo.
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Cientes y CxCobrar

Tabla 46: Prueba Cuentas por Cobrar

Nombre	CRUD CONSULTA PLANILLA /CONSULTA PLANILLA 09
Propósito	Debe mostrar las deudas pendientes de consumo de agua potable según el número de medidor consultado
Prerrequisitos	Generar las lecturas previamente
Ubicación	Base de datos usuarios y medidores
Entrada	Acciones que se pueden realizar: - Consultar planilla
Oráculo	1. CONSULTAR PLANILLA: Verificar los datos y número de clientes registrados 2. ACTUALIZAR PLANILLA: Base de datos usuarios y medidores.
Pasos	1. CONSULTAR PLANILLA: Añadir datos del usuario para obtener el consumo. 2. ACTUALIZAR PLANILLA: Seleccionar el periodo en el que queremos saber el consumo hecho y arrojado el valor en la planilla
Posible Error	Controlado campo solo numérico y valor positivo.
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 47: Prueba Cuentas por Cobrar

Nombre	CRUD FACTURACIÓN / CONSULTA FACTURACIÓN
Propósito	Debe mostrar las deudas pendientes de consumo de agua potable segundo el número de medidor consultado
Prerrequisitos	Generar las lecturas previamente
Ubicación	Base de datos usuarios y medidores
Entrada	Acciones que se pueden realizar: - Consultar planilla para Facturación
Oráculo	1. CONSULTAR PLANILLA PARA FACTURACIÓN: En base a los datos de los clientes. 2. ACTUALIZAR PLANILLA: En base a las lecturas de los medidores.
Pasos	1. CONSULTAR PLANILLA PARA FACTURACIÓN: Seleccionar el usuario correspondiente para que el sistema arroje los datos. 2. ACTUALIZAR PLANILLA: En caso de lecturas o ex cobrar mal ingresadas
Posible Error	Controlado campo solo numérico y valor positivo.
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 48: Prueba Facturación

Nombre	CRUD REPORTES /CONSULTA REPORTES 11
Propósito	Extraer información sobre cada uno de los módulos existentes en el sistema Incluir gráficos estadísticos,cálculos, filtros y más que reflejen la realidad de la Junta administradora de Agua Potable de Mollepamba
Prerequisitos	Generar las Facturas previamente
Ubicación	Base de datos usuarios,medidores,cxcobrar,clientes,lecturas
Entrada	Acciones que se pueden realizar: - Consultar planilla
Oráculo	1. INFORMACIÓN MÓDULOS: Verificar lo recaudado-reporte por fechas. 2. REPORTE DE CLIENTES Y MEDIDORES REGISTRADOS: Reporte de Cuentas por Cobrar. 3. VERIFICAR DATOS: Los datos serán exclusivos de cada cliente. 3. IMPRIMIR REPORTE: Formato Pdf
Pasos	1. INFORMACIÓN MÓDULOS: Verificar lo recaudado-reporte por fechas. 2. REPORTE DE CLIENTES Y MEDIDORES REGISTRADOS: 3. VERIFICAR DATOS: Guiarse en cada una de las pestañas las cuales arrojan los datos correspondientes a cada cliente. 3. IMPRIMIR REPORTE: Click en la opción imprimir.
Posible Error	Controlado campo solo numérico y valor positivo.
Evaluación de la prueba	Prueba satisfactoria.
Módulos Asociados	Base de datos Usuarios y medidores

Tabla 49: Prueba Reportes

3.4.5. Desarrollo de las interfaces

Módulo Pantalla Principal

Pantalla Principal de ingreso como Administrador

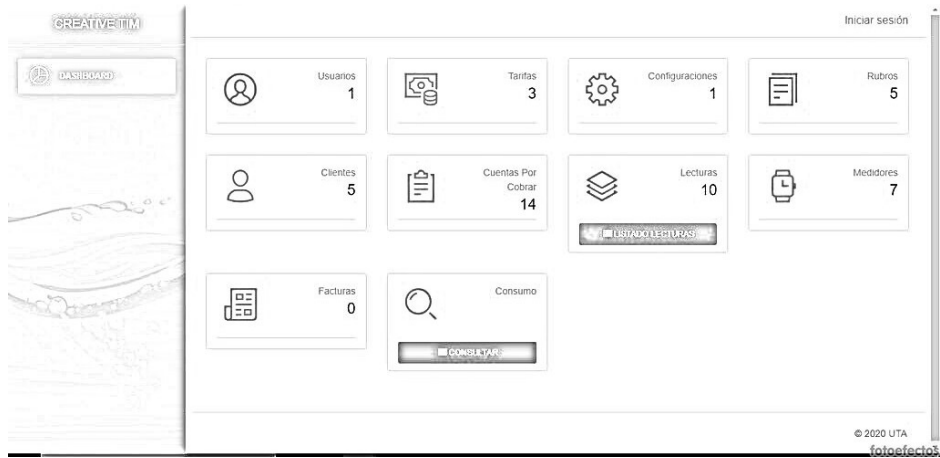


Figura 37: Boceto Pantalla Principal

Módulo Usuario/Contraseña

Iniciar sesión.

Utilice una cuenta local para iniciar sesión.

Nombre de usuario

Contraseña

¿Recordar cuenta?

fotoefectos

Figura 38: Boceto Conexión Usuario-Contraseña

Panel Lateral

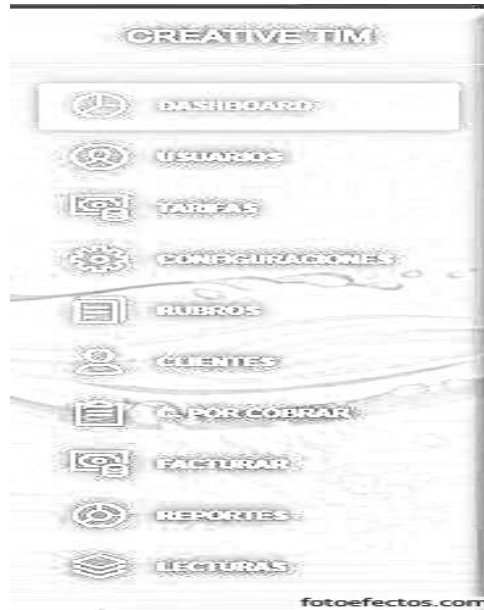


Figura 39: Boceto Menú lateral

Módulo Usuarios

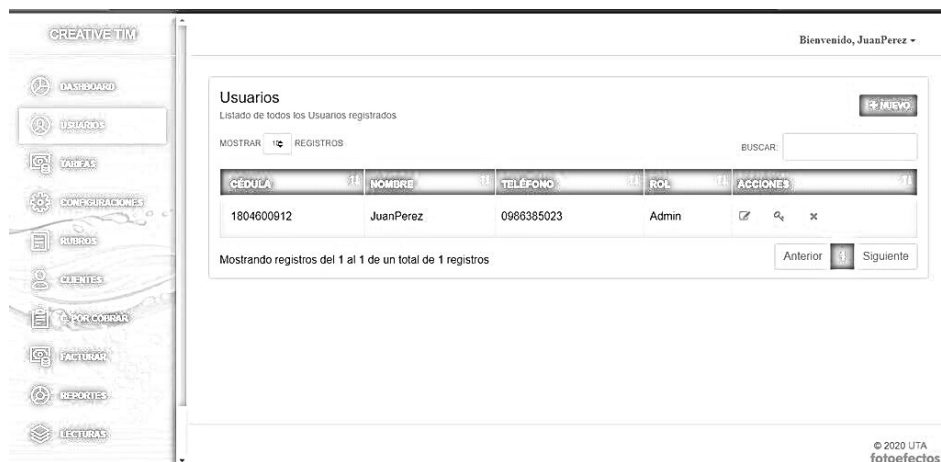


Figura 40: Boceto Módulo Usuarios

Módulo Tarifas



Figura 41: Boceto:Módulo Tarifas

Módulo Configuraciones



Figura 42: Boceto:Módulo Configuraciones

Módulo Rubros



Figura 43: Boceto:Módulo Rubros

Módulo Clientes



Figura 44: Boceto:Módulo Clientes

Módulo Cuentas por Cobrar



Figura 45: Boceto: Módulo Cuentas por Cobrar

Módulo Facturar



Figura 46: Boceto:Módulo Facturar

Módulo Reportes



Figura 47: Boceto:Módulo Reportes

Módulo Lecturas

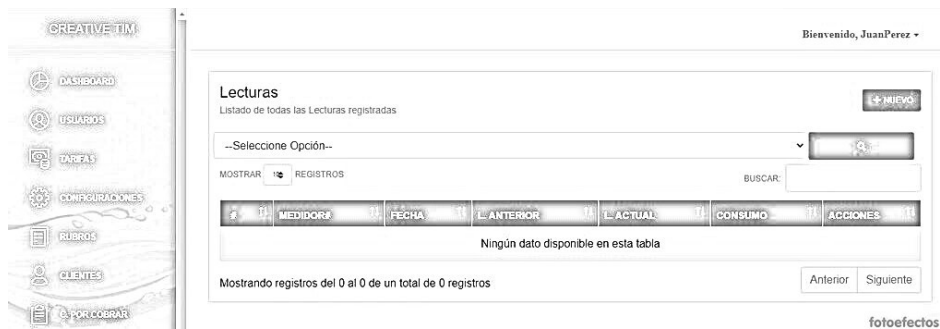


Figura 48: Boceto:Módulo Lecturas

3.4.6. Desarrollo de las interfaces utilizando Entity framework

Para la creación de las pantallas y las funcionalidades interviene Entity Framework (EF) que crea de forma automática los archivos que son personalizados en base a un Modelo que se crea como una tabla en la base de datos de SQLServer. Con el uso de la tecnología Blazor se crea la Vistas que personalizaremos a medida del cliente. La parte de los Controladores accederán a los datos con uso de la tecnología LINQ y no será necesario de la intervención de SQL en el proyecto. A continuación se muestra una imagen con la integración aprovechada de las diferentes tecnologías que ofrece EF basado en la arquitectura MVC.



Figura 49: EF y MVC.

A continuación se muestra un prototipo a seguir para la creación de un CRUD de uno de los módulos que en este caso será del módulo clientes.

```
9
10 namespace WebAppJuntaAdminAgua.Models
11 {
12     14 referencias
13     public class Cliente
14     {
15         [Key]
16         [Display(Name = "id de cliente")]
17         18 referencias
18         public int id { get; set; }
19
20         [Required]
21         [StringLength(13)]
22         [Index(IsUnique = true)]
23         [CedulaAttribute]
24         [Display(Name = "cedula de cliente")]
25         10 referencias
26         public string cedula { get; set; }
27
28         [Required]
29         [StringLength(100)]
30         [Display(Name = "nombres de cliente")]
31         18 referencias
32         public string nombres { get; set; }
33
34         [Required]
35         [StringLength(100)]
36         [Display(Name = "apellidos de cliente")]
37         18 referencias
38         public string apellidos { get; set; }
39
40         [Required]
41         [StringLength(100)]
42         [Display(Name = "telefono de cliente")]
43         18 referencias
44         public string telefono { get; set; }
45     }
46 }
```

100% No se encontraron problemas.

Figura 50: EF modelo entidad.

Se crea el modelo con las propiedades necesarias usando la sintaxis de EF. En consecuencia se actualiza la tabla que representa la entidad en la base de datos que en este caso es de los clientes. Existe una gran ventaja que al actualizar el modelo también se actualiza su entidad en la base de datos.

```

34 <div class="card-body ">
35     @using (Html.BeginForm())
36     {
37         @Html.AntiForgeryToken()
38         <div class="form-horizontal">
39             @Html.ValidationSummary(true)
40             <div class="row">
41                 <div class="form-group col-md-12">
42                     @Html.LabelFor(model => model.cedula, null, null)
43                     @Html.EditorFor(model => model.cedula, null)
44                     <div class="col-md-12">
45                         @Html.ValidationMessageFor(model => model.cedula)
46                     </div>
47                 </div>
48                 <div class="form-group col-md-6">
49                     @Html.LabelFor(model => model.nombres, null, null)
50                     @Html.EditorFor(model => model.nombres, null)
51                     <div class="col-md-12">
52                         @Html.ValidationMessageFor(model => model.nombres)
53                     </div>
54                 </div>
55                 <div class="form-group col-md-6">
56                     @Html.LabelFor(model => model.apellidos, new { @class = "control-label" })
57                     @Html.EditorFor(model => model.apellidos)
58                     <div class="col-md-12">
59                         @Html.ValidationMessageFor(model => model.apellidos)
60                     </div>
61                 </div>
62                 <div class="form-group col-md-6">
63                     @Html.LabelFor(model => model.direccion, new { @class = "control-label" })
64                     @Html.EditorFor(model => model.direccion)
65                     <div class="col-md-12">
66                         @Html.ValidationMessageFor(model => model.direccion)
67                     </div>
68                 </div>

```

Figura 51: EF vista entidad insertar o actualizar registro.

En la imagen anterior se acceden a los datos del modelo que instancia EF para su uso. Esta vista se puede generar automáticamente con el uso de Kit de generación de EF. Las funcionalidades que tenga la vista son declaradas en el controlador.

```

45 // POST: /Cliente/Create
46 // Para protegerse de ataques de publicación excesiva, habilite las propiedades específicas a las que desea enlazarse.
47 // más información vea http://go.microsoft.com/fwlink/?LinkId=317598.
48 [HttpPost]
49 [ValidateAntiForgeryToken]
50 public ActionResult Create([Bind(Include = "id,cedula,nombres,apellidos,direccion,celular,f_nacimiento,discapacitado")
51 {
52     var edad = DateTime.Now.Year - cliente.f_nacimiento.Year;
53     if (edad < 18)
54     {
55         TempData["Error"] = "La fecha de nacimiento no es correcta. La edad debe ser mayor a 18 años";
56         return RedirectToAction("Create");
57     }
58     if (ModelState.IsValid)
59     {
60         cliente.estado = true;
61         cliente.f_ingreso = DateTime.Now;
62         cliente.f_actualizacion = DateTime.Now;
63         cliente.direccion = "MOLLEPAMBA-PICAIHUA";
64         cliente.f_nacimiento = DateTime.Now;
65         db.Clientes.Add(cliente);
66         db.SaveChanges();
67         TempData["Message"] = "Cliente " + cliente.nombres + " + cliente.apellidos + " registrado correctamente";
68         return RedirectToAction("Index");
69     }
70 }
71 return View(cliente);
72 }

```

Figura 52: EF controlador función insertar.

En la imagen posterior se evidencia el método que hace posible el ingreso de un nuevo

registro haciendo uso del modelo Cliente y que se verá reflejado para el Cliente en la vista con el mismo nombre. En este punto se puede observar el avance y la facilidad que proporciona al trabajar con las propiedades del objeto que EF llama entidades que son instancias que representan a la tabla de la base de datos. Esta forma de trabajo se maneja en la mayoría de los módulos y así facilita el desarrollo de un sistema web eficiente.

3.4.6.1. Pantalla principal

La pantalla principal tiene un diseño de fácil manejo y entendible para cualquier usuario, muestra fácilmente las tareas que podemos realizar en el sistema.

La pantalla principal es interactiva nos indica de una forma clara cada uno de los CRUDS.

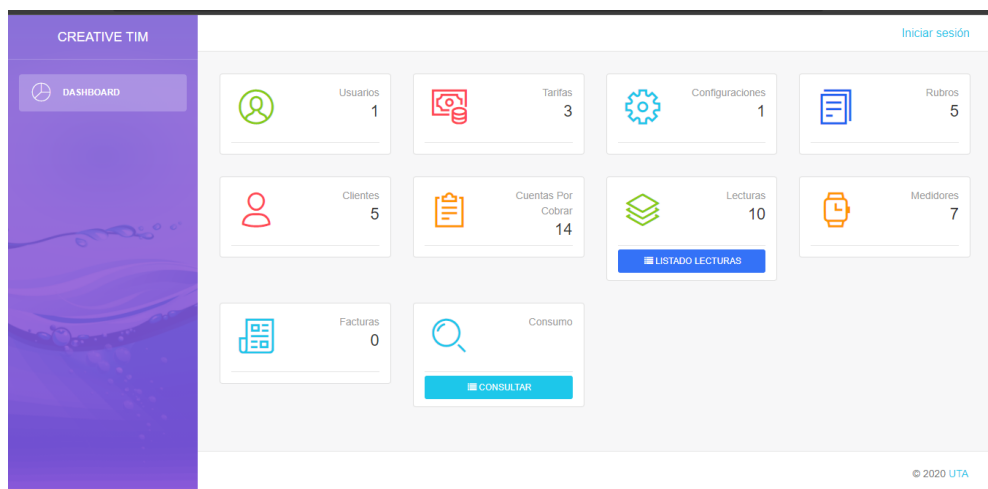


Figura 53: Interfaz:Pantalla Principal

Módulo Conexión

Para Iniciar Sesión se hace uso de una Cuenta de Usuario y su contraseña Respectiva.

Iniciar sesión.

Utilice una cuenta local para iniciar sesión.

Nombre de usuario

Contraseña

¿Recordar cuenta?

Figura 54: Interfaz Conexión-Usuario/Contraseña

Módulo Usuarios

Se visualiza los datos del usuario registrado con sus respectivo número de cédula, nombre ,y rol y actividad.

The screenshot shows a web application interface for the 'Usuarios' module. On the left is a purple sidebar with navigation options: DASHBOARD, USUARIOS (selected), TARIFAS, CONFIGURACIONES, RUBROS, CLIENTES, C. POR COBRAR, FACTURAR, REPORTE, and LECTURAS. The main content area is titled 'Usuarios' and includes a '+ NUEVO' button. Below the title is a search bar and a table of registered users. The table has columns for CÉDULA, NOMBRE, TELÉFONO, ROL, and ACCIONES. One user is listed: JuanPerez with ID 1804600912 and role Admin. The interface also shows pagination controls and a 'Bienvenido, JuanPerez' message in the top right.

CÉDULA	NOMBRE	TELÉFONO	ROL	ACCIONES
1804600912	JuanPerez	0986385023	Admin	[Iconos de acciones]

Figura 55: Interfaz Usuario

Módulo Tarifas

Se visualiza las tarifas junto con el cálculo de los valores correspondientes.

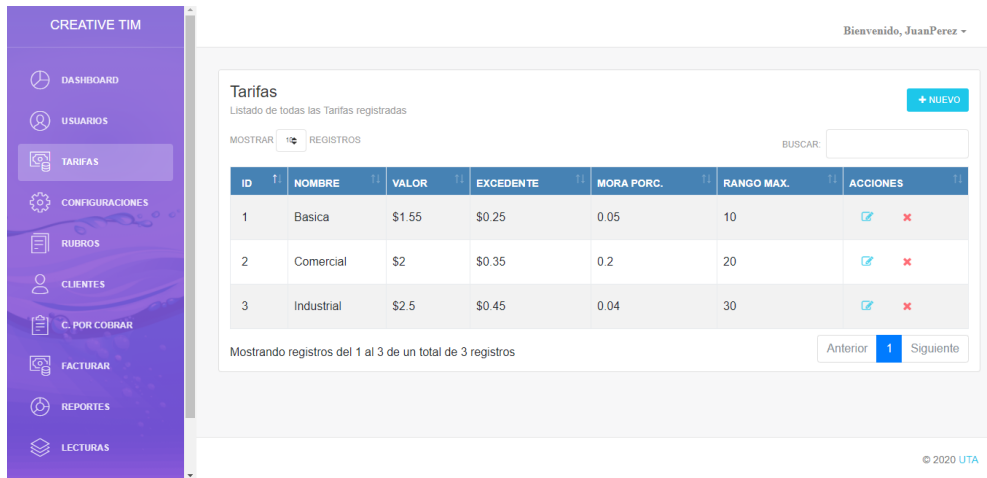


Figura 56: Interfaz Tarifas

Módulo Configuraciones

Se muestra las configuraciones con los datos del inicio y fin de la Factura, así como los recibos respectivos.



Figura 57: Interfaz Configuraciones

Módulo Rubros

Se visualiza los datos correspondientes y la tasa de valor calculado en función al rubro registrado.



Figura 58: Interfaz Rubros

Módulo Clientes

Se observa los datos de los clientes registrados de manera dinámica y de fácil comprensión.



CREATIVE TIM

DASHBOARD

USUARIOS

TARIFAS

CONFIGURACIONES

RUBROS

CLIENTES

C. POR COBRAR

Clientes

Listado de todas los Clientes registrados

+ NUEVO

MOSTRAR: 10 REGISTROS

BUSCAR:

CEDULA	NOMBRE	EDAD	DISCAPACIDAD	ESTADO	ACCIONES
1802061133	olga pauca	50	NO	Activo	i e + x
1802862324	Jaime Carrillo	-1	si	Activo	i e + x
1804327151	JOSE BONILLA	-1	NO	Inactivo	i e + v
1804494142	Walter Danilo Semblantes Quinche	31	NO	Inactivo	i e + v
1804600912	Patricio Landa	20	NO	Activo	i e + x

Figura 59: Interfaz Clientes

Módulo Cuentas por Cobrar

En este apartado podremos visualizar las cuentas por cobrar registradas por medio de los datos del Usuario.



CREATIVE TIM

DASHBOARD

USUARIOS

TARIFAS

CONFIGURACIONES

RUBROS

CLIENTES

C. POR COBRAR

Bienvenido, JuanPerez

Cuentas por Cobrar

Listado de todas las Cuentas por Cobrar registradas

+ NUEVO

CÉDULA CLIENTE

MOSTRAR: 10 REGISTROS

BUSCAR:

ID	RUBRO	CLIENTE	MEDIDOR#	NOMBRE	TOTAL	SALDO	ESTADO	ACCIONES
Ningún dato disponible en esta tabla								

Mostrando registros del 0 al 0 de un total de 0 registros

Anterior Siguiente

Figura 60: Interfaz Cuentas por Cobrar

Módulo Facturas

En este apartado podremos visualizar las cuentas por cobrar registradas por medio de los datos del Usuario.

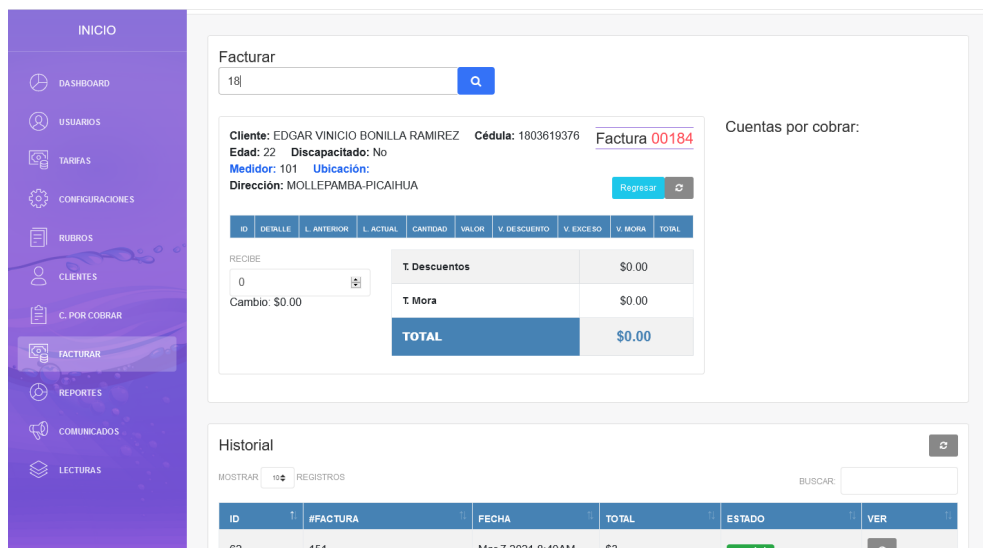


Figura 61: Interfaz facturar.

3.5. Implantación del software

3.5.1. Introducción

3.5.2. Resumen

3.5.2.1. Requisitos de Hardware

Para la implementación de ese sistema de registros de toma de lecturas y facturación de las planillas a partir de los datos de los medidores se a tomado en cuenta que nuestro sistema web se encuentre funcional y operativo, posteriormente como es de conocimiento este deberá ser incrustado en un hosting para su correcta validación. Este servicio provee a los usuarios de Internet un sistema para poder almacenar información, o cualquier contenido accesible vía web. De la ,misma manera optar por un servicio de hosting permitirá que el proceso de nuestra plataforma trabaje ininterrumpidamente para que tu web esté disponible siempre, para cualquier persona que quiera acceder a ella. En nuestro caso hemos comparado dos proveedores de hosting que serán responsables de mantener el servidor en funcionamiento y protegerlo de ataques maliciosos. Hay que tener en cuenta los requerimientos mínimos que se necesita para que una aplicación web pueda trabajar correctamente, las cuales detallamos:

Procesador: Procesador de x86 o x64 bits de doble núcleo de 1,9 gigahercios (GHz)

Memoria: 2 GB de RAM

Mostrar: Súper VGA con una resolución de 1024 x 768

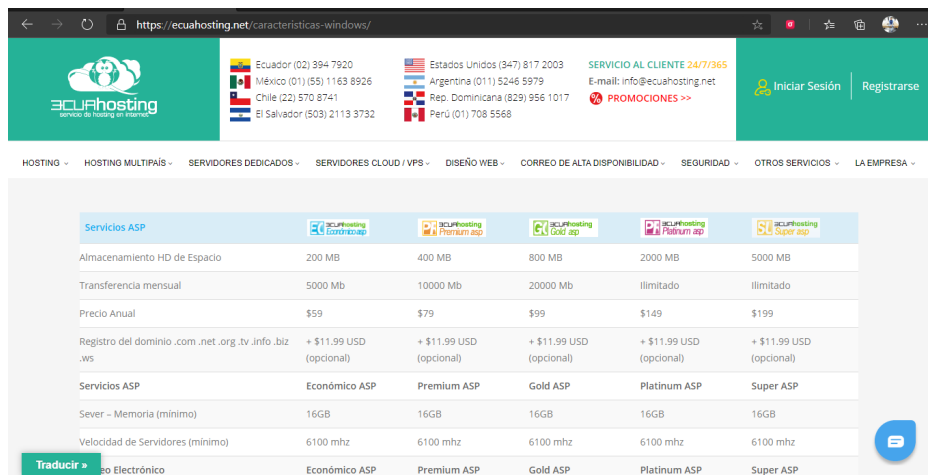
Ancho de banda superior a 50 KBps

Latencia inferior a 150 ms

Google Chrome (última versión lanzada públicamente)

Para lo cual en nuestro proyecto durante el proceso de pruebas se procedió a comparar entre estos servicios de hosting disponibles en Ecuador.

Ecuahosting :<https://ecuahosting.net/>

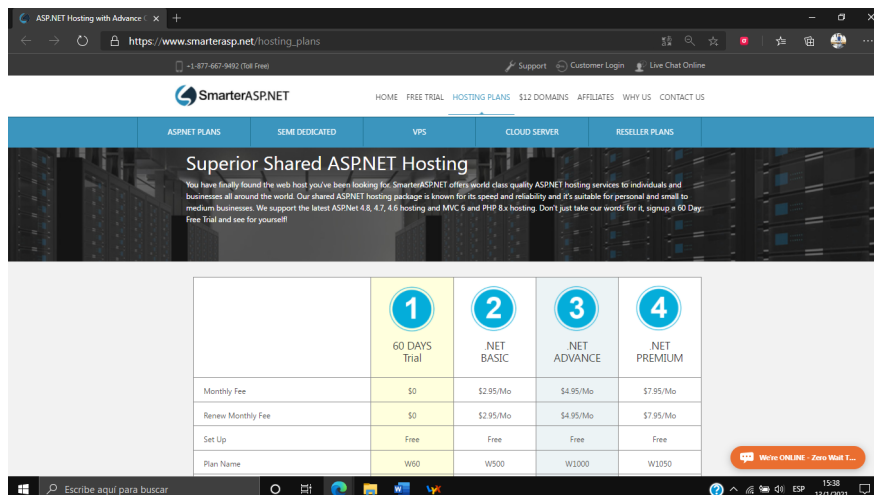


The screenshot shows the Ecuahosting website with a navigation menu and a table of hosting services. The table lists various plans with their respective features and prices.

Servicios ASP	Ecuahosting Contraseña	Ecuahosting Premium asp	Ecuahosting Gold asp	Ecuahosting Platinum asp	Ecuahosting Super asp
Almacenamiento HD de Espacio	200 MB	400 MB	800 MB	2000 MB	5000 MB
Transferencia mensual	5000 Mb	10000 Mb	20000 Mb	ilimitado	ilimitado
Precio Anual	\$59	\$79	\$99	\$149	\$199
Registro del dominio .com .net .org .tv .info .biz .ws	+ \$11.99 USD (opcional)	+ \$11.99 USD (opcional)	+ \$11.99 USD (opcional)	+ \$11.99 USD (opcional)	+ \$11.99 USD (opcional)
Servicios ASP	Económico ASP	Premium ASP	Gold ASP	Platinum ASP	Super ASP
Sever - Memoria (mínimo)	16GB	16GB	16GB	16GB	16GB
Velocidad de Servidores (mínimo)	6100 mhz	6100 mhz	6100 mhz	6100 mhz	6100 mhz

Figura 62: Ecuahosting

smarterasp.net: https://www.smarterasp.net/hosting_plans



The screenshot shows the smarterasp.net website with a navigation menu and a table of hosting plans. The table lists four plans with their respective features and prices.

	1 60 DAYS Trial	2 NET BASIC	3 NET ADVANCE	4 NET PREMIUM
Monthly Fee	\$0	\$2.95/Mo	\$4.95/Mo	\$7.95/Mo
Renew Monthly Fee	\$0	\$2.95/Mo	\$4.95/Mo	\$7.95/Mo
Set Up	Free	Free	Free	Free
Plan Name	W60	W500	W1000	W1050

Figura 63: smarterasp.net

Ventajas:

- De Bajo Costo.
- Perfecto para sitios web de negocios pequeños.
- No se necesitan conocimientos técnicos específicos

- Servidor pre-configurado Panel de control fácil de usar.
- El mantenimiento y la administración del servidor son realizados por el proveedor

Contras:

- Los aumentos del tráfico en otros sitios web pueden ralentizar el sitio

3.5.2.2. Dominio y NET PREMIUM Seguridad

El dominio escogido a sido el plan .NET PREMIUM correspondiente a la página de ventas de host: **smarterasp.net:**

La razones por las que optamos por este plan es que cuenta con los requisitos necesarios para el funcionamiento de nuestra web entre las que se destacan:

ESPECIFICACIONES	NET PREMIUM
Cuota Mensual	\$7.95/Mes
Renovar La Cuota Mensual	\$7.95/Mes
Establecer	Gratis
Nombre Del Plan	W1050
¡Garantía De Devolución De Dinero De 60 Días!	-

Tabla 50: PLAN SELECCIONADO DEL HOSTING

ESPECIFICACIONES	NET PREMIUM
Webpace(GB)	Ilimitado
Número De Entradas De Sitio/IIS	Ilimitado
Nombre De Dominio Alojado	Ilimitado
Banda	Ilimitado
IP Estática	\$2/Mes
Free Let's Encrypt SSL	cumple
Acceso A Archivos De Registro Sin Procesar	cumple
Panel De Control Basado En La Web	cumple
Ftp	Ilimitado
Nombre De Dominio	\$12/Año
Dominio Whois Privacidad	\$8/Año

Tabla 51: Características del Alojamiento

CARACTERÍSTICAS	NET PREMIUM
Se Permite El Acceso Remoto Del Administrador De IIS 10.X	Cumple
Delegación Remota Admitida, Web Deploy	Cumple
Visual Studio 2019/2017/2015/2013/2012/2010	Cumple
Compatibilidad Con WebMatrix	Cumple
Confianza Plena Y Confianza Media Apoyada	Cumple
Cambiar Entre El Modo Integrado Y El Modo Clásico	Cumple
Módulo De Reescritura De URL 2	Cumple
Grupo De Aplicaciones Dedicado	512 Mb
Ejecutar Múltiples Versiones De .NET Y PHP ASP.NET2.0 - 4.8, Core 1.X - 5.X, PHP 5.2 -	Cumple
Programar Tareas / Cron Jobs	Cumple
N.O De Conexiones Simultáneas	Ilimitado

Tabla 52: Características de Hospedaje de windows server.

CARACTERÍSTICAS	.NET PREMIUM
MSSQL 2019, 2017, 2016, 2014, 2012, 2008	20 DB
Espacio En Disco MSSQL	10 GB
MySQL 8.X, 5.X	20 DB
Espacio En Disco MySQL	10 GB
Datos SQL Alojados En Servidores Independientes	Cumple
Derechos DBO De SQL, Procedimientos De Almacén	Cumple
Búsqueda De Texto Completo	Cumple
SQL Reporting Service 2016, 2012	+\$4/Mes
Copia De Seguridad De SQL Y Restauración De SQL	Cumple
Adjuntar Herramienta De Archivo MDF	Cumple
Acceso Permitido De SQL Management Studio	Cumple
Autenticación De Base De Datos Contenida	Cumple
ASP.NET ReportViewer Control	Cumple
SQL C.E 4.0	Cumple
Conexión Remota	Cumple
MS Access	Cumple

Tabla 53: MySQL y SQL Server

CARACTERÍSTICAS	NET PREMIUM
Smartermail Enterprise Edition	Cumple
Cajas De Correo Electrónico	Ilimitado
SMTP, POP3, IMAP	Cumple
Acceso A Webmail	Cumple
Todas Las Demás Características De Smartermail	Cumple
Lista De Correo	Cumple
Suscripción Automática Y Cancelación De Suscripción	Cumple
Generador De Formularios De Lista De Correo	Cumple
API De Webservice De Lista De Correo	Cumple
Administrador De Plantillas De Lista De Correo	Cumple

Tabla 54: Características de la lista de correo electrónico.

CARACTERÍSTICAS	NET PREMIUM
ASP.NET 4.8, 2.0 - 4.7.X	Cumple
ASP.NET Core 1.X - 5.X	Cumple
ASP Clásico	Cumple
PHP 5.2.X - 8.X	Cumple
Cambiar Versiones De .Net	Cumple
Configuración De Directorio Virtual	Cumple
Configuración De ScriptMap	Cumple
Página De Error Personalizada	Cumple
MVC 6 Y 5, 4, 3, 2, 1	Cumple
ASP.NET Websockets	Cumple
WCF 4.0	Cumple
Nodo.Js (Nodejs)	Cumple
Crystal Report	Cumple
Servicios RIA De WCF	Cumple
MS SilverLight	Cumple
CGI/Perl CGI/Exe	Cumple
Ioncube	Cumple
Zend Optimizer	Cumple
WAP 1.1/2.0	Cumple

Tabla 55: Características del Scripting

CARACTERÍSTICAS	.NET PREMIUM
PHPMailer	Cumple
JMail	Cumple
ADO.NET	Cumple
Cdo	Cumple
Fso	Cumple
ASPEmail	Cumple
ASPUpload	Cumple
ASPJpeg	Cumple
ASPEncrypt	Cumple

Tabla 56: Componentes Instalados

CARACTERÍSTICAS	.NET PREMIUM
SSD Application Server Cloud	Cumple
SSD File Storage Cloud	Cumple
Servidor De Base De Datos SSD	Cumple

Tabla 57: Hardware

CARACTERÍSTICAS	.NET PREMIUM
Firewall	Cumple
Anti Virus / Anti Spam	Cumple
Copia De Seguridad Automática De Datos	Cumple
SSAE 16 SOC 2 Tipo 2 Cumplimiento	Cumple

Tabla 58: Seguridad

Finalmente hemos encontrado que el host web SmarterASP.NET ofrece los servicios de alojamiento ASP.NET de calidad aplicables tanto a individuos y empresas .

El paquete de alojamiento compartido ASP.NET es conocido por su velocidad y fiabilidad y es adecuado para empresas personales y pequeñas y medianas.

De igual manera esta plataforma admite los últimos ASP.Net 4.8, 4.7, 4.6 hosting y MVC 6 y PHP 8.x hosting.

3.5.2.3. Análisis de la Aplicación Web.

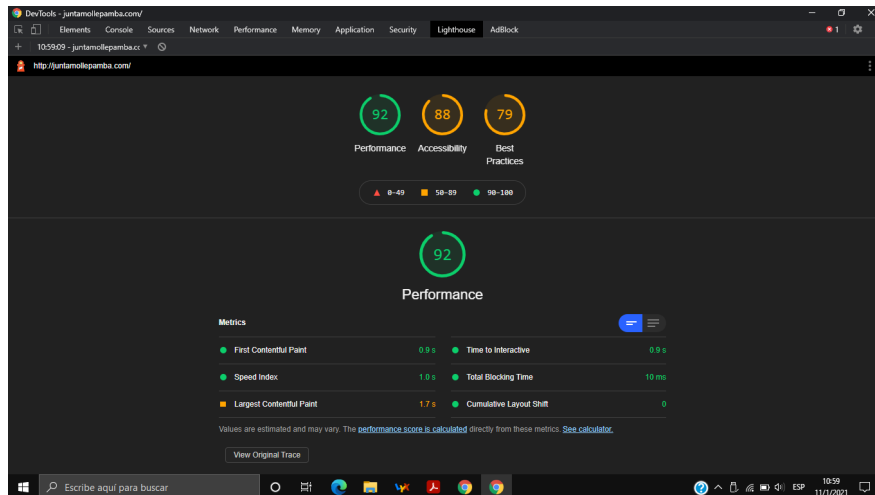


Figura 64: Aplicación Web Análisis 1

Se aplicó el DevTools que es un conjunto de herramientas de creación web y depuración integrado en Google Chrome. Usa DevTools para iterar y depurar tu sitio, y para crear un perfil de él.

El análisis y el porcentaje arrojado de la página es satisfactorio se observa que el rendimiento obtenido es de 92 % resultado que nos indica que se encuentra nuestro sistema web en buenas condiciones y trabajando eficientemente.

El parámetro de accesibilidad obtiene un puntaje de 88 % valor que está dentro del rango esperado para un correcto funcionamiento.

En caso de las buenas prácticas el puntaje obtenido es de 79 % que es un puntaje aceptable pero que puede ser mejorable.

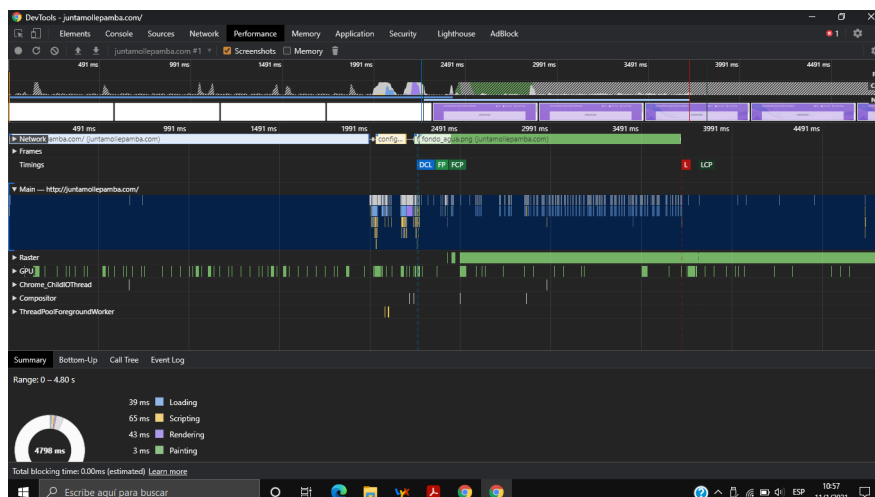


Figura 65: Análisis 2 Rendimiento de la página web

Mediante el panel Timeline de la herramienta DevTools obtenemos la siguiente información:

- Los archivos HTML son azules.
- Las hojas de estilo son púrpuras.
- Los archivos de medios son verdes.
- Otros recursos son grises.

3.5.2.4. SEO a la página web

Es importante resaltar que siendo la web destinada a la consulta de datos de varios usuarios y en su totalidad aloja gran cantidad de información que debe ser devuelta al cliente en el menor tiempo. El posicionamiento en buscadores, optimización en motores de búsqueda es una característica de las aplicaciones web destinadas a consulta de información. En la siguiente figura se establece una captura del resultado de la búsqueda usando el buscador de Google.

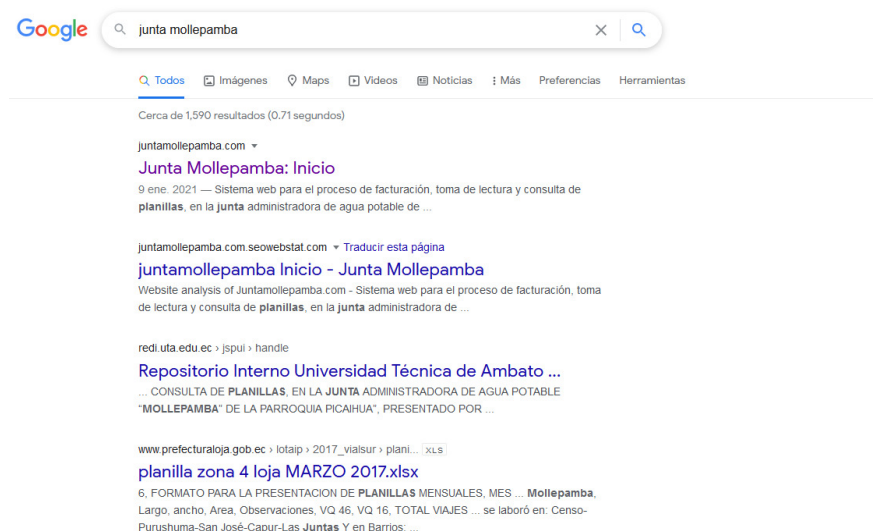


Figura 66: Primer lugar en la Búsqueda del Sitio Web-Google.

3.5.2.5. Configuraciones y costos de servicios

Compra y configuración del dominio juntamollepamba.com. Para la contratación de este servicio es necesario la verificación del nombre del dominio y contar con una tarjeta de débito o crédito para efectuar la transacción.

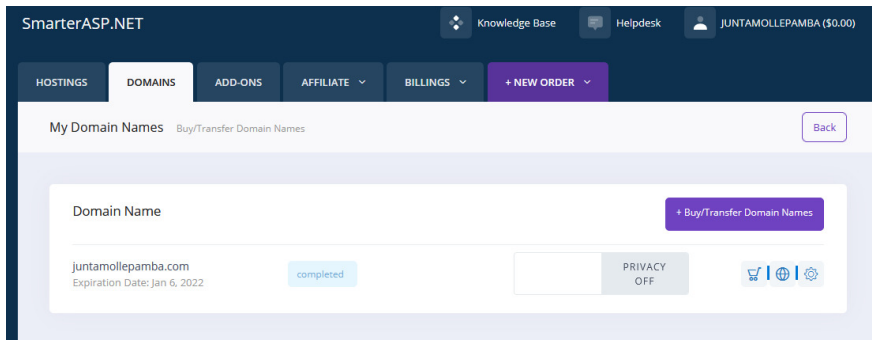


Figura 67: Compra y Configuración del Dominio Mollepamba en el Hosting

La compra y configuración de un certificado SSL (capa de puertos seguros) es muy importante. Este proporciona comunicaciones seguras por una red que protegen el sitio web configurado previamente.

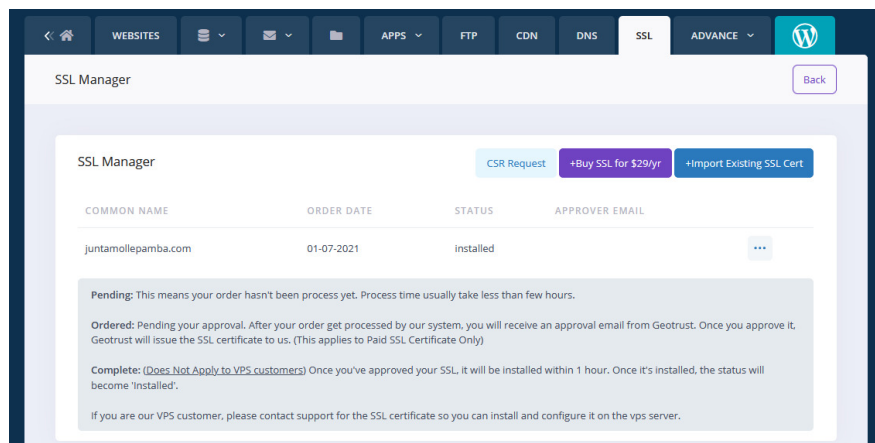


Figura 68: Configuración del Certificado SSL para Protección de los Datos.

Configuración de servicio adicional proporcionado por el Hosting. Cloudflare proporciona el servicio de protección de ataques DDoS (Ataque de denegación de servicio) y un firewall (cortafuegos) de aplicaciones web con reglas predefinidas que bloquea el acceso no autorizado.

Cloudfrade nos garantiza la capa de seguridad de la aplicación alojada.

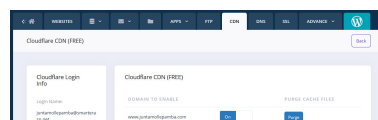


Figura 69: Configuración del Cloudflare-Servicio de Seguridad en el Hosting.

3.5.2.6. Costos de la instalación

En el siguiente cuadro se establecen los costos contratados en el Hosting (alojamiento) previamente seleccionado.

Detalle	precio
Hosting	\$56,00
Dominio	\$13,00
SSL	\$23,00
Cluodfrade	\$0,00
TOTAL	\$93,00

Tabla 59: Costos de servicios anuales instalados.

3.6. Aceptabilidad del software

En esta sección se establece la evaluación del cliente con respecto al sistema. En el **Anexo A** se establece el documento emitido por el Presidente de la Junta de Agua de Mollepamba quien fue parte de este proyecto y constato los diferentes módulos desarrollados. Este documento da fe de la calidad del producto del software entregado.

CAPÍTULO IV

CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

1. Definir un proceso de facturación eficiente nos permite simplificar márgenes de error, y ayuda a un sistema ser más eficiente y rápido.
2. La integración de esta plataforma permitirá a la población correspondiente a la Junta de Agua Mollepamba, acceder de forma veráz y sencilla para la consulta de sus planillas y por ende el costo de las mismas.
3. El cambio drástico que se ha tenido en el último año por la emergencia sanitaria ha obligado a las personas a adaptarse a la consulta y pagos de servicios por medio de plataformas digitales.
4. Con la metodología OPENUP se garantizó un buen proceso de desarrollo del software. Al trabajar con esta metodología permitió una integración directa con los usuarios del sistema, ellos evaluaron las características y el avance del mismo. Esta metodología facilito concentrarse en el desarrollo del proyecto sin perder la flexibilidad de lo planificado y documentando netamente lo necesario para que el proyecto tenga un alcance exitoso.
5. Con Entity Framework se facilitó el mapeo de datos, seguridad, accesibilidad y más recursos que lleva consigo. En el caso de la generación de los CRUD de los módulos mediante comandos se pudo generar y ocuparse más tiempo en el análisis que en el diseño. Además la creación de la base de datos sin la intervención de SQL hace que el avance sea definido en un solo marco de desarrollo en este proyecto.

4.2. Recomendaciones

1. Se recomienda la socialización de la nueva herramienta de consulta a los clientes de la Junta de Agua de Mollepamba mediante marketing digital y más medios de difusión de información local.

2. Se recomienda el estudio del desarrollo del módulo de pagos en línea mediante las plataformas de pagos nacionales para dar continuidad al proyecto y adaptarse a la realidad que se enfrenta.
3. En el caso de la metodología OPENUP se recomienda el uso de documentación basada en las normas ISO de cada etapa del desarrollo del software. En el caso de dar continuidad a este proyecto se recomienda anexar cambios y todo el proceso en una nueva iteración referenciada a la original.
4. En el caso de Entity framework como marco de desarrollo, se recomienda el uso de los productos de la familia de .NET, siempre que se tenga presupuesto, estos usan paquetes desarrollados directamente para su integración con el framework. En el caso de Paquetes externos o base de datos diferentes como por ejemplo: Postgres, MySQL necesitan de paquetes externos que no garantizan el correcto funcionamiento del framework, y se necesita de adaptaciones con componentes hechos para tecnologías externas a .NET. Además se recomienda realizar una documentación del código de forma global e descriptiva para eventuales cambios.

Bibliografía

- [1] Carolina Elizabeth Ruiz Chicaiza and Silvia Janeth Paguay Sánchez. Sistema de cobro de agua potable de la comunidad el socorro utilizando el framework asp.net mvc. B.S. thesis, Escuela Superior Politécnica de Chimborazo, 2017.
- [2] Erika Magdalena Llerena Ortíz. Sistema de facturación para el control automatizado de las tarifas recaudadas en las juntas administradoras de agua potable adscritas al parlamento agua del gobierno provincial de tungurahua. B.S. thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, 2011.
- [3] Franklin Andrés Cheza Luna. Sistema web de gestión de procesos para una junta de agua potable utilizando las tecnologías de software libre, jsf. B.S. thesis, 2014.
- [4] Kerly Patricia Chabla Vintimilla. Implementación de un sistema web de facturación y consulta para la junta administradora de agua potable de mobiloil. B.S. thesis, Universidad Tecnológica Indoamérica, 2017.
- [5] David Bainbridge, Dana McKay, and Ian H Witten. Guía del programador. 2002.
- [6] Franklin Ricardo Barrionuevo Caiza. Sistema de facturación e inventarios para el control tributario de compra y venta en la corporación vpc. B.S. thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, 2012.
- [7] Mónica Romero Alencastro et al. Implementación de metodología ágil en la gestión de proyectos de una entidad financiera en ecuador. B.S. thesis, 2020.
- [8] Elizabeth Gómez Gutiérrez, Marla Marcela Marcillo Guevara, Nicolás Ramírez López, et al. Metodologías ágiles para el desarrollo de proyectos. 2021.
- [9] Javier J Gutiérrez. Qué es un framework web? Available in: http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf Accessed May, 12, 2014.
- [10] Alexander Ivan Quinaluiza Arias. Interfaz de programación de aplicaciones para la generación automática de procedimientos almacenados en mysql. B.S. thesis, 2018.

- [11] Santiago Ríos Salgado, Cecilia Hinojosa Raza, and Ramiro Delgado Rodríguez. Aplicación de la metodología openup en el desarrollo del sistema de difusión de gestión del conocimiento de la espe.
- [12] Mark J Price. *C# 7.1 and .NET Core 2.0—Modern Cross-Platform Development: Create powerful applications with .NET Standard 2.0, ASP.NET Core 2.0, and Entity Framework Core 2.0, using Visual Studio 2017 or Visual Studio Code*. Packt Publishing Ltd, 2017.
- [13] Microsoft. Overview of entity framework core - ef core | microsoft docs. <https://docs.microsoft.com/en-us/ef/core/>, 2020. (Accessed on 10/20/2020).
- [14] Taylor Otwell. Laravel - the php framework for web artisans. <https://laravel.com/>, 2020. (Accessed on 10/20/2020).
- [15] Stackshare. .net vs laravel | what are the differences? <https://stackshare.io/stackups/dot-net-vs-laravel>, 2020. (Accessed on 10/20/2020).
- [16] Freelancer. Trabajos, empleo de laravel vs .net core | freelancer. <https://www.freelancer.cl/job-search/laravel-vs-.net-core/>, 2020. (Accessed on 10/20/2020).
- [17] Microsoft. Creación de aplicaciones en .net con c# - learn | microsoft docs. <https://docs.microsoft.com/es-es/learn/paths/build-dotnet-applications-csharp/>, 02 2020. (Accessed on 05/10/2021).
- [18] Microsoft. Referencia sobre la sintaxis de razor para asp.net core | microsoft docs. <https://docs.microsoft.com/es-es/aspnet/core/mvc/views/razor?view=aspnetcore-5.0>, 06 2020. (Accessed on 05/10/2021).
- [19] Microsoft. Información general de entity framework core - ef core | microsoft docs. <https://docs.microsoft.com/es-es/ef/core/>. (Accessed on 05/10/2021).